СЕРИЯ «ВЫСШАЯ ШКОЛА»

А.Ю. Щербаков

СОВРЕМЕННАЯ КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ

Теоретические основы
Практические аспекты

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО МИНИСТЕРСТВОМ ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ В КАЧЕСТВЕ УЧЕБНОГО ПОСОБИЯ ДЛЯ СТУДЕНТОВ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Москва • Книжный мир • 2009

А.Ю. Щербаков

Современная компьютерная безопасность.
Теоретические основы. Практические аспекты.
Учебное пособие. – М.: Книжный мир, 2009. – 352 с.

ISBN 978-5-8041-0378-2

Книга является уникальным изданием, объединившим под своей обложкой практически все актуальные вопросы компьютерной безопасности, начиная от теоретических моделей безопасности компьютерных систем и заканчивая практическими рекомендациями для аудита безопасности и подробным обзором стандартов и нормативных документов.

Помимо классических разделов компьютерной безопасности, книга содержит ряд уникальных материалов, которые невозможно найти ни в одной современной книге по данной проблематике – в первую очередь это инфраструктурные аспекты компьютерной безопасности, проблемы компьютерной надежности и защиты в операционных системах, а также системные вопросы и специальные разделы компьютерной безопасности.

Материалы книги приведены в строгой методической последовательности и взаимосвязи друг с другом, что позволяет читателю выстроить стройную взаимосвязанную картину современных методов компьютерной безопасности и защиты информации.

Книга предназначена для широкого круга специалистов в области информационной и компьютерной безопасности, инженеров-разработчиков систем безопасности, администраторов и аудиторов, а также для студентов и аспирантов, обучающихся по дисциплинам «Защита программ и данных», «Теоретические основы информационной безопасности», «Компьютерная безопасность».

ISBN 978-5-8041-0378-2

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Предлагаемая Вашему вниманию книга создана на основе лекций, семинаров и практических занятий, проводимых автором в течение более 10 лет в различных вузах – Московском институте электроники и математики, Московском инженерно-физическом институте, Институте криптографии, связи и информатики, а также Московском физико-техническом институте.

Проблематику книги составили в основном теоретические проблемы компьютерной безопасности. Целью ее написания является объединение материалов, касающихся компьютерной безопасности, в рамках одного издания. По сравнению с версией 2001 года книга значительно актуализирована и изменена ее структура.

Книга будет полезна при изучении курсов «Теоретические основы информационной безопасности» и «Защита в операционных системах».

Каждая часть имеет самостоятельную нумерацию определений, утверждений, таблиц и рисунков (исключая части 2 и 3, посвященные смежным проблемам).

ВВЕДЕНИЕ В ПРЕДМЕТ И МЕТОДЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

Компьютерная безопасность, информатика, электронные финансы и новая экономика

Современное общество можно охарактеризовать как пограничное между обществом материального производства и обществом постиндустриальным, информационным, характеризуемым новой политикой и экономикой, базирующейся на накоплении и обработке знаний и информации в различных формах. Происходящие процессы, связанные в первую очередь с глобализацией производства и потребления информации, в весьма скором будущем кардинально изменят и уже сейчас изменяют как способ общественного производства, так и многие общественные институты.

Важная задача в этом процессе – проанализировать, правильно истолковать и спрогнозировать существующие тенденции производства, передачи, обмена информацией, дабы предложить стремительно формирующемуся обществу будущего адекватные механизмы для работы с новыми категориями, понятиями и отношениями в сфере общественного производства и отношений.

В первую очередь необходимо отметить возрастание роли конкретного субъекта-индивидуума в процессе производства им информации в различных формах, в том числе и в форме товара. Из этой роли вытекает, с одной стороны, необходимость персонализации как самого субъекта, так и производимой и обрабатываемой им информации, защиту его интересов в сфере производства и потребления информации, а с другой – необходимость разработки и реализации механизмов безопасного и равноправного обмена произведенной информацией-товаром и адресного получения за нее материальных благ.

Вместе с тем, очевидно, что персонализированный индивидуум вступает в совершенно новые отношения с обществом и государством в лице их институтов. Эти отношения характеризуются в первую очередь высоким динамизмом и отказом от анонимности, а также равноправным участием персонализированного индивидуума в обмене информацией с государственными и общественными организациями. Высокий динамизм определен в первую очередь использованием глобальных компьютерных систем и сетей связи, а равноправный обмен информацией, защищающей интересы индивидуума в постиндустриальном обществе – это защищенный обмен, а также безопасность персональных данных и ресурсов индивидуального пользователя.

Изменяется медийное поле общества, поскольку персонализированный индивидуум, с одной стороны, не может являться источником непроверенной анонимной информации, а с другой – выступает потребителем

вполне конкретных услуг в информационном поле, организуя четкие обратные связи с общественными и государственными медиа-институтами. И если перед конкретным общественным институтом пользователь абсолютно не анонимен, то обязанность институциональных участников информационного обмена – обеспечить его конфиденциальность относительно не участвующих в нем субъектов.

В области финансов изменяется роль денег, которые в большей степени становятся «электронными». Электронным деньгам в полной мере присущи все три функции денег: измерение-учет, обращение и накопление. Особенностью «постиндустриальных» денег может и должна являться возможность их прямого обмена между индивидуумами («электронные кошельки»). Электронный обмен денежными эквивалентами – есть овеществленная безопасность этой информации, ибо ничего кроме символьных эквивалентов мер стоимости при нем не передается.

Совокупность названных факторов свидетельствует о формировании «новой экономики» или «экономики знаний», важнейшей чертой которой является неубывание информации как товара при ее обмене между участниками рынка, что позволяет говорить о потенциальной возможности стабильного экономического роста. Важнейшей стратегической задачей, как для личности, так и для государства становится создание механизмов реализации прав и обязанностей производителей информации, участников рынка информационных товаров и услуг, медийных корпораций, а также государственных и общественных институтов.

Таким образом, безопасность информации является стратегической целью постиндустриального общества. Безопасным должны быть все процессы производства, хранения и обмена информацией, все компоненты технического, программного, протокольного, институционного типа, участвующие в этих процессах. Термин «безопасный», являющийся интуитивно понятным, ниже будет уточнен в приложении к различным аспектам информационного взаимодействия и компонентам участвующих в нем компьютерных систем.

Как мы уже отмечали, существующие особенности и перспективы развития бизнеса создают предпосылки для перехода к электронным формам ведения бизнеса, что обуславливается и естественным ходом научно-технического прогресса общества в целом. При этом появление информационно-коммуникационных технологий привело не только к изменениям в организации бизнеса в виде автоматизированных систем управления производством, систем учета и контроля за движением товаров, электронного документооборота, но и к совершенствованию традиционных видов бизнеса.

Экономический эффект, который достигается за счет внедрения компьютерных систем в процессы организации бизнеса, заключается в ускорении и повышении качества работ с технической и финансовой документацией, в возможности внедрения передовых технологий, в частности, в автоматизированных производствах, в уменьшении зависимости эффективности бизнеса от места расположения предприятия, в расширении возможностей специализации и кооперации.

Другим принципиально новым аспектом, связанным с появлением электронного бизнеса, является создание новых сфер деятельности в области производства товаров и услуг. В первую очередь, это относится к развитию собственно новых видов телекоммуникаций и информационных технологий и возникающих при этом возможностей в области обработки, хранения и передачи информации. Здесь можно выделить такие составляющие, как производство оборудования и технических средств для телекоммуникационных и информационных систем, разработка программного обеспечения, создание банков информации и информационно-аналитических служб. Фактически, это направление связано с обеспечением инфраструктуры электронного бизнеса и разработкой инструментария информационных технологий.

Если абстрагироваться от конкретных видов бизнеса, то можно выделить ряд общих вопросов, которые необходимо решить с точки зрения организации и ведения электронного бизнеса. В большинстве своем эти системы дублируют аналоги из «традиционного бизнеса» на современном уровне развития общества.

С другой стороны, преимущества электронного бизнеса и использования современных информационных технологий создают предпосылки для формирования новых областей применения, например, для обеспечения деятельности правительств, чем и обусловлено появление термина «электронное правительство».

«Электронное правительство» – концепция осуществления государственного управления, присущая информационному обществу. Она основывается на возможностях компьютерных систем и ценностях открытого гражданского общества. Характеризуется направленностью на потребности граждан, экономической эффективностью, открытостью для общественного контроля и инициативы. Принято считать, что «электронное правительство» состоит из трех основных категорийных модулей (G2G – government to government, правительство правительству; G2B – government to business, правительство бизнесу; G2C – government to citizens, правительство гражданам) и включает в себя многочисленные прикладные элементы: свободу доступа граждан к государственной информации; перевод государственных органов на безбумажное делопроизводство; установление для всех государственных органов показателей эффективности работы на год и регулярный их контроль, который проводится как парламентом, так и гражданами; введение в государственных органах пластиковых карт для идентификации госслужащих, перечисление им зарплаты, расчетов за командировки, перенесение в сеть большинства стандартных транзакций между государством и гражданами или бизнесами, проведение тендеров. Электронное правительство тесно связано с такими компонентами информационного общества, как электронная коммерция, электронный бизнес, электронный банкинг, универсальный доступ к информационным ресурсам, электронное образование. Частным случаем данного понятия является использование систем электронной коммерции в отдельно взятых министерствах и ведомствах.

Вне зависимости от того, в какой сфере человеческой деятельности используются информационные технологии, будь то электронная коммерция или электронное правительство, можно выделить следующие основные технологические процессы, в которых используются современные КС:

- юридически значимое оформление документов в электронном виде;
- взаимодействие с клиентами информационных систем, например, заказ товаров и услуг или получение налоговой отчетности;
- доставка товаров и услуг потребителю;
- электронные платежные системы;
- управление организацией;
- кредитование организаций и предприятий.

Применительно к использованию КС существует ряд дополнительных проблем, которые не могут быть решены традиционными методами. К таким специфическим проблемам следует отнести:

- проблемы информационной безопасности;
- создание технологической основы для производства продукции и услуг в сфере обработки и передачи информации;
- страхование информационных рисков.

По существу решение каждой проблемы из перечисленной совокупности сопряжено с созданием соответствующей системы, регулирующей взаимоотношения между различными участниками электронного бизнеса, и обеспечением необходимой технической и технологической поддержки для функционирования. При этом каждая из указанных систем является неотъемлемой частью электронного бизнеса.

Отдельно остановимся на решении задач информационной безопасности. Традиционно принято считать, что основными задачи информационной безопасности являются:

- обеспечение конфиденциальности информации;
- обеспечение целостности и достоверности информации;
- обеспечение юридической значимости информации;
- обеспечение доступности информации и информационных ресурсов.

Градация приоритетов среди перечисленных задач информационной безопасности является вопросом, решаемым только в конкретных условиях применения и зависит от требований, предъявляемых непосредственно к информационным системам. Для государственных организаций на первом месте стоит конфиденциальность, а целостность понимается исключительно как неизменность информации. Для коммерческих структур, вероятно, важнее всего целостность, доступность и юридическая значимость информации. По сравнению с государственными, коммерческие организации более открыты и динамичны, поэтому вероятные угрозы для них отличаются и количественно, и качественно.

Различия будут также и в уровне предъявляемых требований по обеспечению информационной безопасности, которые продиктованы как моделью нарушителя для информационной системы, так и уров-

нем значимости защищаемой информации. Также различия будут проявляться в типе и уровне используемых средств защиты информации, например, если для клиентов интернет-магазинов, проводящих сделки на небольшие суммы, возможно использование криптографических алгоритмов, реализованных в современных операционных системах (ОС), то для инфраструктуры, обеспечивающей проведение государственных тендеров, должны использоваться совершенно другие решения, прошедшие сертификацию в соответствии с принятой государством политикой сертификации.

Соответственно и комплексы мер защиты информации, применяемые в том или ином случае, будут различными. Если в государственных структурах возможно использование широкого перечня организационных мероприятий в силу того, что информационные системы государственных структур являются более закрытыми, то большинство систем электронной коммерции, используемых коммерческими структурами, является потенциально открытыми, и поэтому здесь организационные мероприятия будут представлены менее значительно и, возможно, будут ограничиваться только регламентами и договорами использования такой системы.

Современное содержание компьютерной безопасности

Информационная безопасность (ИБ) – есть динамическое состояние защищенности интересов (целей) организации или владельца информации в условиях угроз в информационной сфере.

Защищенность достигается обеспечением совокупности свойств информационной безопасности – конфиденциальностью, целостностью, доступностью информации и инфраструктуры ее обработки. Приоритетность свойств информационной безопасности определяется значимостью информации для интересов (целей) организации или владельца информации.

Информационная сфера представляет собой совокупность информации, информационной инфраструктуры, субъектов, осуществляющих сбор, формирование, распространение, хранение и использование информации, а также системы регулирования возникающих при этом отношений.

Понятие **компьютерной безопасности** является более узким по отношению к информационной безопасности. Компьютерная безопасность ограничивает информационную сферу следующими компонентами: владельцы и пользователи информации (лица или организации), средства компьютерных (автоматизированных) систем, используемые для хранения, обработки и передачи информации пользователей, институциональные и нормативные формы, используемые для реализации такой деятельности, понимаемой как достижение целевой функции в рамках сформулированных интересов и целей. Интересы и цели являются внешними, априорно заданными по отношению к информационной сфере и описываются в категориях «назначение системы».

Осознание информационной безопасности – есть понимание организацией или лицом необходимости самостоятельно на основе принятых ценностей и накопленных знаний формировать и учитывать в рамках основной деятельности (например, творчества или бизнеса) прогноз результатов от деятельности по обеспечению информационной безопасности, а также поддерживать эту деятельность адекватно прогнозу.

Осознание информационной безопасности является внутренним побудительным мотивом организации или человека инициировать и поддерживать деятельность по поддержанию информационной безопасности, в отличие от побуждения или принуждения, когда решение об инициировании и поддержке деятельности по поддержанию информационной безопасности определяется соответственно либо возникшими проблемами организации, такими, как инцидент информационной безопасности, либо внешними факторами, например, требованиями законов.

Политика информационной безопасности – совокупность правил, процедур, практических приемов и руководящих принципов в области информационной безопасности, которыми руководствуется организация или человек при обеспечении информационной безопасности.

Любую деятельность сопровождают риски, сущность которых – естественная неопределенность, порождаемая движением во временной координате и взаимодействием с другими активными сущностями различных порядков. Это – объективная реальность, и понизить эти риски можно лишь до уровня неопределенности сущностей, характеризующих природу конкретной деятельности. Оставшиеся риски, определяемые факторами среды деятельности, на которые невозможно влиять, характеризуют потенциальные угрозы. При этом степень необходимой защищенности информационной сферы определяется анализом и оценкой рисков ИБ, которые должны быть согласованы с рисками основной деятельности организации или человека.

Деятельность организаций осуществляется через реализацию трех групп процессов: основные процессы (процессы основной деятельности), вспомогательные процессы (процессы по видам обеспечения), процессы менеджмента (управления) организацией. Процессы по обеспечению ИБ организаций составляют один из видов вспомогательных процессов, реализующих поддержку (обеспечение) процессов основной деятельности организации в целях достижения ею максимально возможного результата.

В основе исходной концептуальной схемы информационной безопасности лежит противоборство собственника[1] и злоумышленника[2] за

1. Под собственником здесь понимается субъект хозяйственной деятельности, имеющий права владения, распоряжения или пользования активами, который заинтересован или обязан (согласно требованиям законов или иных законодательных или нормативно-правовых актов) обеспечивать защиту активов от угроз, которые могут снизить их ценность или нанести ущерб собственнику.

2. Под злоумышленником здесь понимается лицо, которое совершает или совершило заранее обдуманное действие с осознанием его опасных последствий или не предвидело, но должно было и могло предвидеть возможность наступления этих последствий (адаптировано из ст. 27 УК РФ). Далее по тексту данные лица именуются злоумышленниками (нарушителями).

контроль над информацией в той или иной материальной форме. Однако другие, незлоумышленные, действия также лежат в сфере нашего рассмотрения.

В рамках данной книги мы рассмотрим компьютерную систему (КС) как объект системного анализа, субъекты и объекты КС, методы обеспечения безопасности в компьютерной системе и основные понятия криптографии.

Далее обратимся к понятиям политики безопасности и реализации политик безопасности в КС. Важной проблемой в современной теории компьютерной безопасности является обеспечение гарантий выполнения политики безопасности.

Для решения вопросов использования различных программных средств как для реализации целевой функции КС, так и для реализации различных механизмов безопасности, необходимо изучить безопасное субъектное взаимодействие в КС и инфраструктурные вопросы компьютерной безопасности.

Следующий раздел в книге посвящен управлению безопасностью в компьютерной системе.

Крайне важная проблема – модели сетевых сред и обеспечение безопасности в распределенной компьютерной системе позволяет строить модели сетевого взаимодействия компьютерных систем.

Компьютерной безопасности и надежности, а также защите в операционных системах посвящен отдельный раздел книги.

Раздел, посвященный проблемам интеграции и специальным разделам компьютерной безопасности включает методологию создания защищенных компьютерных систем, описание типовых архитектур безопасности, защиту объектов КС при изменении их формы, защищенный документооборот, электронную коммерцию, защиту объектов интеллектуальной собственности, аудит компьютерной безопасности и вопросы эксплуатации защищенных систем и понятие системы обеспечения информационной безопасности.

Финальный раздел книги посвящен рассмотрению нормативных документов для решения задач компьютерной безопасности.

1. МОДЕЛИ КОМПЬЮТЕРНОЙ СИСТЕМЫ И МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

1.1. Компьютерная система как объект системного анализа. Субъекты и объекты

В современной информатике **модель компьютерной системы** (КС) чаще всего рассматривается в виде **совокупности элементов**, которые можно разделить на два подмножества: множество **объектов** и множества **субъектов**.

В любой системе с точки зрения системного анализа выделяются существенные для ее качественной определенности части, подсистемы или компоненты. В данном случае компонентами КС будут субъекты и объекты. Данное разделение основано на свойстве элемента компьютерной системы «быть активным» или «получать управление» (в компьютерной литературе применяются также термины «использовать ресурсы» или «пользоваться вычислительной мощностью»). Оно исторически сложилось на основе модели вычислительной системы, принадлежащей фон Нейману[1], согласно которой последовательность исполняемых инструкций для вычисляющего процессора (программа, рассматриваемая как «субъект» компьютерной системы) находится в единой среде с данными (выступающими в качестве «объекта»).

Здесь необходимо сделать важное уточнение. В гуманитарных науках (в частности, в юриспруденции) под *субъектами* имелись в виду **люди**, а под *объектами* – **организации, технологические процессы, материальные продукты и услуги**, то в данном случае мы понимаем под субъектом **программу**, управляемую человеком, а под **объектом** – данные, обрабатываемые или порождаемые этой программой.

Сформулируем важнейшие свойства субъектов, которые также относятся к числу системообразующих компонентов компьютерной системы. Самое главное из них состоит в том, что пользователь-человек воспринимает объекты и получает информацию только через субъекты, которыми он управляет и которые отображают информацию, относящуюся к окружающему миру.

На практике пользователь сообщает компьютерной системе свои запросы, используя такие инструменты управления, как клавиатура, «мышь», джойстик, сенсорный экран, электронное стило, которые являются внешним оборудованием компьютера и передают информацию субъектам нижнего уровня, обслуживающим эти устройства и также передающим информацию далее, субъектам или программным модулям операционной системы, обеспечивающим функционирование компьютера в целом. Отличие терминов «программа» и «программный модуль» состоит в том, что программа является системной целостностью более высокого порядка, чем программный модуль, а программный модуль является подсистемой, обладающей в рамках программы особой целостностью.

1. Биктимиров М.Р., Щербаков А.Ю. Избранные главы компьютерной безопасности. – Казань: Изд-во казанского матем. общества, 2004. – 372 с.

Из этого следует что, программа состоит из взаимосвязанной совокупности программных модулей. Программа, как системная целостность, предназначена для решения законченной задачи, которая сформулирована ее разработчиком. Модули же решают отдельные подзадачи. Например, программа текстового редактора Microsoft Word, предназначенного для полнофункциональной работы с текстами и электронными документами, состоит из нескольких десятков программных модулей, часть которых относится к операционной среде Windows. Выделение программного модуля оправдано при решении задач управления доступом, о которых будет сказано ниже, а также при разработке программ для решения частных задач пользователей.

Субъекты бывают разного уровня: нижнего – драйверы, обслуживающие внешние устройства компьютера, среднего – программы–субъекты операционной системы, обеспечивающие работу компьютера и пользователя независимо от решаемых ими задач, и верхнего – прикладные программы, обеспечивающие выполнение целевых функций, например, поиск, анализ, визуализация информации, составление отчетов и т. д.

Передача информации от субъектов верхнего уровня также происходит иерархически, только направление передачи информации меняется. Прикладные программы передают результаты своей деятельности операционной среде. Она в свою очередь передает информацию драйверам средств отображения, выводящим информацию на экран или другие средства визуального или графического отображения. Например, команда пользователя в меню программы текстового редактора «Сохранить файл» приводит к тому, что набранный в редакторе текст передается модулям операционной системы, последовательно передающим его модулям, управляющим работой жестких дисков или флеш–носителей. И только после этого на диске возникает файл, содержащий набранный текст.

Передача информации от одного объекта к другому происходит по инициативе субъекта, а сама такая передача называется «потоком» или **«потоком данных»**.

Изменение и порождение новых объектов компьютерной системы производится субъектом, как активной компонентой, опосредованно управляемым пользователем. Именно субъекты порождают потоки информации и изменяют состояние объектов. Субъекты также могут влиять друг на друга через изменяемые ими объекты.

Будем считать разделение компьютерной системы на субъекты и объекты априорным. Будем считать также, что существует безошибочный критерий различения субъектов и объектов (по свойству их активности). Кроме того, считаем, что декомпозиция (разложение) КС на субъекты и объекты фиксирована. На практике это означает стационарно протекающий этап работы, когда используемые субъекты не обновляются и не уничтожаются. На языке же администраторов это означает работу «с установленным и зафиксированным и неизменяемым софтом». В терминах системных целостностей это соответствует тому, что КС относится к стабильной или функционирующей системе.

Подчеркнем отличие понятия «субъекта компьютерной системы» от «человека–пользователя» следующим определением. **Пользователь** – лицо (физическое лицо), аутентифицируемое некоторой инфор-

мацией и управляющее субъектом компьютерной системы через органы управления компьютером. Пользователь КС является, таким образом, внешним фактором, управляющим состоянием субъектов. **Аутентифицируемость пользователя** означает, что он должен некоторым образом «представить себя» управляемой им КС, в противном случае компьютерная система не различит одного пользователя от другого. Представление пользователя компьютерной системе протекает обычно в два этапа: первый этап – **идентификация** – пользователь указывает свое **имя**, второй – собственно **аутентификация** – пользователь подтверждает свою индивидуальность некоторой никому не известной информацией, обычно **паролем**. Процедуры идентификации и аутентификации есть своего рода «основа» защищенной компьютерной системы, поскольку без точного определения пользователей, без фильтра «свой-чужой» невозможно определение прав и функций пользователя в системе. Кротко опишем основные подходы, используемые при идентификации и аутентификации пользователей КС. Эта информация важна как для пользования защищенной КС, так и для ее разработки.

Обратимся теперь к понятию распределенной компьютерной системы. Заметим, что в качестве синонима часто употребляют термин информационно–телекоммуникационная система. Чтобы понять, что это такое, введем понятия локального и внешнего сегмента КС.

Локальный сегмент КС (ЛС КС) – подмножество субъектов и объектов компьютерной системы, выделяемое по одному из следующих критериев:

• критерию группирования в одно множество всех субъектов с возможностью непосредственного управления субъектами;

• критерию локализации некоторого подмножества объектов и субъектов в рамках некоторой технической компоненты КС (одного компьютера или одной локальной сети);

• критерию присвоения объектам и субъектам КС некоторой информации, однозначно характеризующей субъект или объект (которая, как правило, называется адресом или сетевым адресом ЛС КС).

Поясним введенные критерии выделения локальных сегментов. В первом случае (все субъекты управляются одним или несколькими пользователями) речь идет о персональном рабочем месте пользователя, во втором – о корпоративной локальной сети, содержащей несколько (возможно, что и достаточно большое число) рабочих мест, в третьем – о «скрытии» некоторой структуры за сетевым именем или адресом (например, mail.ru – огромное количество серверов, управляющих компьютеров и каналов связи, расположенных в нескольких странах, но решающих одну задачу по хранению и доставке электронной почты).

Внешний сегмент КС – дополнение множества субъектов и объектов локального сегмента до всего множества объектов КС. Очевидно, что во внешнем сегменте могут быть выделены несколько локальных. В дальнейшем для простоты будем рассматривать один произвольно выделенный ЛС КС. Данное определение необходимо пояснить, поскольку оно излишне «математично». Речь идет о том, что, выделив ЛС КС по одному из предложенных критериев, мы весь остальной «мир» компьютерных систем считаем «внешним» относительно себя.

Удаленным субъектом будем называть субъект, принадлежащий множеству субъектов внешнего сегмента КС. Очевидно, что множества субъектов локального и внешнего сегмента КС не пересекаются.

Доступ удаленного субъекта к локальному объекту подразумевает организацию **потока информации** (формальное определение потока мы дадим ниже) от удаленного субъекта к объектам локального субъекта.

Понятие потока описывает реальную работу компьютерной системы под управлением пользователя. Потоки во «внешний мир» соответствуют запросам во внешние информационные хранилища и базы данных, обратные потоки – ответы на эти запросы. Потоки от одного ЛС КС к другому через внешний сегмент КС означают информационный обмен между пользователями. Не всегда эти потоки могут быть «дружественны» – например, регулярный внешний поток от вашей локальной сети к неизвестному локальному сегменту, скорее всего, означает, что у вас крадут корпоративную информацию или, что еще более печально, в корпоративной сети имеется *внутренний злоумышленник* («крот»), который передает конфиденциальную информацию вовне. Не всегда это человек, чаще – программа, но подробнее об этом – ниже.

Потоком информации между объектом O_m и объектом O_j называется произвольная операция над объектом O_j, реализуемая в субъекте S_i и зависящая от O_m. Обозначения: ***Stream(***S_i***,*** O_m***)–>*** O_j – поток информации от объекта O_m к объекту O_j. При этом поток информации произведен субъектом S_i. При этом для практической исследовательской работы полезно и следует выделять источник (O_m) и получатель (приемник) потока (O_j). Значения индексов принимают целые положительные значения и означают тот простой факт, что все субъекты и объекты могут быть перенумерованы. Приведенные только что определения и разъяснения важны еще и потому, что они позволяют понять, как конструируются узко локальные КС и КС, соединяющие несколько локальных систем, и притом, конструируются так, чтобы сохранить конфиденциальность компьютерной информации и предотвратить доступ к ней злоумышленников, например, пользователей или программистов конкурирующей организации.

Существует весьма важное отношение **транзитивности**, которое является основой для передачи информации и контроля информационных потоков. В общем виде оно звучит так: «Если А относится к В, а В относится к С, то и А относится к С». Например, если «10 больше 5, а 5 больше 1, то и 10 больше 1». На языке потоков отношение транзитивности выражается так: «Если существует поток от А к В и поток от В к С, то существует и поток от А к С». Поток, проходящий через несколько объектов, называется **составным потоком**, в приведенном примере составным будет поток от А к С. Верно также и то, что если на каком-то участке поток прерван, то и общий составной поток не существует, т.е., «если существует поток от А к В, но не существует потока от В к С, то не существует потока от А к С».

Свойство транзитивности потоков является основой для построения систем безопасности.

По свойству транзитивности потоков имеет место доступ субъекта X к объекту O_j через субъект S_i.

Во внешнем сегменте КС логично предположить существование субъекта, который управляется внешним злоумышленником (мы помним, что доступ к данным ЛС КС возможно получить только через программы!).

Целью удаленного злоумышленника является организация потоков от данных локальных объектов в своих интересах. В случае разделения КС на локальный и внешний сегменты множество всех потоков можно разделить на четыре группы конструкций:

1. Потоки между локальными субъектами и локальными объектами.
2. Потоки между локальными субъектами и удаленными объектами.
3. Потоки между удаленными субъектами и локальными объектами.
4. Потоки между удаленными субъектами и удаленными объектами.

Понимание, как сконструированы и как работают эти группы КС, очень важно, для того чтобы точно знать, как пользователь может получать информацию из «чужих» компьютерных систем, как он может обмениваться данными внутри своей замкнутой локальной системы, не имеющей выхода во внешние КС, как к его (может быть, конфиденциальным) данным может получить доступ удаленный субъект, в том числе злоумышленник, как можно обмениваться данными с дружественными локальными системами или с внешними КС. Скажем, для аналитика, работающего на фондовой бирже, компьютер которого подключен к внешним КС, очень важно непрерывно получать и обрабатывать информацию о состоянии котировок ценных бумаг на других фондовых биржах в том случае, конечно, если они открыты для доступа. Но в то же время ему важно, чтобы данные, находящиеся в его личных файлах, не стали известны третьим лицам. Это поясняет, почему так важна только что приведенная классификация.

Потоки между локальными субъектами и локальными объектами (первая конструкция) описывают работу пользователя на своем рабочем месте или внутри корпоративной сети. Он запускает программы и обрабатывает данные из своего ЛС КС. Потоки между локальными субъектами и удаленными объектами означают работу пользователя с внешним ресурсом при помощи своих собственных программ, размещенных на ЛС КС. Потоки между удаленными субъектами и локальными объектами означают, что удаленные субъекты пользуются вашими внутренними ресурсами. Это может быть санкционированное действие, а может быть и злоумышленное. Злоумышленное воздействие может быть пассивным – поток направлен вовне и тогда у вас, скорее всего, крадут информацию, а может быть активным – это означает, что вашу информацию с какими–то целями изменяют. Еще одно вероятное событие в этом случае – что в ваш ЛС КС внедряют постороннее программное обеспечение, которое будет красть информацию по некоторому событию или сигналу извне. Потоки между удаленными субъектами и удаленными объектами описывают «работу сторонних организаций». Эти потоки тоже могут представлять интерес.

Введем еще одно понятие порождения субъекта ***Create(***S_j***, Oi)–>***S_k – из объекта Oi порожден субъект S_k при активизирующем воздействии субъекта S_j. ***Create*** назовем операцией порождения субъектов (см. также ***рис. 2***).

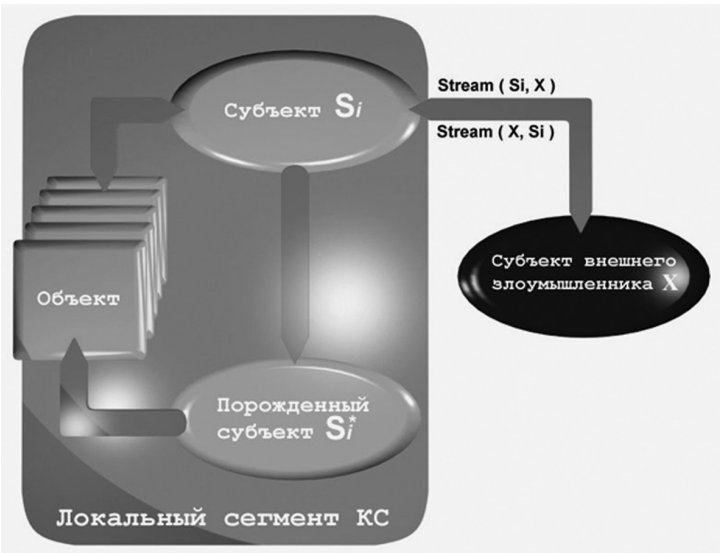


Рис. 1. Модель распределенной компьютерной системы

Операция порождения субъектов описывает запуск программ в рамках локального или внешнего сегмента КС. Если мы вспомним, как пользуемся различными программами (например, в операционной среде Windows мы «щелкаем мышкой» по изображению программы на рабочем столе), то обратим внимание на то, что первично программа существует в виде **объекта**, который называется **исполняемым модулем**. Этот объект мы подвергаем «запуску», при этом находящиеся в нем инструкции для процессора начинают исполняться, и объект становится субъектом! Но важно понимать еще, что новый субъект в КС не может образоваться сам, его запускает другой, уже активный субъект. В примере с Windows это управляющая программа Explorer. У субъекта–программы обязательно есть связанные с ним объекты, которые называются **ассоциированными** с ним. Это некоторые переменные и поля программы, содержание окон на экране и, наконец, сама последовательность действий программы, которой размещена в памяти компьютера. Ассоциированные объекты отражают состояние субъекта.

В терминах потоков рассмотрим межсубъектное взаимодействие между удаленным субъектом X и локальным субъектом S_i. В данном случае i-й субъект – просто одна из программ нашего компьютера, взаимодействующая с сетью. Целью данного взаимодействия является реализация потока между локальным объектом O_j и ассоциированным объектом O_x субъекта X, причем данный поток проходит через ассоциированные объекты локального субъекта S_i. На практике это означает, что мы обращаемся во внешний сегмент КС через свою программу S_i, которая связывается с внешней программой X, причем это взаимодействие происходит всегда – и в случае «добросовестного» сетевого обще-

ния, и в случае атак злоумышленника. Хорошо знакомый всем пример, это Internet Explorer, который чаще называют браузером. С другой стороны, мы видим внешний сегмент КС, внешнюю сеть, только тогда, когда нам «отвечает» на наши запросы со стороны внешнего мира некоторый субъект X. Кроме того, вполне существует возможность того, что из сети нам будет передан внешний объект Ov, который может быть запущен в рамках нашего ЛС КС и породить новый субъект в нашем «локальном мире».

Порождение нового субъекта может произойти из объекта, находящегося:

1. В локальном сегменте КС,
2. Во внешнем сегменте КС.

На ***рис. 1*** схематически представим рассматриваемое взаимодействие субъектов S_i и X.

Введем следующие обозначения: X – субъект внешнего сегмента КС, который инициирует поток через S_i – субъект, принадлежащий подмножеству субъектов локального сегмента КС, O_j – объект локального сегмента КС, S_i* – субъект локального сегмента, порожденный субъектом S_i. O_k – ассоциированный объект субъекта S_i. Объекты O_j и O_k на рис. 1 входят в множество объектов ЛС КС.

Рассмотрим следующую упрощенную модель работы «распределенной» компьютерной системы, которая состоит из двух компьютеров. На рассматриваемых компьютерах обязательно будет установлено телекоммуникационное программное обеспечение, обеспечивающее совместную работу прикладных программ и аппаратуры передачи данных для обмена информацией по каналу связи. Отметим, что передаваемая и принимаемая информация представляется в различных частях КС на различных уровнях (файлы, части файлов, пакеты). **Злоумышленника** полагаем в данном случае лицом, которое имеет доступ к каналу связи и располагает идентичным по отношению к нашему компьютеру комплексом программных и аппаратных средств. Работу как злоумышленника, так и легального пользователя можно представить как работу передающего либо принимающего компьютера, или как посылку (или прием) на наш компьютер некоторой информации. Если речь идет о злоумышленнике, то на нашем атакуемом компьютере работает некий субъект, который традиционно называется телекоммуникационным субъектом и либо входит в состав телекоммуникационного программного обеспечения на нашем компьютере, либо был прислан нам извне. Злоумышленные действия в рассматриваемом случае возможны двух основных видов:

• пассивное воздействие, связанное с чтением данных с атакуемого компьютера и транспортировкой их на компьютер злоумышленника;

• активное воздействие, связанное с присылкой на наш компьютер новых данных (например, новых файлов) и модификацией уже существующих файлов, в том числе и исполняемых.

Обобщим сказанное и сформулируем его на языке потоков. Обозначим потоки от ассоциированного объекта O_x субъекта X к ассоциированному объекту O_k субъекта S_i и наоборот: ***Stream***(X,O_x)–>O_k и ***Stream***(X,O_k)–>O_x. Предположим также, что свойства субъекта S_i таковы, что возможно существование потоков вида ***Stream***(S_i,O_j)–>O_k и ***Stream***(S_i,O_k)–>O_j.

Существует весьма важное отношение **транзитивности**, которое является основой для передачи информации и контроля информационных потоков. В общем виде оно звучит так: «Если А относится к В, а В относится к С, то и А относится к С». Например, если «10 больше 5, а 5 больше 1, то и 10 больше 1». На языке потоков отношение транзитивности выражается так: «Если существует поток от А к В и поток от В к С, то существует и поток от А к С». Поток, проходящий через несколько объектов, называется **составным потоком**, в приведенном примере составным будет поток от А к С. Верно также и то, что если на каком-то участке поток прерван, то и общий составной поток не существует, т.е., «если существует поток от А к В, но не существует потока от В к С, то не существует потока от А к С».

Свойство транзитивности потоков является основой для построения систем безопасности.

По свойству транзитивности потоков имеет место доступ субъекта X к объекту O_j через субъект S_i.

В локальном сегменте КС возможны также две основные ситуации, связанные с упомянутой выше возможностью порождения нового субъекта:

1. Доступ к объекту O_j со стороны субъекта S_i при управляющем воздействии субъекта X.

2. Порождение субъектом S_i из локального объекта нового субъекта S_i*, для которого существует поток ***Stream***(X,S_i^*).

Первая ситуация связана с тем, что субъект X может получить доступ к объектам нашего локального сегмента непосредственно через наш телекоммуникационный субъект, через который мы «видим» внешний сегмент КС. Вторая – с порождением нового субъекта и доступом через него.

Наличие канала связи между удаленным субъектом и локальными объектами позволяют говорить о задаче семантической защиты передаваемых данных, которую изучает **криптография**.

Подводя итоги, можно сформулировать несколько важных принципов работы КС. **Первый** – пользователь компьютерной системы взаимодействует с активными компонентами КС – субъектами (программами), а информацию черпает из пассивного компонента – объектов. **Второй** принцип состоит в том, что пользователь работает во внутреннем сегменте КС, общаясь с различными субъектами и пользователями внешнего сегмента КС и получая информацию также от объектов внешнего субъекта КС. **Третий** принцип – получение информации есть реализация различных потоков, многие из которых имеют сложную структуру, практически все потоки являются составными. **Четвертый** принцип – внешний сегмент КС не управляем и потенциально враждебен пользователю, он содержит массу источников информации с различными свойствами, которые (в первую очередь достоверность) необходимо тщательно проверять. Из внешнего сегмента могут прийти угрозы для данных размещенных в локальном сегменте КС. Угрозы возможны и при передаче информации от одного локального сегмента к другому. **Пятый** принцип – для результативной работы пользователя его компьютерная система должна быть наполнена субъектами, реализующими все необходимые ему функции, иными словами – перечень используемых программ должен быть функционально полным, но не избыточным, пос-

кольку через «лишние» субъекты возможна утечка информации, появление в системе «вирусов» и посторонних программ.

1.2. Методы обеспечения безопасности в компьютерной системе

Защищаемая компьютерная система может изменяться, дополняться новыми компонентами (субъектами, объектами, операциями субъектов над объектами). Точно так же каждый практический пользователь и конструктор защищенной КС должен ясно понимать, что конкуренты, которых на языке информационной безопасности называют злоумышленниками, также могут совершенствовать и разнообразить системы и способы несанкционированного доступа к чужим базам данных, могут намеренно засылать «компьютерные вирусы», искажающие информацию, осуществлять различного рода атаки и «взломы» хранилищ информации, которые могут содержать секретный материал, составляющий коммерческую тайну данной организации. Поэтому ясное понимание значение приведенных аксиом – одно из условий эффективности работы пользователя КС. Очевидно, что политика безопасности должна быть поддержана во времени, следовательно, в процесс изменения свойства защищаемой системы должны быть дополнены процедуры управления безопасностью.

Разумеется, компьютерная система не является живым существом. Но, как и многие технические изделия, она обладает своим жизненным циклом. Типовой **жизненный цикл защищенной компьютерной системы** состоит из следующих стадий:

1. Проектирование КС и проектирование политики безопасности;
2. Моделирование политики безопасности и анализ ее корректности, включающий установление адекватности политики безопасности целевой функции КС;
3. Реализация политики безопасности, разработка гарантий ее надежности и неуязвимости;
4. Эксплуатация защищенной системы.

Безопасность КС достаточно часто описывается в категориях «достоверность», «конфиденциальность», «целостность» и «доступность».

Свойство **достоверности** понимается как сохранение информацией своих свойств в любой момент времени, начиная с ввода в систему. Свойство **доступности** заключается в возможности пользования некоторым ресурсом КС и информацией в произвольный момент времени. При этом, разумеется, имеется в виду доступность информации для лиц, имеющих на это разрешение и соответствующие санкции руководителей организации. Это, в первую очередь, относится к лицам, для которых работа с данной КС входит в состав их служебных обязанностей. Свойство **целостности** (связанное со свойством достоверности) подразумевает неизменность свойств информации и ресурсов в любой момент времени. Свойство **конфиденциальности** понимается как недоступность информации или сервисных услуг КС для лиц, которые не имеют на это соответствующего разрешения или допуска, санкционированного руководством организации.

Иногда выделяют также свойство актуальности информации, связанное со свойством доступности, которое заключается в том, что база

данных и знаний, являющаяся важнейшим компонентом данной КС, содержит информацию и знания, которые нужны для выполнения тех или иных служебных функций сотрудников. Устаревшие или потерявшие значение данные становятся неактуальными и либо хранятся в архивах в системах компьютерной памяти, либо подлежат удалению из КС.

Рассмотрим теперь качественные характеристики и классификацию различных угроз в компьютерной системе.

По цели реализации угрозы – нарушение конфиденциальности, целостности, доступности.

По принципу и типу воздействия – с использованием физического доступа к элементам компьютерной системы (локальное воздействие) или удаленное воздействие. Пассивное воздействие (с использованием телекоммуникационных каналов передачи информации, как это было рассмотрено в предыдущем параграфе, либо с использованием технических каналов утечки информации) и активное воздействие (с использованием каналов удаленного воздействия). **Канал утечки информации** – состоит из источника информации, канала распространения информации от источника и средства извлечения информации из этого канала.

Канал утечки информации связан с пассивным воздействием злоумышленника на объект, т. е. вектор переноса информации направлен от объекта к злоумышленнику, расположенному вовне или внутри системы. Через канал утечки реализуется **угроза конфиденциальности** информации в КС.

Принято выделять следующие **каналы утечки**:

1. Электромагнитный канал. Причиной его возникновения является электромагнитное поле, связанное с протеканием электрического тока в технических средствах КС. Электромагнитное поле может индуцировать токи в близко расположенных проводных линиях (наводки). Например, работа мониторов с лучом развертки сопряжена с излучением сигнала, полностью повторяющего изображения на компьютерном мониторе, на сотни метров, а телефонные провода, выходящие за пределы здания могут унести информацию о других работающих в помещении приборах, например, сканерах. Интересно, что нить накаливания обычной электрической лампочки является очень чувствительным микрофоном, она модулирует напряжение в присоединенных к ней проводах и разговор в помещении может быть прослушан достаточно далеко от него и без установки в этом помещении микрофонов.

Электромагнитный канал в свою очередь делится на следующие каналы:

1.1. Радиоканал (высокочастотное излучение).

1.2. Низкочастотный канал.

1.3. Сетевой канал (наводки на сеть электропитания).

1.4. Канал заземления (наводки на провода заземления).

1.5. Линейный канал (наводки на линии связи между компьютерами).

2. Акустический (виброакустический) канал, который связан с распространением звуковых волн в воздухе или упругих колебаний в других средах, возникающих при работе устройств отображения информации КС.

3. Визуальный канал (визуальное восприятие информации на различных носителях).

Более общим понятием по сравнению с каналом утечки является **канал воздействия на КС**. Он может включать: изменение компонент КС – активное воздействие – угроза свойству целостности. Несанкционированный доступ в КС может иметь как пассивный, так и активный характер, поэтому его корректнее отнести к воздействию на КС.

По причине реализации угрозы:

• несоответствие (неадекватность) политики безопасности реальным условиям жизненного цикла КС;

• ошибки управления системой безопасности;

• ошибки проектирования системы гарантий;

• ошибки программной реализации;

• недостоверная работа вычислительной базы (ошибки программ, неисправности и сбои оборудования).

Перечисленные угрозы на самом деле являются последствиями недооценки влияния внешних и внутренних факторов работы КС. Например, неисправность в оборудовании компьютера может привести к неработоспособности жесткого диска, а сдача в ремонт этого диска – к утечке имеющихся на нем конфиденциальных сведений. Эта ситуация может быть вызвана искусственно – например инициирован бросок напряжения, который и вызовет поломку.

Защита информации в КС – комплекс организационных, организационно–технических и технических мер, предотвращающих или снижающих возможность образования каналов утечки информации и/или каналов воздействия на КС. *Организационные меры защиты* – меры общего характера, затрудняющие доступ к ценной информации злоумышленников вне зависимости от особенностей способа обработки информации и каналов утечки. *Организационно–технические меры защиты* – меры, связанные со спецификой канала утечки (канала воздействия) и метода обработки информации, но не требующие для своей реализации нестандартных приемов, оборудования или программных средств. *Технические (программно–технические) меры защиты* – меры, жестко связанные с особенностями канала утечки (воздействия) и требующие для своей реализации специальных приемов, оборудования или программных средств. Приведенная классификация весьма важна для практической работы. Дело в том, что технические и программно-технические меры защиты часто не видны пользователям и реализуются профессиональными специалистами в области защиты информации на этапе проектирования и монтажа КС. А вот организационные и организационно-технические меры напрямую связаны с повседневной работой защищенной КС. В качестве примера можно привести требование не выносить из здания, где размещена корпоративная аналитическая КС, съемных носителей информации (флеш-памятей), дабы исключить возможность преднамеренной передачи информации или съема ее с менее защищенных домашних компьютеров, подключенных к Интернет. Другая мера – запретить внос на территорию мобильных телефонов, которые могут порождать акустический канал утечки. Еще один показательный пример – экраны компьютерных мониторов не должны быть развернуты к окнам не только потому, что это вредно для зрения, но и по причине реализации визуального канала утечки (содержание экранов может быть сфотографировано из окон соседних зданий). Понимание

наличия и специфики организационных и организационно-технических мер необходимо как для понимания рисков, так и для более спокойного отношения к требованиям служб режима.

1.3. Основные понятия криптографии

Чтобы понять необходимость использования криптографических методов защиты информации, обратимся снова к рис. 1. Мы видим, что между ЛС КС и субъектом X имеется поток информации, к которому могут иметь доступ другие субъекты внешнего сегмента компьютерной системы. Этот поток принадлежит внешнему сегменту и не никак контролируется пользователем ЛС КС. Например, отправив электронное письмо, мы абсолютно не будем уверены в том, что оно не прочитано во время его движения кем-то еще, а, получая доступ к некоторому сайту в Интернет, либо удаленному хранилищу данных, мы также не можем быть уверены, что в наш канал обмена данными не включен сторонний наблюдатель. Никакими мерами защиты внутри ЛС КС этот доступ предотвратить невозможно. Единственно возможное решение – изменение информации в этом потоке таким образом, чтобы даже ознакомление с ним не давало никаких сведений о содержании информации в этом потоке.

Дополнительно можно рассматривать случай, когда субъект, которому не разрешен доступ к нашим локальным базам данных (находящимся в рамках ЛС КС в нашем личном пользовании), пытается прочитать содержащуюся в них информацию, изменить ее, разрушить, внести намеренную путаницу. Чтобы воспользоваться криптографической защитой, нам необходимо, чтобы специалисты по информатике, которым поручено организовать такую защиту, пользовались основными понятиями криптографической защиты однозначно и употребляли их в одном и том же смысле. Вот эти понятия.

Шифр – совокупность алгоритмов или отображений открытой (общедоступной) информации, представленной в формализованном виде, в недоступном для восприятия шифрованном тексте (также представленным в формализованном виде). Шифр зависит от внешнего параметра (ключа), без знания которого невозможно шифрованную информацию преобразовать в открытую. Ключ должен быть задан или сконструирован таким образом, чтобы его нельзя было определить ни по шифрованной, ни по открытой информации.

Шифрование информации – взаимнооднозначное математическое (криптографическое) преобразование, зависящее от ключа (секретный параметр преобразования), которое ставит в соответствие блоку открытой информации, представленной в некотором цифровом представлении, блок шифрованной информации, также представленной в цифровом представлении. Термин шифрование объединяет в себе два процесса: **зашифрование** и **расшифрование** информации.

Шифратор – аппарат или программа, реализующая шифр. В современной литературе вводится понятие «средства криптографической защиты информации», которое включает в себя шифратор, но в целом является более широким.

Ключ – некоторый неизвестный параметр шифра, позволяющий выбрать для шифрования и расшифрования конкретное преобразование

из всего множества преобразований, составляющих шифр. Простая ассоциация – ключ от замка – во многом проясняет смысл термина. Есть много одинаковых качественно ключей, но лишь некоторые (или один) откроют замок.

Шифрование – процесс получения шифрованного текста, основанный на знании ключа.

Дешифрование – восстановление открытого текста или ключа по шифрованному тексту.

Злоумышленник – субъект (или физическое лицо), не знающий ключа или открытого текста и стремящийся получить его. Злоумышленник в криптографии является конкретизацией злоумышленника для КС – в данном случае это некто, пытающийся выполнить атаку на конфиденциальность и прочитать зашифрованные данные.

Понятие «злоумышленник» в криптографии тесно связано с понятием «твердого незнания» ключа. В соответствии с методологией, принятой в современной криптографии, надежность шифра определяется степенью безопасности используемых в ней ключей, поскольку все долговременные элементы криптографической системы (множество правил шифрования, его механизм) рано или поздно станут известными злоумышленнику. Этот принцип был сформулирован еще в конце XIX в. и получил название «правила Керкхоффса» (Kerckhoffs' desiderata). В литературе по криптографии еще используется термин «правило Кирхгофа». В современной криптографии считается также, что злоумышленник имеет возможность также произвольным образом изменять сообщения. Поэтому вводится еще один термин.

Подлинность – принадлежность сообщения конкретному автору и неизменность содержания сообщения.

Пусть имеется некоторое множество X. Назовем его исходным множеством или множеством открытых текстов. Это могут быть, например, тексты, содержащие финансовые, торговые, политические сведения и передаваемые на некотором языке, с помощью двоичных векторов или целых чисел. Чаще всего X полагается множеством векторов длины n каждая его координата может принадлежать множеству M, которое называется **алфавитом**.

Зададим также множество Y – другое множество, которое назовем множеством зашифрованных текстов. Его также удобно представлять множеством векторов длины m, причем каждая координата вектора принадлежит множеству S. Это множество может совпадать с M или быть отличным от него, в первом случае при шифровании буквы текста отображаются в другие буквы, во втором случае для представления шифртекста используются символы другого алфавита – пример – «пляшущие человечки»).

Зададим также множество K – множество параметров преобразования или множество ключей. Данный параметр задает, какое именно преобразование будет использовано. Далее введем некоторое отображение E множества X на множество Y, зависящее от параметра k из множества K:

$E(k, x) = y$, где
k принадлежит K,
x принадлежит X,
y принадлежит Y

причем для любого x из X существует отображение D, также зависящее от параметра k:

D(E (k, x)) = x и x определяется однозначно. Данное условие говорит о том, что после расшифрования D должен получиться тот же самый исходный текст, а не какой–либо другой.

Отображение E назовем шифрующим отображением, а D – расшифровывающим отображением. Вместо «отображения» в литературе по криптографии используется также термин «преобразование».

В литературе по криптографии еще используются такие обозначения

Ek() – функция зашифрования на ключе k,

Dk() – функция расшифрования на ключе k.

Посмотрим, какие выводы мы можем сделать из этого определения:

1. Мощность множества (количество элементов множества) X всегда не больше мощности множества Y, поскольку в противном случае невозможно добиться однозначного отображения Y в X и получить однозначного расшифрования.

2. При любом фиксированном k отображение E взаимно однозначно.

Однако в сформулированном определении мы ничем не выделили шифры среди прочих преобразований информации. Поэтому сформулируем дополнительные свойства:

1. Множество K должно иметь мощность, достаточную, для того чтобы невозможно было осуществить перебор всех различных преобразований E.

2. По y=E(k,x) было бы очень трудоемко определить как x, так и k.

Дополнив основное определение данными свойствами можно ли утверждать, что шифр будет «хорошим»? По–видимому, нет – данные требования являются **необходимыми, но недостаточными** – мы не можем быть абсолютно уверенными в том, что по y определить k или x действительно «очень трудоемко».

Пример 1.

Пусть x_i , y_i и k_i принадлежат {0,1}.

Операция # задается таблицей

#	0	1
0	0	1
1	1	0

Легко видеть, что операция шифрования и расшифрования одинакова!

y=x#k – шифрование,

x=y#k – расшифрование

Пример 2.

Зададим шифрующее преобразование таблицей замены, в которой каждая буква с номером i заменяется другой буквой из того же алфавита

с номером p_i. Такая таблица в криптографии называется подстановкой. Если для каждой буквы текста подстановка не меняется, тогда такой шифр называется **шифром простой замены**.

Если букв в алфавите n, то ключ шифра в примере – подстановка степени n (их число n! – «эн факториал», n! = 1*2*3*…*n) и эта подстановка воздействует на каждую букву сообщения (которая берется из алфавита 1, 2, ... , n). Мощность множества подстановок степени n совпадает с множеством возможных ключей. Легко видеть, что это число крайне велико, например, для алфавита в 32 символа (русский язык) 32! (32 факториал) больше, чем 10^{80}. На основании этого Леонард Эйлер считал, что шифр простой замены является очень стойким (кстати, это считали и все те, кто в то время использовал такие шифры для переписки). Рассмотрим процесс шифрования.

Зададим подстановку p:

1 2 3 4 5

4 1 2 5 3

Исходное сообщение в алфавите {1,2,3,4,5}:

X = 11532451134

Зашифрованное сообщение (в том же алфавите):

Y = 44321534425

Мы видим, что наиболее часто встречается в зашифрованном тексте символ 4 – образ символа 1 после воздействия подстановки p. Если нам известны статистические закономерности исходного текста, то они сохранятся в шифртексте, но уже для образов соответствующих символов. К статистическим закономерностям текста относятся частоты встречаемости отдельных букв, сочетаний букв и отдельных слов. Например, искусственное слово «СЕНОВАЛИТР» означает расположение букв русского языка по убыванию частоты их встречаемости в «среднем» тексте. Не будем смеяться над гениальным математиком Эйлером – он будет прав, если нам не удастся набрать нужной статистики на шифрованном тексте. Почему это может произойти? Потому, что символов шифртекста будет немного. Для установления сколько–нибудь достоверных статистических оценок нужно не меньше символов, чем мощность алфавита. Увеличим алфавит – например, каждая «буква» файла будет последовательностью из 64 бит. Тогда на реальных объемах данных мы почти никогда не получим достоверной статистики. В этом состоит идея **блочного шифрования.**

Оценивая качества шифрующего отображения E, мы должны задаться конкретными условиями. В этих условиях желательно добиться выполнения условия 2 определения шифра, а именно, того, чтобы по y – шифртексту или по паре y и x – зашифрованному и открытому тексту найти параметр преобразования k было бы достаточно трудоемко.

Можно сформулировать несколько типовых классов задач, которые решает злоумышленник для нахождения открытых текстов или ключей:

• нахождение ключа только по шифртексту (а, значит, в силу обратимости отображения E, и открытого текста x по уравнению x=D(k, y).

• нахождение открытого текста по шифртексту без нахождения ключа шифрования.

• нахождение ключа k по паре x и y, связанной соотношением y=E(k,x) или нескольким таким парам.

• нахождение некоторых x для известных совокупностей пар (x, y) таких, что: y = E(k, x), т.е. для зашифрованных на одном ключе.

Первая задача решается, когда злоумышленник наблюдает за информацией, циркулирующей во внешнем сегменте КС и желает определить ключ, использованный для шифрования перехваченной им информации. Вторая – когда ключ не интересует злоумышленника, а интересует только переданный открытый текст, третья – когда известен какой-либо переданный открытый текст, соответствующий перехваченному шифрованному тексту, а четвертая – когда известно, что несколько сообщений зашифровано на одном и том же ключе.

Введем теперь интуитивно ясное понятие **стойкости шифра**, описывающее сложность преобразования E относительно задачи нахождения параметра преобразования. Итак, **стойкость шифрующего преобразования** – трудоемкость задачи нахождения параметра преобразования ключа k либо текста X при тех или иных условиях (например, тех, которые сформулированы выше).

Понятие **трудоемкости** связано с понятием алгоритма, т.е. полагается, что для нахождения ключа k строится некоторый алгоритм и стойкость оценивается его трудоемкостью. С другой стороны, ниже мы увидим, что иногда алгоритм нахождения ключа или исходного текста будет порождать несколько ключей или несколько осмысленных текстов, среди которых также придется выбирать (а иногда выбор просто невозможен). Рассмотрим **идеальный случай – шифр является абсолютно стойким, т.е. текст x невозможно найти никогда**. Клод Шеннон сформулировал условия для такого шифра. Эти условия, в общем, достаточно логичны – при перехвате некоторого зашифрованного текста злоумышленник не должен получить никакой информации о переданном открытом тексте.

Введем следующие обозначения:

p(x/y) – вероятность того, что зашифрован текст x при перехвате текста y,

p(y/x) – вероятность того, что при условии зашифрования x получен был именно y – или суммарная вероятность использования всех ключей, которые переводят x в y,

p(y) – вероятность получения криптограммы y,

p(x) – вероятность взятия для зашифрования текста x из множества возможных.

Для того, чтобы злоумышленник не получил никакой информации о ключе или об открытом тексте необходимо и достаточно, чтобы:

$$p(x) = p(x/y)$$

т.е. вероятность выбора текста для шифрования из множества возможных текстов не изменилась бы при получении криптограммы, соответствующей данному тексту.

По формуле Байеса:

$$p(x/y) = \frac{p(x)p(y/x)}{p(y)}$$

Из равенства ***p(x) = p(x/y)*** следует ***p(y) = p(y/x),*** т.е. суммарная вероятность всех ключей, переводящих x в y должна быть равна вероятности получения криптограммы y и не должна зависеть от x.

Из этого равенства следуют также два важных следствия:

• число всевозможных ключей не должно быть меньше числа сообщений,

• для каждого y должен найтись ключ k, который переводит любой x в данный y (так называемое условие криптографической транзитивности), в противном случае получив конкретный шифртекст y, можно будет исключить из рассмотрения некоторые ключи или открытые тексты.

Эти условия являются необходимыми, т.е. невыполнение хотя бы одного из них делает шифр не абсолютно стойким.

Пусть x_i, y_i, k_i принадлежат {0,1}.

y=x#k – шифрование,

x=y#k – расшифрование.

Пусть рассматриваются сообщения длины l, все меньшие сообщения дополняются до длины l хаотической информацией.

Ключ K – двоичный вектор длины L и он выбирается случайно равновероятно из множества возможных векторов длины L.

Такая система шифрования будет **абсолютно стойкой по Шеннону.**

Можно возразить: если перебрать все ключи длиной l. Ведь мы же получим нужное сообщение. Безусловно, но оно будет далеко не единственным осмысленным сообщением среди полученных перебором ключей.

До сих пор мы рассматривали системы шифрования, основанные на ключе, который знают и отправитель, и получатель сообщения.

Чтобы отправитель и получатель узнали свой ключ, его нужно им **доставить**, причем так, чтобы сохранить его в тайне от всех других. Кроме того, **очень желательно** для каждого сообщения использовать **новый ключ**.

Для практических пользователей и разработчиков весьма важно знать, что используемый алгоритм шифрования должен быть приближающимся к абсолютно стойкому, ключ для шифрования необходимо держать в тайне и вырабатывать новый ключ для шифрования каждого нового сообщения.

Очень привлекательным со всех точек зрения было бы вырабатывать ключ для каждого сообщения – однако понятно, что детерминированные алгоритмы для этого не годятся – злоумышленник может прогнозировать ключи. Но если мы будем вырабатывать ключи каждый раз случайно, то как оповестить об этом своего корреспондента?

Решение проблемы – в **односторонних** (однонаправленных) функциях. Функцию y=f(x) назовем односторонней, если вычисление y по x имеет малую **трудоемкость**, а вычисление x по y – высокую.

Односторонние функции – это чаще всего возведение в степень (обратная функция – логарифмирование). Надо отметить, что полагаемая из общих соображений «односторонность» функции часто впоследствии не подтверждается. Это связано либо с появлением новых математических методов, либо с появлением мощной вычислительной техники. Кроме того, необратимость функции часто сильно зависит от параметров этой функции.

Итак, попробуем использовать односторонние функции при выработке ключей. Например, имеются две функции f(a,x) и g(a,x). Отправитель А сообщения выбирает или формирует случайное значение x, вычисляет t=f(a,x), получатель В тоже вырабатывает случайное значение y, вычисляет r=g(a,y). Далее А получает r, а В – t.

Будем предполагать, что f(r,x)=f(g(a,y),x) совпадает с g(t,y)=g(f(a,x),y) (иначе у корреспондентов не получится одно и то же значение ключа).

С другой стороны важно чтобы сами функции f и g были легко вычислимы, а также равенство ***g(f(a,x),y)=f(g(a,y),x)*** выполнялось бы для любых x и y и хотя бы для некоторых a. А вот обратное должно быть неверно – по значению f(a,x) и g(a,x) без знания x и y сложно было бы вычислить K.

Именно такой метод был предложен в работе Диффи и Хеллмана[2], он получил название «открытое распределение ключей». Данный метод описывает лишь протокол выработки ключа для произвольного алгоритма шифрования.

Система Диффи–Хеллмана на основе дискретной экспоненты.

Задано простое число P и некоторое число a < P.

Пусть имеется два корреспондента. Каждый из них вырабатывает случайное число x и y соответственно, а затем вычисляет

$r = a^x \pmod P$ – первый корреспондент,

$t = a^y \pmod P$ – второй корреспондент.

В данном случае функция $f(a,x) = a^x \pmod P$.

Затем корреспонденты обмениваются этими числами и вырабатывают общий ключ K.

$K= (a^x \pmod P)^y \pmod P = (a^y \pmod P)^x \pmod P$

Злоумышленник будет располагать только числами t и r, и для нахождения x и y он должен будет проводить операцию **логарифмирования в простом поле** и, следовательно, трудоемкостью данной операции и будет определяться стойкость.

Нельзя ли построить алгоритм шифрования так, чтобы отправитель сообщения мог лишь зашифровать его, а расшифровать мог только получатель? Для того, чтобы это стало возможным, необходимо следующее:

• отправитель и получатель сообщения должны пользоваться разными ключами или разными алгоритмами,

• ключи отправителя и получателя должны быть связаны однозначным соотношением,

• по одному из ключей было бы крайне сложно определить другой ключ.

Оказывается, так построить схему шифрования возможно. Для этого необходимо использовать однонаправленные алгоритмы. Такой способ шифрования называется «открытым шифрованием», поскольку в данном случае однонаправленная функция существенно участвует в самом процессе шифрования, а не только при выработке ключа. Таков, например, алгоритм шифрования RSA.

2. Диффи У., Хеллмен Э. Новое направление в криптографии //ТИИЭР, IT-22, 1976.

1.4. Электронная цифровая подпись

В современном сетевом обществе при передаче важных экономических, финансовых и т. д. документов чрезвычайно важно, чтобы получатели были уверены, что полученные документы и информация – подлинные, а не поддельные, что они содержат достоверную информацию, и что подпись отправителя также не поддельная. Иначе и действия менеджера, и обосновывающий их системный анализ могут оказаться не только ошибочными, но даже вредными. На языке современной информатики эта задача может быть сформулирована следующим образом. Необходимо обеспечить передачу информации от одного пользователя к другому, чтобы получатель мог убедиться в том, что:

• полученная информация **тождественна переданной** и **не исказилась** в процессе ее доставки;

• **автором** информации является именно **тот пользователь**, который ее отправил.

Иногда также рассматривают задачу **контроля времени** отправки сообщения – т.е., что сообщение было отправлено или положено в хранилище не позже некоторого момента.

Эта задача решается в механизмах электронной цифровой подписи под сообщениями, циркулирующими в КС.

Электронная цифровая подпись – дополнительные данные, добавляемые к передаваемому блоку данных, полученные в результате криптографического преобразования, зависящего от секретного ключа и блока данных, которые позволяют получателю данных удостовериться в целостности блока данных и подлинности источника данных, а также обеспечить защиту от всевозможных подлогов и подделок.

Проверка электронной цифровой подписи (ЭЦП) под блоком открытой информации производится с помощью криптографического преобразования и открытого ключа, соответствующего секретному ключу, участвовавшего в процессе установки ЭЦП. Такая подпись обеспечивает целостность сообщений (документов), передаваемых по телекоммуникационным каналам общего пользования в компьютерных системах различного назначения, с гарантированной идентификацией ее автора (лица, подписавшего документ). Электронная цифровая подпись позволяет заменить при безбумажном документообороте традиционные печать и подпись. При построении цифровой подписи вместо обычной связи между печатью или рукописной (собственноручной) подписью и листом бумаги выступает сложная математическая зависимость между электронным документом, секретным и открытым ключами. Неодинаковость процедур выработки и проверки ЭЦП, а также разность ключей для выполнения этих процедур позволяет в данном случае говорить об ассиметричных (несимметричных) криптографических алгоритмах или преобразованиях.

Практическая невозможность подделки электронной цифровой подписи опирается на очень большой объем определенных математических вычислений. При этом проставление электронной цифровой подписи под документом не меняет самого документа. Подпись только дает возможность проверить подлинность и авторство полученной информации.

Секретный ключ – криптографический ключ, который хранится пользователем системы в тайне. Он используется для формирования электронной цифровой подписи. Часто вместо термина «секретный ключ» используют термин «закрытый ключ», подчеркивающий постоянное неизвлекаемое хранение этого ключа в некотором устройстве. А для подчеркивания принадлежности этого закрытого ключа конкретному владельцу часто используется термин «личный закрытый ключ».

Основные компоненты ЭЦП: хеш–функция (алгоритм «сворачивания» текста в более короткий) и некоторый криптографический алгоритм (шифрующее преобразование), который воздействует на хеш–значение.

Рассмотрим теперь эти компоненты. Для начала обратимся к важному вопросу хеширования данных.

Наша задача состоит в том, чтобы при помощи некоторой не слишком сложно вычисляемой процедуры «свернуть» любой текст любой длины в некоторую относительно короткую информацию фиксированной длины. Простым примером свертки (правда, абсолютно не годящимся на роль хеш-функции) является суммирование букв текста. В результате мы получим одну букву, которая в случае искажения текста покажет нам, что мы где-то ошиблись при его наборе. Предположим, что имеется некоторый текст F – последовательность букв заданного алфавита и некоторый алгоритм A, преобразующий F в некоторый текст M меньшей длины.

Этот алгоритм должен быть таков, что при случайном равновероятном выборе двух текстов из множества возможных соответствующие им тексты M с высокой вероятностью различны. Тогда проверка целостности данных строится так:

- после пересылки или хранения текста F получен некоторый текст T,
- рассматриваем текст T, полагая, что текст F был изменен),
- по известному алгоритму A строим K = A(T),
- сравниваем M, заранее вычисленное как M = A(F) с K.

При совпадении считаем текст неизменным. Алгоритм A называют хеш–функцией, а число M – хеш–значением. Чрезвычайно важным является в данном случае выполнение следующих условий:

- по известному числу M = A(F) очень трудоемким должно быть нахождение другого текста G не равного F, такого, что M=A(G) (еще говорят, что задача компенсации хеш–значения (или задача построения коллизий) очень трудоемка),
- число M должно быть недоступно для изменения.

Поясним смысл этих условий. Пусть злоумышленник изменил текст F. Тогда, вообще говоря, хеш–значение M для данного текста изменится. Если злоумышленнику доступно число M, то он может по известному алгоритму A вычислить новое хеш–значение для измененного им текста и заместить им исходное. Именно с этой целью хеш–значение подвергается шифрованию, как правило, с использованием системы открытого шифрования. С другой стороны, пусть само хеш–значение недоступно, тогда можно попытаться так построить измененный файл, чтобы хеш–значение его не изменилось (принципиальная возможность этого имеется, поскольку отображение, задаваемое алгоритмом A неоднозначно). Выбор хорошего хеш–алгоритма так же, как и построение

качественного шифра, является достаточно сложной задачей. Требования несовпадения с высокой вероятностью хеш–значений для разных файлов приходит в определенное противоречие с требованием трудоемкости их компенсации за счет подбора текста.

Необходимость наличия для выполнения криптографических операций двух ключей, один из которых держится в секрете, а другой принципиально общедоступен, приводит к реализации так называемой атаки «посредника». Для описания атаки воспользуемся следующими обозначениями:

P – открытый текст,

C – зашифрованный текст,

Ek() – функция зашифрования,

Dk() – функция расшифрования.

Om – открытый ключ абонента сети защищенной связи M,

Sm – секретный ключ соответственно.

Абонент A хочет направить абоненту B зашифрованное сообщение. Он производит преобразование, воспользовавшись открытым ключом абонента B из справочника, расположенного в общедоступном месте. Полученное сообщение C абонент B расшифровывает с использованием своего секретного ключа Sb.

При атаке «посредника» в сети связи появляется злоумышленник со своей ключевой парой Ox, Sx. Имея доступ к справочнику открытых ключей, злоумышленник осуществляет подмену ключа на свой собственный, после чего злоумышленник может свободно читать корреспонденцию, предназначенную другому получателю. Для того чтобы последний ничего не заподозрил, злоумышленник может сохранить у себя локальную копию первоначального ключа для формирования оригинального сообщения. В основе атаки лежит предположение о том, что данные в общедоступном справочнике могут быть изменены.

Ниже мы рассмотрим способы защиты от «атак посредника».

2. ПОНЯТИЕ ПОЛИТИКИ БЕЗОПАСНОСТИ. РЕАЛИЗАЦИЯ ПОЛИТИК БЕЗОПАСНОСТИ

2.1. Понятие политики безопасности

Рассматривая вопросы безопасности информации в компьютерных системах (будем далее использовать термин «компьютерная система» для обозначения объекта исследования, поскольку термин «автоматизированные системы», принятый в нормативных документах и научных работах по компьютерной безопасности, на сегодняшний день практически всегда описывает системы, содержащие компьютерную технику), можно говорить о наличии некоторых «желательных» состояний данных систем. Эти желательные состояния (описанные в терминах модели собственно компьютерной системы, например, в терминах субъектно-объектной модели – см. ниже) описывают «защищенность» системы. Понятие «защищенности» принципиально не отличается от любых других свойств технической системы, например, «надежной работы» и является для системы внешним, априорно заданным. Особенностью понятия «защищенность» является его тесная связь с понятиями «злоумышленник» (как обозначение внешней причины для вывода системы из состояния «защищенности») или «угроза» (понятие, обезличивающее причину вывода системы из защищенного состояния действий злоумышленника).

При рассмотрении понятия «злоумышленник» практически всегда выделяется объект его воздействия – часть системы, связанная с теми или иными действиями злоумышленника («объект атаки»). Следовательно, можно выделить три компоненты, связанные с нарушением безопасности системы: «злоумышленник» – внешний по отношению к системе источник нарушения свойства «безопасность», «объект атаки» – часть, принадлежащая системе, на которую злоумышленник производит воздействие, «канал воздействия» – среда переноса злоумышленного воздействия.

Интегральной характеристикой, описывающей свойства защищаемой системы, является **политика безопасности** – качественное (или качественно-количественное описание) свойств защищенности, выраженное в терминах, описывающих систему. Описание политики безопасности может включать или учитывать свойства злоумышленника и объекта атаки.

Приведем пример описания политики безопасности. Наиболее часто рассматриваются политики безопасности, связанные с понятием «доступ». Доступ – категория субъектно-объектной модели (субъекты – активная компонента системы (активность понимается как возможность выполнять операции над объектами – пассивной компонентой), описывающая процесс выполнения операций субъектов над объектами.

Описание политики безопасности включает:

1. множество возможных операций над объектами,

2. для каждой пары «субъект, объект» (S_i,O_j) назначение множества разрешенных операций, являющееся подмножеством всего множества возможных операций. Операции связаны обычно с целевой функций защищаемой системы (т.е. с категорией, описывающей назначение системы и решаемые задачи), например, операции «создание объекта», «удаление объекта», «перенос информации от произвольного объекта к предопределенному – чтение» и т.д.

Можно сформулировать две аксиомы защищенных компьютерных систем (КС):

Аксиома 1. В защищенной КС всегда присутствует активная компонента (субъект), выполняющая контроль операций субъектов над объектами.

Данная компонента фактически отвечает за **реализацию** некоторой **политики безопасности**.

Аксиома 2. Для выполнения в защищенной КС операций над объектами необходима дополнительная информация (и наличие содержащего ее объекта) о разрешенных и запрещенных операциях субъектов с объектами.

В данном случае мы оперируем качественными понятиями «контроль», «разрешенная и запрещенная операция», данные понятия будут раскрыты и проиллюстрированы ниже.

Дополнительная аксиома 3[1].

Все вопросы безопасности информации описываются доступами субъектов к объектам.

Важно заметить, что политика безопасности описывает в общем случае нестационарное состояние защищенности. Защищаемая система может изменяться, дополняться новыми компонентами (субъектами, объектами, операциями субъектов над объектами). Очевидно, что политика безопасности должна быть поддержана во времени, следовательно, в процесс изучения свойства защищаемой системы должны быть дополнены процедуры **управления безопасностью**.

С другой стороны, нестационарность защищаемой КС, а также вопросы реализации политики безопасности в конкретных конструкциях защищаемой системы (например, программирование контролирующего субъекта в командах конкретного процессора) предопределяет необходимость рассмотрения задачи **гарантирования заданной политики безопасности**.

Итак, резюмируя, можно сказать, что компьютерная безопасность решает четыре класса взаимосвязанных задач:

1. Формулирование и изучение политик безопасности.
2. Реализация политик безопасности.
3. Гарантирование заданной политики безопасности.
4. Управление безопасностью.

Типовой **жизненный цикл КС** состоит из следующих стадий:

1. Проектирование КС и проектирование политики безопасности.

2. Моделирование ПБ и анализ корректности ПБ, включающий установление адекватности политики безопасности и целевой функции КС.

3. Реализация ПБ и механизмов ее гарантирования, а также процедур и механизмов управления безопасностью.

1. А.А. Грушо, Е.Е.Тимонина. Теоретические основы защиты информации. М. Яхтсмен, 1996. – 192 с.

4. Эксплуатация защищенной системы.

Безопасность КС достаточно часто описывается в категориях «достоверность», «конфиденциальность», «целостность» и «доступность».

Свойство **достоверности** понимается как сохранение информацией своих семантических свойств в любой момент времени от момента ввода в систему. Свойство **доступности** понимается как возможность пользования некоторым ресурсом КС и информацией в произвольный момент времени. Свойство **целостности** (связанное со свойством достоверности) подразумевает неизменность свойств информации и ресурсов в любой момент времени от момента их порождения или ввода в систему. Свойство **конфиденциальности** понимается как недоступность информации или сервисов для пользователей, которым априорно не задана возможность использования указанных сервисов или информации (данных). Иногда выделяют также свойство *актуальности* информации, связанное со свойством доступности.

Перейдем теперь к рассмотрению модели КС, которая позволит формализовать описание защищенной КС.

2.2. Понятие доступа и монитора безопасности

В теории компьютерной безопасности, как мы уже отмечали выше, практически всегда рассматривается модель произвольной КС в виде конечного множества элементов. Указанное множество можно разделить на два подмножества: множество объектов и множества субъектов. Данное разделение основано на свойстве элемента «быть активным» или «получать управление» (применяется также термин «использовать ресурсы» или «пользоваться вычислительной мощностью»). Оно исторически сложилось на основе модели вычислительной системы, принадлежащей фон Нейману, согласно которой последовательность исполняемых инструкций (программа, соответствующая понятию «субъект») находится в единой среде с данными (соответствующими понятию «объект»).

Модели, связанные с реализацией ПБ, не учитывают возможности субъектов по изменению КС, которые могут привести к изменению ее свойств и, как предельный случай, к полной неприменимости той или иной модели к описанию отношений «субъект-объект» в измененной КС.

Этот факт не является недостатком политик безопасности. Достоверность работы механизмов реализации политики безопасности считается априорно заданной, поскольку в противном случае невозможна формализация и анализ моделей. Однако вопрос гарантий политики безопасности является ключевым как в теории, так и в практике.

Рассматривая активную роль субъектов в КС, необходимо упомянуть о ряде важнейших их свойств, на которых базируется излагаемая ниже модель.

Во-первых, необходимо заметить, что человек-пользователь воспринимает объекты и получает информацию о состоянии КС через субъекты, которыми он управляет и которые производят отображение информации в воспринимаемом человеком виде.

Во-вторых, угрозы компонентам КС (КС рассматривается в модели потоков или состояний) исходят от субъектов как активной компоненты, порождающей потоки и изменяющей состояние объектов в КС.

В-третьих, субъекты могут влиять друг на друга через изменяемые ими объекты, связанные с другими субъектами, порождая в конечном итоге в системе субъекты (или состояния системы), которые представляют угрозу для безопасности информации или для работоспособности самой системы.

Будем считать разделение КС на субъекты и объекты априорным. Будем считать также, что существует априорный безошибочный критерий различения субъектов и объектов в КС (по свойству активности). Кроме того, считаем в условиях всех утверждений, что декомпозиция КС на субъекты и объекты фиксирована.

Подчеркнем отличие понятия субъекта компьютерной системы от человека-пользователя следующим определением.

Пользователь – лицо (физическое лицо), аутентифицируемое некоторой информацией и управляющее субъектом компьютерной системы через органы управления ЭВМ. Пользователь КС является, таким образом, внешним фактором, управляющим состоянием субъектов. В связи с этим далее будем считать пользовательское управляющее воздействие таким, что свойства субъектов, сформулированные в ниже приводимых определениях, не зависят от него (т.е. свойства субъектов неизменяемы внешним управлением). Смысл данного условия состоит в предположении того факта, что пользователь, управляющий программой, не может через органы управления изменить ее свойств (условие неверно для систем типа компиляторов, средств разработки, отладчиков и др.).

Будем также полагать, что в любой дискретный момент времени множество субъектов КС не пусто (в противном случае соответствующие моменты времени исключаются из рассмотрения и рассматриваются отрезки с ненулевой мощностью множества субъектов).

Аксиома 4. Субъекты в КС могут быть порождены только активной компонентой (субъектами) из объектов.

Специфицируем механизм порождения новых субъектов следующим определением.

Определение 1. Объект O_i называется источником для субъекта S_m, если существует субъект S_j, в результате воздействия которого на объект O_i в компьютерной системе возникает субъект S_m.

Субъект S_j, порождающий новый субъект из объекта O_i в свою очередь называется активизирующим субъектом для субъекта S_m, S_m назовем порожденным объектом.

Введем обозначение: ***Create(***S_j***,Oi)->***S_k – из объекта Oi порожден субъект S_k при активизирующем воздействии субъекта S_j. ***Create*** назовем операцией порождения субъектов (см. ***рис. 1***).

Операция ***Create*** задает отображение декартова произведения множеств субъектов и объектов на объединение множества субъектов с пустым множеством. Заметим также, что в КС действует дискретное время и фактически новый субъект S_k порождается в момент времени t+1 относительно момента t, в который произошло воздействие порождающего субъекта на объект-источник.

Очевидно, что операция порождения субъектов зависит как от свойств активизирующего субъекта, так и от содержания объекта-источника.

Считаем, что если ***Create(***S_j***,Oi)***-> NULL (конструкция NULL далее обозначает пустое множество), то порождение нового субъекта из объ-

екта Oi при активизирующем воздействии S_j невозможно. Так, практически во всех операционных средах существует понятие исполняемого файла – объекта, могущего быть источником для порождения субъекта. Например, для MS DOS файл edit.com является объектом-источником для порождения субъекта-программы текстового редактора, а порождающим субъектом является, как правило, командный интерпретатор shell (объект-источник – command.com).

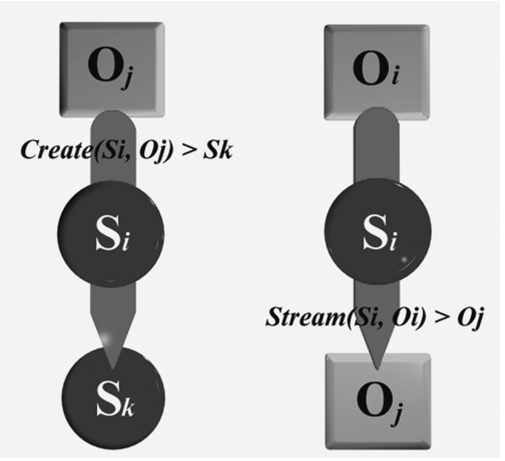


Рис. 1. Порождение субъекта и понятие потока

Из архитектуры фон Неймана следует также, что с любым субъектом связан (или ассоциирован) некоторый объект (объекты), отображающий его состояние (например, для активной программы (субъекта) ассоциированным объектом будет, например, содержание участка оперативной памяти с исполняемым кодом данной программы).

Определение 2. Объект Oi в момент времени t ассоциирован с субъектом S_m, если состояние объекта Oi повлияло на состояние субъекта в следующий момент времени (т.е. субъект S_m использует информацию, содержащуюся в объекте Oi).

Введем обозначение «множество объектов $\{O_m\}t$ ассоциировано с субъектом S_i в момент времени t»: $S_i(\{O_m\}t)$.

В данном случае определение не в полной мере является формально строгим, поскольку состояние субъекта описывается упорядоченной совокупностью ассоциированных с ним объектов, а ассоциированный объект выделяется по принципу влияния на состояние субъекта, т.е. в определении прослеживается некая рекурсия. С другой стороны, известны рекурсивные определения различных объектов (например, дерева). Зависимость от времени позволяет однозначно выделять ассоциированные объекты в том случае, если в начальный момент ассоциированный объект можно определить однозначно (как правило, это вектор исполняемого кода и начальные состояния ряда переменных программы).

Субъект в общем случае реализует некоторое отображение множества ассоциированных объектов в t-й момент времени на множество ассоциированных объектов в t+1-й момент времени. В связи с этим можно выделить ассоциированные объекты, изменение которых изменяет вид отображения ассоциированных объектов (объекты, содержащие, как правило, код программы – функционально ассоциированные) и ассоциированные объекты-данные (являющиеся аргументом операции, но не изменяющие вида отображения). Далее под ассоциированными объектами понимаются функционально ассоциированные объекты, в иных случаях делаются уточнения.

Следствие (к определению 2). В момент порождения субъекта S_m из объекта Oi он является ассоциированным объектом для субъекта S_m.

Необходимо заметить, что объект-источник может быть ассоциированным для активизирующего субъекта, тогда порождение является автономным (т.е. не зависящим от свойств остальных субъектов и объектов). Если же объект-источник является неассоциированным (внешним) для активизирующего субъекта, то порождение не является автономным и зависит от свойств объекта-источника.

Свойство субъекта «быть активным» реализуется и в возможности выполнения действия над объектами. При этом необходимо отметить, что пассивный статус объекта необходимо требует существования потоков информации от объекта к объекту (в противном случае невозможно говорить об изменении объектов), причем данный поток инициируется субъектом.

Определение 3. Потоком информации между объектом O_m и объектом O_j называется произвольная операция над объектом O_j, реализуемая в субъекте S_i и зависящая от O_m.

Заметим, что как O_j, так и O_m могут быть ассоциированными или неассоциированными объектами, а также «пустыми» объектами (NULL).

Обозначения: ***Stream(***S_i***,*** O_m***)->*** O_j – поток информации от объекта O_m к объекту O_j. При этом будем выделять источник (O_m) и получатель (приемник) потока (O_j). В определении подчеркнуто, что поток информации рассматривается не между субъектом и объектом, а между объектами, например, объектом и ассоциированными объектами субъекта (либо между двумя объектами), а активная роль субъекта выражается в реализации данного потока (это означает, что операция порождения потока локализована в субъекте и отображается состоянием его функционально ассоциированных объектов). Отметим, что операция ***Stream*** может создавать новый объект или уничтожать его.

На ***рис. 2*** схематически изображены различные виды потоков.

Далее будем для краткости говорить о потоке, подразумевая введенное понятие потока информации.

Понятие ассоциированных с субъектом объектов, как легко видеть из вышеизложенного, не является искусственной конструкцией. Корректно говорить о потоках информации можно лишь между одинаковыми сущностями, т.е. объектами. Кроме того, в ассоциированных объектах отображается текущее состояние субъекта. Отображениями ***Stream*** и ***Create*** описываются с точки зрения разделения на субъекты и объекты, все события (изменения субъектов и объектов), происходящие в КС.

Из данного определения также следует, что поток всегда инициируется (порождается) субъектом.

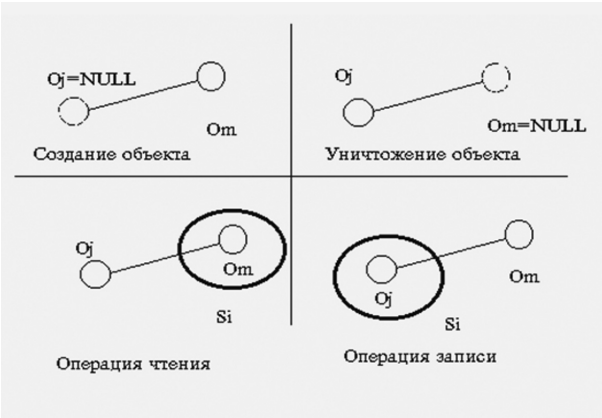


Рис. 2. Примеры потоков в КС

Определение 4. Доступом субъекта S_i к объекту O_j будем называть порождение потока информации между некоторым объектом (например, ассоциированным с субъектом объектами $S_i(O_m)$) и объектом O_j.

Выделим все множество потоков P для фиксированной декомпозиции КС на субъекты и объекты во все моменты времени (все множество потоков является объединением потоков по всем моментам дискретного времени) и произвольным образом разобьем его на два непересекающихся подмножества: N и L, P = N U L.

Обозначим:

N – подмножество потоков, характеризующее несанкционированный доступ,

L – подмножество потоков, характеризующих легальный доступ.

Дадим некоторые пояснения к разделению множеств L и N. Состояние «безопасности» подразумевает наличие и некоторого состояния «опасности» – нежелательных состояний какой-либо системы (в данном случае КС). Будем считать парные категории типа «опасный – безопасный» априорно заданным для КС и описываемыми политикой безопасности, а результатом применения политики безопасности к КС – разделение на множества «опасных» потоков N и множество «безопасных» L. Деление на L и N может описывать как свойство целостности (потоки из N нарушают целостность КС) или свойство конфиденциальности (потоки из N нарушают конфиденциальность КС), так и любое другое произвольное свойство.

Определение 5. Правила разграничения доступа субъектов к объектам есть формально описанные потоки, принадлежащие подмножеству L.

В предлагаемой субъектно-ориентированной модели не производится уточнений известных моделей политик безопасности (политика безопасности описывает только критерии разбиения на множества L и N), но формулируются условия корректного существования элементов КС, обеспечивающих реализацию той или иной политики безопасности. Поскольку критерий разбиения на множества L и N не связан со следу-

ющими далее утверждениями (постулируется лишь наличие субъекта, реализующего фильтрацию потоков), то можно говорить об инвариантности субъектно-ориентированной модели относительно любой принятой в КС политики безопасности (не противоречащей условиям утверждений).

Определение 6. Объекты O_i и O_j тождественны в момент времени t, если они совпадают как слова, записанные в одном языке.

Например, при представлении в виде байтовых последовательностей объекты O1= (o_{11}, o_{12}, ..., o_{1m}) и O_2=(o_{21}, o_{22}, ..., o_{2k}) одинаковы, если m=k и $o_{1i} = o_{2i}$ для всех i от 1 до k (o_{ij} – байты).

Для введения понятия тождественности субъектов условимся о наличии процедуры сортировки ассоциированных объектов, которая позволяет говорить о возможности попарного сравнения. На практике всегда существует алгоритм, обеспечивающий возможность попарного сравнения и зависящий от конкретной архитектуры КС. Например, достаточно легко выделить и попарно сравнивать, например, участки оперативной памяти, отвечающие коду программ (отличающиеся абсолютным адресом загрузки в оперативную память) или содержанию переменных и массивов.

Определение 7. Субъекты S_i и S_j тождественны в момент времени t, если попарно тождественны все ассоциированные с ними объекты.

Следствие (из определений 6 и 7). Порожденные субъекты тождественны, если тождественны порождающие субъекты и объекты-источники.

Верность данного следствия вытекает из тождества функционально ассоциированных объектов в порождающих субъектах, которые отвечают за порождение нового субъекта, а также из тождества аргументов (ассоциированных объектов-данных), которые отвечают объектам-источникам.

Для разделения всего множества потоков в КС на подмножества L и N необходимо существование активной компоненты (субъекта), который:

- активизировался бы при возникновении любого потока,
- производил бы фильтрацию потоков в соответствии с принадлежностью множествам L или N.

Заметим, что если существует Stream(S_i, O_j)->O_m и существует Stream (S_k, O_m)->O_l, то существует и Stream ((S_i, S_k), O_j)->O_l, т.е. отношение «между объектами существует поток» является транзитивным (относительно пары субъектов). Именно в этом смысле будем говорить об участии субъекта (S_k) в потоке (если O_m является ассоциированным объектом субъекта, не тождественного S_i). Введем несколько определений.

Определение 8. Монитор обращений (МО) – субъект, активизирующийся при возникновении потока от любого субъекта к любому объекту.

Можно выделить два вида МО:

Индикаторный МО – устанавливающий только факт обращения субъекта к объекту.

Содержательный МО – субъект, функционирующий таким образом, что при возникновении потока от ассоциированного объекта O_m любого субъекта S_i (S_i(O_m)) к объекту O_j и обратно существует ассоциированный с МО объект O_mo (в данном случае речь идет об ассоциированных объ-

ектах-данных), тождественный объекту O_m или $S_i(O_m)$. Содержательный МО полностью участвует в потоке от субъекта к объекту (в том смысле, что информация проходит через его ассоциированные объекты-данные и существует тождественное отображение объекта на какой-либо ассоциированный объект МО).

Теперь сформулируем понятие монитора безопасности (в литературе также применяется понятие монитора ссылок). Это понятие связано с упоминаемой выше задачей фильтрации потоков. Поскольку целью является обеспечение безопасности КС, то и целевая функция монитора – фильтрация с целью обеспечения безопасности (отметим еще раз, что разделение на N и L задано априорно).

Определение 9. Монитор безопасности объектов (МБО) – монитор обращений, который разрешает поток, принадлежащий только множеству легального доступа L. Разрешение потока в данном случае понимается как выполнение операции над объектом-получателем потока, а запрещение – как невыполнение (т.е. неизменность объекта-получателя потока).

Монитор безопасности объектов фактически является механизмом реализации политики безопасности в КС. Обратимся теперь к основным моделям работы МБО.

2.3. Описание типовых политик безопасности

Постулируя наличие в КС субъекта, реализующего политику безопасности, рассмотрим описания (на уровне моделей) некоторых известных политик безопасности (описания политик безопасности приводятся с существенным использованием[2]).

2.3.1. Модели на основе дискретных компонент

В основе этих моделей лежит следующая идея: система защиты (в смысле описания алгоритма работы субъекта, обеспечивающего выполнение политики безопасности) представляется в виде некоторого декартова произведения множеств, составными частями которых являются элементы системы защиты. При этом в качестве математического аппарата для описания и анализа модели выбирается аппарат дискретной математики.

Результатом применения политики безопасности, задаваемой той или иной моделью к конкретному объекту защиты КС, является разрешение выполнения операции над объектом защиты, либо запрещение.

2.3.1.1. Модель АДЕПТ-50

Одна из первых моделей безопасности[3]. Данная модель рассматривает только объекты, которые типизированы на 4 группы объектов безопасности*: пользователи (Users – u), задания(Jobs – j), терминалы(Terminal – t) и файлы(*File – *f).* Каждый объект безопасности

2. Биктимиров М.Р., Щербаков А.Ю. Избранные главы компьютерной безопасности. – Казань: Изд-во казанского матем. общества, 2004. – 372 с.

3. Мафтик С. Механизмы защиты в сетях ЭВМ. /Пер. с англ. – М: Мир, 1993. – 216 с., ил.

описывается вектором (A, C, F, M), включающим следующие параметры безопасности.

Компетенция A – элемент из набора упорядоченных универсальных положений о безопасности, включающих априорно заданные возможные в КС характеристики объекта безопасности, например, категория конфиденциальности объекта: НЕСЕКРЕТНО, КОНФИДЕНЦИАЛЬНО, СЕКРЕТНО, СОВЕРШЕННО СЕКРЕТНО.

Категория C – Рубрикатор (тематическая классификация). Рубрики не зависят от уровня компетенции. Пример набора рубрик: ФИНАНСОВЫЙ, ПОЛИТИЧЕСКИЙ, БАНКОВСКИЙ.

Полномочия F – перечень пользователей, имеющих право на доступ к данному объекту.

Режим M – набор видов доступа, разрешенных для определенного объекта или осуществляемых объектом. Пример: ЧИТАТЬ ДАННЫЕ, ЗАПИСЫВАТЬ ДАННЫЕ, ИСПОЛНИТЬ ПРОГРАММУ (исполнение программ понимается как порождение активной компоненты из некоторого объекта – как правило, исполняемого файла).

Обозначим U={u} множество всех пользователей, известных системе (зарегистрированных пользователей), F(i) – набор всех пользователей, имеющих право использовать объект i, то для модели формулируются следующие правила:

1. Пользователь u получает доступ к системе $\Leftrightarrow u \in U$, где U – набор всех известных системе пользователей (правило легального пользователя).

2. Пользователь u получает доступ к терминалу t $\Leftrightarrow u \in F(t)$ – пользователь u имеет право использовать терминал t.

3. Пользователь u получает доступ к файлу j $\Leftrightarrow A(j) \geq A(f)$, $C(j) \supseteq C(f)$, $M(j) \supseteq M(f)$ и $u \in F(f)$, т.е. только в случае:

• привилегии выполняемой программы (по компетенции, категории и режиму одновременно) выше привилегий файла или равны им;

• пользователь является членом F(f).

Четырехмерный вектор, полученный на основе прав задания (субъекта), а не прав пользователя, используется в модели для управления доступом. Такой подход обеспечивает контроль права на доступ над множеством программ и данных, файлов, пользователей и терминалов.

Например, наивысшим полномочием доступа к файлу для пользователя «СЕКРЕТНО», выполняющего задание с «КОНФИДЕНЦИАЛЬНОГО» терминала будет «КОНФИДЕНЦИАЛЬНО».

2.3.1.2. Пятимерное пространство безопасности Хартстона

В данной модели используется пятимерное пространство безопасности для моделирования процессов установления полномочий и организации доступа на их основании.

Модель безопасности описывается пятью множествами:

A – установленных полномочий;
U – пользователей;
E – операций;
R – ресурсов;
S – состояний;

Область безопасности будет представлена декартовым произведением: A×U×E×R×S.

Доступ рассматривается как ряд запросов, осуществляемых пользователями u для осуществления операции e над ресурсами R, в то время, когда система находится в состоянии s. Например, запрос q на доступ представляется четырехмерным кортежем q=(u, e, R, s), u∈ U, e∈ E, s∈ S, r⊆R. Величины u и s задаются системой в фиксированном виде. Таким образом, запрос на доступ – подпространство четырехмерной проекции пространства безопасности. Запросы получают право на доступ в том случае, когда они полностью заключены в соответствующие подпространства.

Процесс организации доступа можно описать следующим образом.

Для запроса q, где q(u, e, R, s), набора U' определенных групп пользователей, набора R' определенных единиц ресурсов и набора P правильных (установленных) полномочий, процесс организации доступа будет состоять из следующих процедур:

1. Вызвать все вспомогательные программы, необходимые для «предварительного принятия решений».

2. Определить из U те группы пользователей, к которым принадлежит u. Затем выбрать из P спецификации полномочий, которым соответствуют выделенные группы пользователей. Этот набор полномочий F(u) определяет привилегию пользователя u.

3. Определить из P набор F(e) полномочий, которые устанавливают e как основную операцию. Этот набор называется привилегией операции e.

4. Определить из P набор F(R)(привилегию единичного ресурса R) – полномочия, которые определяют подмножество набора ресурсов из R', имеющего общие элементы с запрашиваемой единицей ресурса R.

Полномочия, которые являются общими для трех привилегий в процедурах 2, 3, 4 образуют D(q), так называемый домен полномочий для запроса

q: D(q)=F(u)∩F(e)F(R).

5. Удостовериться, что запрашиваемый ресурс R полностью включается в D(q), т.е. каждый элемент из R должен содержаться в некоторой единице ресурса, которая определена в домене полномочий D(q).

6. Осуществить разбиение набора D(q) на эквивалентные классы так, чтобы два полномочия попадали в эквивалентный класс тогда и только тогда, когда они специфицируют одну единицу ресурса. Для каждого такого класса логическая операция ИЛИ (или И) выполняется с условиями доступа элементов каждого класса.

Новый набор полномочий – один на каждую единицу ресурса, указанную в D(q), есть F(u, q) – фактическая привилегия пользователя u по отношению к запросу q.

7. Вычислить EAC – условие фактического доступа, соответствующего запросу q, осуществляя логическое И (или ИЛИ) над условиями доступа членов F(u, q). Эта операция И(или ИЛИ) выполняется над всеми единицами ресурсов, которые перекрывают единицу запрошенного ресурса.

8. Оценить EAC и принять решение о доступе:

• разрешить доступ к R, если R перекрывается;

• отказать в доступе в противном случае.

9. Вызвать все программы, необходимые для организации доступа после «принятия решения».

10. Выполнить все вспомогательные программы, вытекающие для каждого случая из условия 8.

12. Если решение о доступе было положительным – завершить физическую обработку.

Автор модели, Хартстон, отмечает, что приведенная последовательность шагов не всегда необходима в полном объеме. Например, в большинстве реализаций шаги 2 и 6 осуществляются во время регистрации пользователя в системе.

2.3.1.3. Резюме по моделям Адепт и Хартстона

К достоинствам дискретных моделей можно отнести относительно простой механизм реализации политики безопасности.

В качестве примера реализаций можно привести матрицу доступа, строки которой соответствуют субъектам системы, а столбцы – объектам; элементы матрицы характеризуют права доступа. К недостаткам относится статичность модели. Это означает что модель не учитывает динамику изменений состояния КС, не накладывает ограничений на состояния системы.

Для моделей, построенных на основе дискретной защиты, можно доказать следующую теорему (доказательство приведено в книге Л. Дж. Хоффмана «Современные методы защиты информации»[4]):

Не существует алгоритма, который может решить для произвольной системы дискретной защиты, является или нет заданная исходная конфигурация безопасной.

Данный факт обусловил процесс совершенствования моделей в сторону достижения некоторых доказательных свойств.

2.3.2. Модели на основе анализа угроз системе

Модели этого класса исследуют вероятность преодоления системы защиты за определенное время Т. Данный подход свойственен различным игровым задачам и в ряде случаев успешно применим для построения и анализа политик безопасности КС. К достоинствам этих моделей можно отнести числовую вероятностную оценку надежности идеальной реализации системы защиты. К недостаткам – изначально заложенное в модель допущение того, что система защиты может быть вскрыта. Задача анализа модели – минимизация вероятности преодоления системы защиты.

2.3.2.1. Игровая модель

Игровая модель системы защиты строится по следующему принципу. Разработчик создает первоначальный вариант системы защиты. После этого аналитик-«злоумышленник» начинает его преодолевать. Если к моменту времени Т, в который злоумышленник преодолел систему защиты, у разработчика нет нового варианта, система защиты пре-

4. Хоффман Л. Дж. Современные методы защиты информации. – М: Советское Радио. – 1980.

одолена. Если нет – процесс продолжается. Данная модель описывает процесс эволюции системы защиты в течение времени.

Как правило, данная модель реализуется на уровне «техническое задание – эскизный проект системы» и описывает рекурсивную процедуру совершенствования защитных механизмов. Основным недостатком предлагаемого подхода является тесная связь с мнением экспертов-аналитиков системы защиты (т.е. если аналитик не придумал способ обхода защиты, это не значит, что его объективно не существует).

2.3.2.2. Модель системы безопасности с полным перекрытием

Основным положением данной модели является тезис (аксиома) о том, что система, спроектированная на основании модели безопасности с полным перекрытием, должна иметь, по крайней мере, одно средство (субъект) для обеспечения безопасности на каждом возможном пути проникновения в систему.

В модели точно определяется каждая область, требующая защиты (объект защиты), оцениваются средства обеспечения безопасности с точки зрения их эффективности и их вклад в обеспечение безопасности во всей вычислительной системе. Считается, что несанкционированный доступ к каждому из набора защищаемых объектов сопряжен с некоторой величиной ущерба и этот ущерб может (или не может) быть определен количественно. Если ущерб не может быть определен количественно, то его полагают равным некоторой условной (как правило, средней) величине. Количественная категория «ущерба» может быть выражена в стоимостном (ценовом) эквиваленте, либо в терминах, описывающих системы (например, единицах времени, необходимых для достижения тех или иных характеристик КС после злоумышленного воздействия). Ущерб может быть связан с целевой функцией системы (например, для финансовой системы ущерб от конкретного злоумышленного действия есть сумма финансовых потерь участников системы).

С каждым объектом, требующим защиты, связывается некоторое множество действий, к которым может прибегнуть злоумышленник для получения несанкционированного доступа к объекту. Можно попытаться перечислить все потенциальные злоумышленные действия по отношению ко всем объектам безопасности для формирования набора угроз, направленных на нарушение безопасности. Основной характеристикой набора угроз является вероятность проявления каждого из злоумышленных действий. В любой реальной системе эти вероятности можно вычислить с ограниченной степенью точности.

2.3.2.3. Резюме по моделям анализа угроз

Основное преимущество метода моделирования состоит в возможности численного получения оценки степени надежности системы защиты информации. Данный метод не специфицирует непосредственно модель системы защиты информации, а может использоваться только в сочетании с другими типами моделей систем защиты информации.

При синтезе систем защиты в КС данный подход полезен тем, что позволяет минимизировать накладные расходы (ресурсы вычислительной системы) для реализации заданного уровня безопасности. Модели данного типа могут использоваться при анализе эффективности вне-

шних по отношению к защищаемой системе средств защиты информации. Для систем защиты, построенных на основании других моделей, данная модель может применяться для анализа эффективности процедур идентификации и аутентификации (см. вторую главу).

При анализе систем защиты информации модели данного типа позволяют оценить вероятность преодоления системы защиты и степень ущерба системе в случае преодоления системы защиты.

2.3.3. Модели конечных состояний

Для математической модели конечных состояний системы безопасности может быть доказана основная теорема безопасности:

Если начальное состояние системы безопасно, и все переходы системы из состояния в состояние не нарушают ограничений, сформулированных политикой безопасности, то любое состояние системы безопасно.

2.3.3.1. Модель Белла-ЛаПадула

Модель Белла-ЛаПадула описывает компьютерную систему абстрактно, без связи с ее реализацией. В модели определяется множество ограничений на систему, реализация которых будет гарантировать некоторые свойства потоков информации, связанных с безопасностью. Модель включает:

Субъекты S – активные сущности в системе;

Объекты O – пассивные сущности в системе.

Далее будем придерживаться именно такой модели представления КС.

Субъекты и объекты имеют уровни безопасности. Уровни безопасности являются некоторой характеристикой субъектов и объектов, связанной, как правило, с целевой функцией системы (наиболее часто уровень безопасности связан с уровнем конфиденциальности информации); ограничения на систему имеют форму аксиом, которые контролируют способы доступа субъектов к объектам. Эти аксиомы имеют вид:

1. запрет чтения информации субъектом с уровнем безопасности меньшим, чем у объекта, из которого информация читается (NO READ UP, «не читать ниже» – NRU);

2. запрет записи информации субъектом с уровнем безопасности большим, чем у объекта, в который информация записывается(NO WRITE DOWN, «не записывать выше» – NWD).

Данные ограничения описывают как безопасные состояния КС, связанные с существованием потоков только от нижних уровней безопасности к высшим.

В отличие от дискретной модели безопасности модель Белла-ЛаПадула не определяет прав доступа для каждого пользователя. Это означает, что разные субъекты могут иметь один уровень полномочий. Данная модель служит основой для мандатной (полномочной) системы безопасности.

При строгой реализации модели Белла-ЛаПадула возникает ряд проблем:

1) *Завышение уровня секретности* – вытекает из одноуровневой природы объектов. Это означает, что некоторой информации может

быть дан уровень секретности выше того, что она заслуживает. Пример – несекретный параграф в секретном сообщении.

2) *Запись вслепую* – это проблема, вытекающая из правила NRU. Рассмотрим ситуацию, когда субъект производит запись объекта с более высоким уровнем безопасности – эта операция не нарушает правила NWD. Однако после завершения операции субъект не может проверить правильность выполнения записи объекта путем выполнения контрольного чтения, так как это нарушает правило NRU.

3) *Удаленная запись* – это проблема, вытекающая из правила NWD. Рассмотрим ситуацию, когда некоторый субъект выполняет операцию чтения в распределенной системе. Такая операция возможна при выполнении правила NRU, так что уровень безопасности субъекта больше уровня безопасности объекта. Однако в распределенных системах операция чтения инициируется запросом с одной компоненты на другую, что можно рассматривать в данном случае как посылку сообщения от субъекта с более высоким уровнем безопасности к объекту с более низким уровнем, что является нарушением правила NWD.

4) *Привилегированные субъекты* – эта проблема связана с работой системного администратора. Функционирование системного администратора подразумевает выполнение в системе таких критических операций, как добавление и удаление пользователей, восстановление системы после аварий, установка программного обеспечения, устранение ошибок. Очевидно, что такие операции не вписываются в модель, что означает невозможность осуществления правильного администрирования без нарушения правил данной модели. Поэтому правила модели Белла-ЛаПадула нужно рассматривать для множества всех субъектов, исключая привилегированные.

2.3.3.2. Модель low-water-mark (LWM)

Данная модель является конкретизацией модели Белла-ЛаПадула и является примером того, что происходит, когда изменения уровня конфиденциальности объекта возможны. Политика безопасности задается в следующих предположениях:

• все субъекты КС классифицированы по уровню конфиденциальности fs(S) – уровень доступа субъекта, fc(S) – текущий уровень доступа субъектов, fo(O) – гриф (уровень) конфиденциальности объекта,

• поток информации (в данном случае рассматриваются потоки от объектов к ассоциированным объектам некоторого субъекта) разрешен только «снизу вверх» (в смысле повышения уровня конфиденциальности).

В рассматриваемой системе один объект, три операции с объектом, включающие запросы на доступ: read, write, reset.

Эти операции используются несколькими субъектами (процессами), имеющими фиксированные уровни секретности. Напомним формальное требование политики о том, что информация может двигаться только «снизу вверх». Поток информации возможен тогда и только тогда, когда реализуется доступ субъекта к объекту вида w или r.

Уровень объекта O в LWM может меняться: при команде write может снизиться, а при команде reset подняться следующим образом. По команде reset класс объекта поднимается и становится максимальным в линейном порядке. После этого все субъекты приобретают право w,

но право read имеют только субъекты, находящиеся на максимальном уровне. При команде write гриф объекта снижается до уровня субъекта, давшего команду w. При снижении уровня секретности объекта вся прежняя информация в объекте стирается, и записывается информация процессом, вызвавшим команду write. Право write имеет любой субъект, у которого fs(S) не больше fo(O), где fo(O) – текущий уровень объекта. Право reset имеет только тот субъект, который не имеет право write. Право read имеет любой субъект, для которого fg(S) больше, либо равно fo(0). Суммируем вышесказанное в следующей таблице.

Таблица 1. Операции в модели LWM

Операция	Организация доступа	Результат операции
Read	fs(S)не меньше fo(0)	Процесс S получает содержимое объекта 0 (поток от внешнего объекта к ассоциированным объектам субъекта).
Write	fs(S) не больше fo(0)	Уровень объекта становится равным уровню S. Данные от S записываются в 0.
Reset	fs(S) не больше fo(0)	Уровень объекта устанавливается выше всех уровней субъектов.

Откажемся от условия, что при команде write в случае снижения уровня объекта его содержимое стирается (например, оно становится равным нулевому слову). Ясно, что в этом случае возможна утечка информации. В самом деле, любой процесс нижнего уровня, запросив объект для записи, снижает гриф объекта, а получив доступ w, получает возможность r. Возникает канал утечки с понижением грифа. Данный пример показывает, что определение безопасного состояния в модели Белла-ЛаПадула неполное и смысл этой модели только в перекрытии каналов указанных видов. Если процесс снижения грифа объекта работает неправильно, то система перестает быть безопасной.

Рассмотрим пример, поясняющий политику безопасности в модели LWM.

Пусть в системе имеется три файла F1, F2 и F3. Гриф секретности их соответственно 1, 2, 3 (считаем, что конфиденциальность растет с ростом грифа). Субъект имеет категорию доступа 2. Зададим объект отображения текущего грифа m. При открытии файла (если операция возможна) в момент времени t+1 для выполнения операций Read или Write ему присваивается значение максимума из текущего его значения и грифа открываемого файла. $M_{t+1}=MAX(m_t, F_{it+1})$. Правила чтения и записи остаются без изменений.

Далее разрешение или запрещение операций происходит с учетом измененного грифа объекта m согласно правилам чтения и записи «вверх» и «вниз».

Пусть в нулевой момент времени m=1. Открывается файл F3 (гриф 3), открытие невозможно – категория субъекта равна 2 – ниже грифа файла, m не меняется.

Пусть открыт файл F2, следовательно, m=2. Чтение из файлов F1 и F2 возможно (их гриф не превосходит m). Запись возможна только в файл F2 (m больше, либо равно грифу файла). Тем самым потоки информации от файлов с высоким грифом к файлам с низким грифом невозможны.

В заключение необходимо сделать два замечания.

1. При управлении потоками описанным образом требуется, чтобы все объекты, участвующие в потоках, поддерживали метки конфиденциальности, в противном случае возникает проблема Clipboard (буфера обмена). В рассмотренном примере: необходимо записать информацию из F2 в буфер (Clipboard), закрыть файл, открыть F1, m примет значение 1, записать информацию из буфера в файл F_1 – реализован канал утечки информации с понижением грифа.

2. Максимальное значение m совпадает с категорией доступа субъекта.

2.3.3.3. Модель Лендвера

Определение 10

Классификация – обозначение, накладываемое на информацию, которое отражает ущерб, который может быть причинен несанкционированным доступом; включает уровни: TOP SECRET, SECRET и т.д.

Степень доверия пользователю – уровень благонадежности пользователя (априорно заданная характеристика).

Пользовательский идентификатор – строка символов, используемая для того, чтобы отметить пользователя системы.

Роль – работа пользователя в КС. Пользователь в данный момент всегда ассоциирован как минимум с одной ролью из нескольких, и он может менять роль в течение сессии. Для действий в данной роли пользователь должен быть уполномочен. Некоторые роли могут быть связаны только с одним пользователем в данный момент времени. С любой ролью связана способность выполнения определенных операций.

Объект – одноуровневый блок информации. Это минимальный блок информации в системе, который имеет классификацию (может быть раздельно поименован).

Контейнер – многоуровневая информационная структура. Имеет классификацию и может содержать Объекты (каждый со своей классификацией) и (или) другие Контейнеры. Различие между Объектом и Контейнером базируется на типе, а не на текущем содержимом.

Сущность – Объект или Контейнер.

Требование Степени Доверия Объектов – атрибут некоторых Контейнеров. Для некоторых Контейнеров важно требовать минимум Степени Доверия, т.е. Пользователь, не имеющий соответствующего уровня благонадежности, не может просматривать содержимое Контейнера. Такие Контейнеры помечаются соответствующим атрибутом.

Идентификатор – имя Сущности без ссылки на другие Сущности.

Ссылка на сущность *Прямая*, если это идентификатор Сущности.

Ссылка на сущность *Косвенная*, если это последовательность двух или более имен Сущностей (из которых только первая – идентификатор). Пример: «текущее сообщение, первый абзац, вторая строка».

Операция – функция, которая может быть применена к сущности. Некоторые Операции могут использовать более одной сущности (пример – Операция копирования).

Множество Доступа – множество троек (Пользовательский Идентификатор или Роль, Операция, Индекс операнда), которое связано с сущностью.

Система, реализующая модель безопасности Лендвера должна реализовывать приводимые ниже ограничения (ограничения запрещают пользователю операции, нарушающие эти ограничения). Часть этих ограничений должна реализовываться пользователями системы (правила безопасности), а часть – системой (ограничения безопасности).

Правила Безопасности.

A1. Администратор безопасности системы присваивает уровни доверия, классификацию устройств и правильные множества ролей.

A2. Пользователь вводит корректную классификацию, когда изменяет или вводит информацию.

A3. В пределах классификации пользователь классифицирует сообщения и определяет набор доступа для сущностей, которые он создает, так что только пользователь с требуемой благонадежностью может просматривать информацию.

A4. Пользователь должным образом контролирует информацию объектов, требующих благонадежности.

Ограничения безопасности.

B1. Авторизация – пользователь может запрашивать операции над сущностями, только если пользовательский идентификатор или текущая роль присутствуют в множестве доступа сущности вместе с этой операцией и с этим значением индекса, соответствующим позиции операнда, в которой сущность относят в требуемой операции.

B2. Классификационная иерархия – классификация контейнера всегда, по крайней мере, больше или равна классификации сущностей, которые он содержит.

B3. Изменения в объектах – информация, переносимая из объекта всегда содержит классификацию объекта. Информация, вставляемая в объект должна иметь классификацию ниже классификации этого объекта.

B4. Просмотр – пользователь может просматривать (на некотором устройстве вывода) только сущности с классификацией меньше, чем классификация устройства вывода и степень доверия к пользователю.

B5. Доступ к объектам, требующим степени доверия – пользователь может получить доступ к косвенно адресованной сущности внутри объекта, требующего степени доверия, только если его степень доверия не ниже классификации контейнера.

B6. Преобразование косвенных ссылок – пользовательский идентификатор признается законным для сущности, к которой он обратился косвенно, только если он авторизован для просмотра этой сущности через ссылку.

B7. Требование меток – сущности, просмотренные пользователем должны быть помечены его степенью доверия.

B8. Установка степеней доверия, ролей, классификации устройств – только пользователь с ролью администратора безопасности системы

может устанавливать данные значения. Текущее множество ролей пользователя может быть изменено только администратором безопасности системы или этим пользователем.

B9. Понижение классификации информации – никакая классифицированная информация не может быть понижена в уровне своей классификации, за исключением случая, когда эту операцию выполняет пользователь с ролью «пользователь, уменьшающий классификацию информации».

B10. Уничтожение информации – операция уничтожения информации проводится только пользователем с ролью «пользователь, уничтожающий информацию».

Модель Лендвера достаточно близка категориям объектно-ориентированного программирования и описывает иерархические объекты в КС. Однако полная реализация данной модели представляется достаточно сложной.

2.3.3.4. Резюме по моделям состояний

Для систем, построенных на основании моделей конечных состояний, характерна более высокая степень надежности, чем для систем, построенных на основании модели дискретного доступа. Это связано с тем, что система, построенная на основании данной модели, должна отслеживать не только правила доступа субъектов системы к объектам, но и состояния самой системы. Таким образом, каналы утечки в системах данного типа не заложены непосредственно в модель (что мы наблюдаем в системах, построенных на основании модели дискретного доступа), а могут появиться только при практической реализации системы, вследствие ошибок разработчика. С другой стороны, реализация систем данного типа довольно сложна и требует значительных ресурсов вычислительной системы.

3. ОБЕСПЕЧЕНИЕ ГАРАНТИЙ ВЫПОЛНЕНИЯ ПОЛИТИКИ БЕЗОПАСНОСТИ

3.1. Обеспечение гарантий выполнения политики безопасности

Очевидно, что при изменении функционально ассоциированных с субъектом реализации политики безопасности (МБО) объектов, могут измениться и свойства самого МБО, заключающиеся в фильтрации потоков и, как следствие, могут возникнуть потоки, принадлежащие множеству N. Иллюстрация этого факта приведена на ***рис.1***. Введем в связи с этим понятие корректности субъектов.

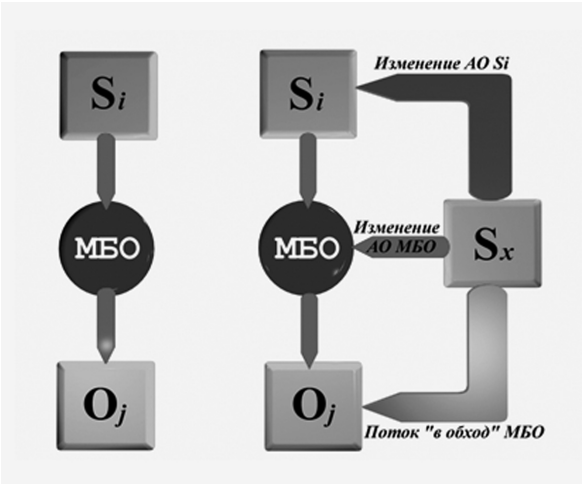


Рис. 1. Возможные пути нарушения ПБ (АО - ассоциированные объекты)

Определение 11. Пара субъектов S_i и S_j называется невлияющими друг на друга (или корректной относительно друг друга), если в любой момент времени отсутствует поток (изменяющий состояние объекта) между ассоциированным объектом субъекта S_i(Osi) и S_j(Osj), причем Osj не является ассоциированным объектом S_i, а Osi не является ассоциированным объектом S_j.

Дадим некоторые пояснения к определению: «изменение состояния объекта» трактуется в данном определении как нетождественность объектов в соответствующие моменты времени, но при этом подчеркнуто, что операция изменения объекта локализована в субъекте, с которым этот объект не ассоциирован. Смысл понятия корректности можно пояснить на примере: существующие в едином пространстве ОП программы

не должны иметь функциональных возможностей изменения «чужого» вектора кода и состояния переменных.

Вообще говоря, можно сформулировать более жесткое определение.

Определение 12. Пара субъектов S_i и S_j называется абсолютно невлияющей друг на друга (или абсолютно корректной относительно друг друга), если в условиях определения 11 множества ассоциированных объектов указанных субъектов не имеют пересечения.

Абсолютная корректность легко достижима в случае виртуального адресного пространства.

Определение абсолютной корректности позволяет сформулировать достаточные условия гарантированного осуществления только легального доступа.

Утверждение 1 (достаточное условие гарантированного выполнения политики безопасности в КС 1).

Монитор безопасности объектов разрешает порождение потоков только из множества L, если все существующие в системе субъекты абсолютно корректны относительно него и друг друга.

Доказательство

Условие абсолютной корректности (по определению 12) предполагает неизменность функционально ассоциированных объектов МБО (поскольку потоков, изменяющих ассоциированные объекты МБО, не существует). С другой стороны, такие потоки могут появиться при изменении ассоциированных объектов, принадлежащих другим субъектам КС (изменятся свойства субъекта, в том числе (возможно) и по порождению потоков к МБО). Условие корректности субъектов относительно друг друга делает это невозможным (по определению абсолютной корректности). Это в свою очередь означает, что МБО реализует только потоки из подмножества L. Утверждение доказано.

Однако сформулированное утверждение накладывает весьма жесткие и трудноисполнимые условия на свойства субъектов в КС. Кроме того, невозможно гарантировать корректность любого субъекта, активизируемого в КС, относительно МБО. В связи с этим логично ограничить множество порождаемых субъектов, которые априорно корректны относительно МБО. В связи с этим введем определение монитора порождения субъектов (по аналогии с монитором обращений) и монитора безопасности субъектов.

Определение 13. Монитор порождения субъектов (МПС) – субъект, активизирующийся при любом порождении субъектов.

По аналогии с переходом от МО к МБО введем понятие монитора безопасности субъектов.

Определение 14. Монитор безопасности субъектов (МБС) – субъект, который разрешает порождение субъектов только для фиксированного подмножества пар активизирующих субъектов и порождающих объектов.

Воздействие МБС выделяет во всем множестве субъектов S подмножество разрешенных E. Заметим также, что если в подмножество субъектов в момент времени t включается субъект МБС, то первым аргументом операции Create может быть только субъект, входящий во множество субъектов, а аргумент-объект, вообще говоря, любым.

Сформулируем теперь ряд базовых определений, которые в дальнейшем будут повсеместно использоваться.

Определение 15. КС называется замкнутой по порождению субъектов, если в ней действует МБС, разрешающий порождение только фиксированного конечного подмножества субъектов для любых объектов-источников, рассматриваемых для фиксированной декомпозиции КС на субъекты и объекты.

При рассмотрении вопросов реализации защищенных сред будет использоваться термин «замкнутая программная среда», который по существу эквивалентен приведенному выше определению.

Однако замкнутости КС по порождению субъектов недостаточно для описания свойств системы в части защищенности, поскольку необходимо обеспечить корректность порождаемых МБС субъектов относительно его самого и МБО. Механизм замкнутой программной среды сокращает множество возможных субъектов до некоторого множества фиксированной мощности, но при этом допускает существование некорректных субъектов, включенных в замкнутую среду.

Сформулируем определение изолированности КС.

Определение 16. Множество субъектов КС называется изолированным (абсолютно изолированным), если в ней действует МБС и субъекты из порождаемого множества корректны (абсолютно корректны) относительно друг друга и МБС.

Следствие. Любое подмножество субъектов изолированной (абсолютно изолированной КС), включающее МБС, также составляет изолированную (абсолютно изолированную) среду.

Следствие. Дополнение изолированной (абсолютно изолированной) КС субъектом, корректным (абсолютно корректным) относительно любого из числа входящих в изолированную (абсолютно изолированную) среду, оставляет ее изолированной (абсолютно изолированной).

Теперь возможно переформулировать достаточное условие гарантированного выполнения политики безопасности следующим образом.

Утверждение 2 (достаточное условие гарантированного выполнения политики безопасности в КС 2).

Если в абсолютно изолированной КС существует МБО и порождаемые субъекты абсолютно корректны относительно МБО, а также МБС абсолютно корректен относительно МБО, то в такой КС реализуется только доступ, описанный в ПРД.

Доказательство

Из определения абсолютной изолированности следует возможность существования в КС только конечного множества субъектов, которые в свою очередь корректны относительно МБС (по определению 16 и следствию из него).

Далее, по условию утверждения (корректность МБО относительно любого из порождаемых субъектов и МБС), ассоциированные объекты могут изменяться только самим МБО, следовательно, в КС реализуются только потоки, принадлежащие множеству L. Утверждение доказано.

Легко видеть, что данное утверждение является более конструктивным относительно предыдущего достаточного условия гарантированной защищенности, поскольку ранее требовалась корректность МБО относительно произвольного субъекта, что практически невозможно. В данном же случае множество субъектов ограничено за счет применения механизма МБС и возможно убедиться в попарной корректности порождаемых субъектов.

При рассмотрении технической реализации изолированности субъектов в КС будет употребляться термин «изолированная программная среда» (ИПС), который описывает механизм реализации изолированности для конкретной программно-аппаратной реализации КС и при соответствующей декомпозиции на субъекты и объекты.

При рассмотрении операции порождения субъекта возникает весьма важная проблема, связанная с тем, что в реальных КС одинаково поименованные объекты могут иметь различное состояние в пространстве (например, быть размещенными в различных каталогах) или во времени.

Предположим, что зафиксировано состояние объекта O_m в некоторый момент времени t_0. Будем обозначать состояние объекта O_m в момент времени t как $O_m[t]$.

Определение 17. Операция порождения субъекта ***Create***(S_k,O_m)**->**S_i называется порождением с контролем неизменности объекта, если для любого момента времени $t>t_0$, в который активизирована операция порождения объекта ***Create***, порождение субъекта S_i возможно только при тождественности объектов $O_m[t_0]$ и $O_m[t]$.

Следствие. В условиях определения 17 порожденные субъекты $S_i[t1]$ и $S_i[t2]$ тождественны, если $t1>t_0$ и $t2>t_0$. При t1=t2 порождается один и тот же субъект.

При порождении субъектов с контролем неизменности объекта в КС допустимы потоки от субъектов к объектам-источникам, участвующим в порождении субъектов, с изменением их состояния.

Утверждение 3 (базовая теорема ИПС)

Если в момент времени t_0 в изолированной КС действует только порождение субъектов с контролем неизменности объекта, и существуют потоки от любого субъекта к любому объекту, не противоречащие условию корректности (абсолютной корректности) субъектов, то в любой момент времени $t>t_0$ КС также остается изолированной (абсолютно изолированной).

Доказательство

По условию утверждения в КС возможно существование потоков, изменяющих состояние объектов, не ассоциированных в этот момент времени с каким-либо субъектом. Если объект с измененным состоянием не является источником для порождения субъекта, то множество субъектов изолированной среды нерасширяемо, в противном случае (измененный объект является источником для порождения субъекта) по условиям утверждения (порождение субъекта с контролем) порождение субъекта невозможно. Следовательно, мощность множества субъектов не может превышать той, которая была зафиксирована до изменения состояния любого объекта. По следствию из определения 17 (о замкнутости множества субъектов в ИПС с невозрастанием мощности множества субъектов), получим, что множество субъектов КС изолировано. Утверждение доказано.

Можно сформулировать методологию проектирования гарантированно защищенных КС. Сущность данной методологии состоит в том, что при проектировании защитных механизмов КС необходимо опираться на совокупность приведенных выше (утверждения 1-3) достаточных условий, которые должны быть реализованы для субъектов, что гаран-

тирует защитные свойства, определенные при реализации МБО в КС (т.е. гарантированное выполнение заданной МБО политики безопасности).

Рассмотренная концепция изолированной программной среды является расширением зарубежных подходов к реализации ядра безопасности.

Обычно модель функционирования ядра безопасности изображается в виде следующей схемы, представленной на рис. 4.

На ***рис. 2*** «база данных защиты» означает объект, содержащий в себе информацию о потоках множества L (защита по «белому списку» – разрешения на потоки) или N (защита по «черному списку» – запрещение на потоки).

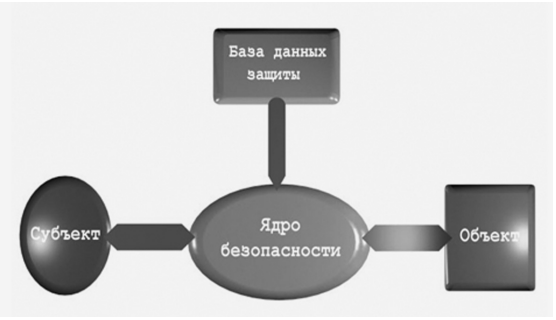


Рис. 2. Классическая модель ядра безопасности

Для учета влияния субъектов в КС необходимо рассматривать расширенную схему взаимодействия элементов системы реализации и гарантирования ПБ.

На **рис. 3** подчеркнута роль монитора безопасности субъектов при порождении субъектов из объектов. Взаимодействие субъектов и объектов при порождении потоков уточнено введением ассоциированных с субъектом объектов. Конструкция ОУ на схеме обозначает объект управления, т.е. объект, содержащий информацию о разрешенных значениях отображения ***Stream*** (об элементах множества L или N) и ***Create*** (элементы множества E). Объект управления может быть ассоциирован (ассоциированный объект-данные) как с МБО, так и с МБС.

Перейдем к описанию практических методов построения ИПС. Целью рассмотрения практических подходов является иллюстрация тезиса о том, что достаточные условия гарантированной защищенности могут быть практически выполнены в реальных КС.

3.2. Метод генерации изолированной программной среды при проектировании механизмов гарантированного поддержания политики безопасности

Опираясь на утверждение 3 (базовую теорему ИПС), сформулированное и доказанное выше, опишем метод субъектно-объектного взаимодействия в рамках ИПС для более конкретной архитектуры КС.

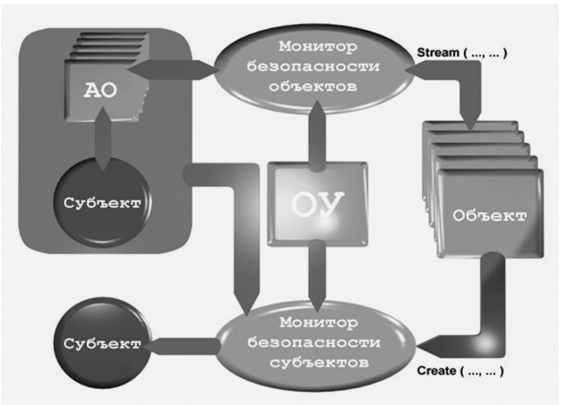


Рис. 3. Ядро безопасности с учетом контроля порождения субъектов

Из утверждения 3 следует, что для создания гарантированно защищенной КС (в смысле выполнения заданной политики безопасности) необходимо:

1. Убедиться в попарной корректности субъектов, замыкаемых в ИПС (либо убедиться в корректности любого субъекта относительно МБО и МБС).

2. Спроектировать и реализовать программно (или программно-аппаратно) МБС так, чтобы:

• для любого субъекта и любого объекта производился контроль порождения субъектов (т.е. чтобы реализация МБС соответствовала его определению),

• порождение любого субъекта происходило с контролем неизменности объекта-источника.

3. Реализовать МБО в рамках априорно сформулированной политики безопасности.

Надо заметить, что приводимые выше утверждения верны только тогда, когда описанная и реализованная политика безопасности не нарушает их условий (проверка данного факта зависит от модели ПБ и является отдельной весьма важной задачей).

Кроме того, необходимо обратить внимание на следующее. Объект управления, который является ассоциированным объектом МБС (обычно ассоциированный объект-данные), играет решающую роль в проектировании ИПС. При возможности изменения состояния объекта управления потенциально возможно «размыкание» программной среды, т.е. добавление к множеству разрешенных субъектов дополнительных, реализующих злоумышленные функции. С другой стороны, процесс управления безопасностью подразумевает возможность изменения объекта управления (подробнее в части 3). Возможность изменения объекта управления (реализация потока ***Stream***(субъект управления, АО объекты субъекта управления)->ОУ) должна присутствовать для выделенных субъектов (возможно с дополнительным условием активизации этого субъекта выделенным пользователем (пользователями)).

Важную роль при проектировании ИПС играет свойство КС, заключающееся в поэтапной активизации субъектов из объектов различного уровня представления информации. Рассмотрим в таблице 2 иерархию уровней при загрузке операционной системы.

Таблица 2. Иерархия уровней при загрузке ОС

Уровень	Субъект	Локализация	Представление информации	Через какие функции реализуются потоки
0	Субъект аппаратно-программного уровня	ПЗУ (Bios)	сектора	через микропрограммы ПЗУ
1	Субъект уровня первичной загрузки	Загрузчик ОС	сектора	через Bios или первичный загрузчик
2	Субъект уровня вторичного загрузчика (драйвер)	драйверы ОС	сектора	через Bios или первичный загрузчик
3	Субъект уровня ОС	ядро ОС	файлы	через драйверы
4	Субъект пользовательского уровня	прикладные приложения	файлы	через ядро ОС

В таблице выделен термин «сектор» для обозначения представления объекта аппаратно-программного уровня. Он обозначает непрерывную последовательность элементов хранения (байт) на материальном носителе, характеризуемую местом расположения.

Термин «файл» обозначает абстрактный объект, построенный по списочной структуре из объектов «сектор». Объекты типа «файл» и «сектор» выделены исключительно исходя из типовой архитектуры объектов КС.

В общем случае можно говорить о рекурсивной структуре объектов некоторого уровня, вмещающей объекты предыдущего уровня. На нулевом уровне первичный объект (элементарная структура нижнего уровня) в таблице 2 соответствует термину «сектор».

С учетом иерархической структуры представления объектов можно говорить о том, что в начальные этапы активизации КС декомпозиция на субъекты и объекты динамически изменяется. Следовательно, основная теорема ИПС может быть применима только на отдельных интервалах времени, когда уровень представления объектов постоянен и декомпозиция фиксирована. Можно утверждать, что ИПС, действующую от момента активизации до момента окончания работы КС, невозможно сформировать в начальный момент активизации КС.

Пусть в КС выделяется конечное число уровней представления объектов U={0, ..., R}, R – максимальный уровень представления объекта.

С точки зрения выполнения условий утверждения 3 имело бы смысл говорить о некотором «стационарном» состоянии КС, когда в отображениях ***Stream*** и ***Create*** участвуют только объекты уровня R. Тогда реали-

зация МБС может быть значительно упрощена (в том смысле, что все аргументы-объекты операции ***Create*** имеют тот же уровень). Необходимо обратить внимание на то, что такое требование, с одной стороны, может накладывать ограничительные условия на свойства прикладного ПО (невозможность инициирования потоков, включающих объекты уровня менее R, прикладными программами), а с другой стороны, быть следствием проектировочных решений реализации субъекта, локализованного в ядре операционной системы (примером является ОС Windows NT 4.0, запрещающее операции ниже уровня «файл» со стороны субъектов прикладного уровня).

Практическая реализация всех операционных систем позволяет выделить две фазы их работы: активизация субъектов с ростом уровня представления объектов (фаза загрузки или начальная фаза) и фаза стационарного состояния (когда уровень представления объектов не увеличивается). Конечно, необходимо сделать оговорку, касающуюся возможности реализации потоков к объектам нижнего уровня (операционные системы типа DOS, в которых возможна операция с любым объектом нижнего уровня (сектор) из программ прикладного уровня).

Тогда практическая реализация ИПС может состоять из двух этапов: предопределенное выполнение начальной фазы, включающее в себя момент активизации МБС (и МБО) и работу в стационарной фазе в режиме ИПС (возможно, с контролем неизменности объектов-источников).

Введем понятие последовательности активизации компонент КС. Смысл вводимых понятий и формулируемых ниже утверждений состоит в необходимости приведения субъектов КС в одно и то же состояние после активизации первичного субъекта аппаратно-программного уровня или, иначе говоря, в задании предопределенной последовательности активизации субъектов КС.

Обозначим: Z_L последовательность пар (i, j)t (t=0, 1, 2, ... ,l-1 – моменты времени) длины l таких, что ***Create***$(S_i,O_j)[t]->S_m[t+1]$.

Обозначим также:

Sz – множество всех субъектов, включенных в последовательность Z_L,

O_Z – множество всех объектов, включенных в последовательность Z_L.

Для многопотоковых КС можно рассматривать несколько (возможно, зависимых друг от друга) последовательностей Z_L и соответственно множеств Sz и O_Z.

Определение 18. Состоянием КС в момент времени t называется упорядоченная совокупность состояний субъектов.

Напомним, что каждый объект есть слово в априорно определенном языке, а понятие состояния субъекта сформулировано выше.

Утверждение 4 (условие одинакового состояния КС).

Состояние КС в моменты времени tx1 и tx2 (tx1 и tx2 исчисляются для двух отрезков активности КС от нулевого момента активизации КС $t_0$1 и $t_0$2- например, включения питания аппаратной части) одинаково, если:

1. tx1 = tx2,
2. тождественны субъекты $S_i[t_0 1]$ и $S_i[t_0 2]$,
3. неизменны все объекты из множества O_Z,
4. неизменна последовательность Z_L.

Доказательство (по принципу математической индукции)

Верность утверждения при t=1 следует из определения тождественности субъектов.

Пусть утверждение верно для t=k<l.

Тогда в момент времени k+1 могут быть порождены только тождественные субъекты, поскольку тождественны активизирующие субъекты (по предположению индукции) и по условию утверждения неизменны элементы множества O_Z.

Длина l последовательности Z_L определяется:

1. По признаку невозможности управления субъектами, принадлежащими множеству Sz со стороны пользователя (в противном случае последовательность активизации субъектов может быть изменена).

2. По признаку доступности для контроля неизменности всех объектов из множества Oz.

3. По признаку невозрастания уровня представления информации (в данном случае имеется в виду, что существует такой момент времени t_x такой, что для любого $t>t_x$ объект-аргумент O_j операции ***Stream***$(S_i,O_j)_t$ принадлежит одному уровню представления).

Необходимо заметить, что последовательность Z_L локализуется в некотором объекте, либо совокупности объектов (например, для DOS последовательность активизации субъектов предопределена содержанием файлов AUTOEXEC.BAT и CONFIG.SYS) и неизменность последовательности Z_L тождественна неизменности указанных объектов, для ОС Windows NT последовательность активизации компонент определена содержанием соответствующих ключей реестра (registry).

Пусть в последовательности Z_L можно выделить z_i такое, что для любого z_k, k>i, отображений ***Create*** и ***Stream*** используют только объекты уровня R. Другими словами, с момента времени i наступает стационарная фаза функционирования КС.

В этих условиях, а также при попарной корректности субъектов и действии МБС с контролем неизменности объектов-источников на уровне R с момента времени m>k верно:

Утверждение 5 (достаточное условие ИПС при ступенчатой загрузке).

При условии неизменности Z_L и неизменности объектов из O_Z в КС с момента времени установления неизменности Z_L и O_Z действует изолированная программная среда.

Доказательство

Необходимо заметить, что все условия утверждения 5 соответствуют утверждению 4. Уточнения касаются структуры последовательности Z_L.

Согласно утверждению 4 с момента времени t=0 до момента t=l действует изолированная (в рамках) Sz программная среда.

Для доказательства утверждения необходимо убедиться в том, что:

- МБС в момент времени t=m гарантировано активизируется,
- в любой момент t>m программная среда изолирована.

Первое следует из утверждения 4 (при t=m состояние программной среды всегда будет одинаково, следовательно, всегда будет активизирован субъект МБС). Второе следует из определения МБС и условия теоремы.

С момента времени t=0 до момента времени l программная среда изолирована, с момента времени t>m программная среда также изоли-

рована, следовательно, КС изолирована при любом t>0. Утверждение доказано.

Используя утверждения 3, 4 и 5, рассмотрим процесс практического проектирования защищенного фрагмента КС.

Первоначально необходимо убедиться в выполнении условий корректности или абсолютной корректности для субъектов, участвующих в порождении ИПС. Указанные субъекты в основном могут быть локализованы на уровне программно-аппаратной компоненты ЭВМ (программы ПЗУ, загрузчики операционных сред), т.е. работающие на уровне, близком к взаимодействию с оборудованием КС, либо на уровне операционной среды. Доказательство корректности субъектов программно-аппаратного уровня значительно отличается от соответствующих доказательств для субъектов прикладного уровня. В связи с этим выделим проверку условий корректности субъектов в два шага. Шагом 1 назовем доказательство корректности субъектов программно-аппаратного уровня. Понятие **модуль** обозначает реализацию объекта-источника, а совокупность субъекта, порожденного из объекта-источника и всего множества ассоциированных с этим субъектом объектов в течение всего времени существования субъекта называется, как правило, **процессом (или задачей, заданием)**.

Далее, необходимо определить состав программных средств базовой вычислительной среды, т.е. определить конкретную операционную среду, дополнительные программные средства сервиса (например, программные оболочки или средства телекоммуникации) и программные средства поддержки дополнительного оборудования (программы управления принтером и др.). После этого наступает самый трудоемкий этап (Шаг 2), на котором необходимо убедиться в корректности субъектов описанного базового набора программных средств. При этом важно заметить следующее.

В составе ПО КС не должно быть целого класса возможностей – назовем их инструментальными. Прежде всего, это возможность изменения состояния ассоциированных объектов со стороны субъекта (например, изменение содержимого оперативной памяти) других субъектов (изменение содержания подразумевает существование операций ***Stream*** типа запись), возможность инициирования и прекращения выполнения процессов нестандартным образом (помимо механизмов операционной среды). Кроме того, при реализации МБС и МБО на стационарной фазе функционирования КС необходимо отсутствие в любых субъектах, замкнутых в ИПС, операций порождения потоков ***Stream*** к объектам уровня k<R.

Обобщенно достаточные условия к базовому набору ПО можно сформулировать следующим утверждением.

Утверждение 6 (требования к субъектному наполнению изолированной программной среды).

Для того чтобы ИПС поддерживалась в течение всего времени активности КС, достаточно, чтобы в составе программного обеспечения, могущего быть инициированным в ИПС, не было функций порождения субъектов и прекращения их работы, кроме заранее предопределенных при реализации МБС и не существовало возможностей влияния на среду выполнения (под средой выполнения понимается множество ассоци-

ированных объектов) любого процесса, а также инициирования потоков к объектам логического уровня менее R.

Легко видеть, что данное утверждение есть собранные воедино условия выполнения вышеприводимых утверждений. Более детализированное описание свойств субъекта приводится в приложении 1.

Поясним требование невозможности прекращения выполнения субъекта каким-либо иным образом, кроме предопределенного. В данном случае необходимо учитывать, что во множестве субъектов, замкнутых в ИПС, выделены два особых субъекта – МБС и МБО. Прекращение существования МБС означает нарушение условия замкнутости среды, а прекращение существования МБО означает допустимость потоков множества N, т. е. несанкционированный доступ.

Шаг 3 заключается в проектировании и разработке программных или программно-аппаратных средств защиты в КС, а затем и их тестировании. Он подразумевает проектирование и реализацию в заданном множестве субъектов МБС и МБО.

Практически шаги 1-3 могут быть выполнены, исходя из описанных в литературе методик разработки и тестирования ПО.

Шаг 4 заключается в «замыкании» всего комплекса программного обеспечения, включая и средства защиты, в изолированную программную среду.

Итак, показано, что основными элементами поддержания изолированности программной среды являются контроль целостности и контроль порождения процессов.

Выше мы уже сформулировали понятия МБС и порождения субъектов с контролем их неизменности. Необходимо заметить, что для достоверного контроля неизменности объекта (т.е. с вероятностью ошибки, равной 0) необходимо убедиться в полном тождестве проверяемого объекта и образца. Из этого следует, что эталон должен содержать не менее информации, чем проверяемый объект. Из этого в свою очередь следует, что эталонный объект должен быть, как минимум, одинаковой длины с проверяемым. На практике такой подход может быть применен с серьезными ограничениями (например, для объектов небольшого объема типа программ ПЗУ или загрузчиков ОС).

В связи с этим для контроля целостности применяют объекты, содержащие информацию, зависящую от всего содержания объекта, но, тем не менее, значительно меньшего объема, вычисленную при помощи класса функций типа «хеш-функций». Очевидно, что в этом случае процесс установления неизменности объекта становится вероятностным.

Исходя из данного факта, невозможно говорить о гарантированных (детерминировано) свойствах системы (поскольку неизменность объекта гарантируется лишь с некоторой вероятностью, не равной 1). Следовательно, все условия утверждений выполняются с некоторой вероятностью, зависящей от свойств применяемых для контроля целостности хеш-функций. Для подчеркивания изменившихся условий будем говорить далее не о **контроле неизменности объекта**, а о **контроле целостности** (КЦ) объекта.

Необходимо отметить также, что в процедуре контроля неизменности (которая теперь принимает вероятностный характер) участвует как минимум два объекта: объект контроля и эталонный объект (хеш-зна-

чение), а также субъект, реализующий хеш-функцию и производящий сравнение.

Поэтому для субъекта контроля целостности важным является выполнение следующих условий:

• качественный алгоритм контроля целостности (термин «качественный» будет пояснен ниже),

• контроль реальных данных (т.е. отображение состояния контролируемого и эталонного объектов в ассоциированные объекты-данные субъекта контроля целостности, совпадающее с тождественным).

Поясним подробнее второй пункт. Контроль целостности всегда сопряжен с чтением данных (т.е. с инициированием потоков от объектов к ассоциированным объектам-данным субъекта контроля целостности, причем потоки могут соответствовать различному уровню представления информации (чтение по секторам, по файлам и т.д.)). Например, встроенный в BIOS ПЭВМ субъект (практически это программная закладка – см. ниже) может навязывать при чтении вместо одного сектора другой или редактировать непосредственно буфер, в который были прочитаны данные. Аналогичный эффект может быть вызван субъектами операционной среды, например, субъектами, локализованными в первичных загрузчиках ОС. С другой стороны, даже контроль самого BIOS может происходить «под наблюдением» какой-либо дополнительной аппаратуры и не показать его изменения. Аналогичные эффекты могут возникать и при обработке файла. Цель организации режима чтения реальных данных состоит в тождественном отображении параметров чтения на АО субъекта чтения (поток от АО субъекта КЦ к АО субъекта чтения) и тождественном отображении считываемого объекта (в соответствии с параметрами, переданными субъекту чтения) к ассоциированным объектам-данным субъекта КЦ

Поясним теперь понятие качественного КЦ с точки зрения математических свойств функции КЦ. Предположим, что имеется некоторый объект F и некоторый алгоритм H, преобразующий объект F в некоторый объект M, который представляется словом того же языка, но меньшей длины. Этот алгоритм таков, что при случайном равновероятном выборе двух объектов F_1 и F_2 из множества возможных соответствующие им объекты M1=H(F_1) и M2=H(F_2) с высокой вероятностью различны. Тогда проверка целостности данных строится так: рассматриваем объект F, по известному алгоритму H строим K=H(F) и сравниваем M заранее вычисленное как M = H(F) с K. При совпадении считаем объект неизменным. Алгоритм H называют хеш – функцией, а число M – хеш – значением.

Качество КЦ определяется в данном случае выполнением следующих условий:

1. по известному объекту M=H(F) нахождение другого объекта G не тождественного F, такого, что M=H(G), является задачей с трудоемкостью не менее заданной Th,

2. объект M должен быть недоступен для изменения,

3. длина объекта M должна обеспечивать условную вероятность P(H(F_1)=H(F2)/F_1 не тождественны F_2) не более заданной P_h.

Таким образом, при условии недоступности хеш-значения для изменения и доступности для изменения объекта-источника трудоемкость нарушения ИПС с КЦ объектов-источников (т.е. возможность породить

субъект из объекта-источника, нетождественного исходному объекту) совпадает с Th. При однократной попытке инициировать субъект из случайно равновероятно выбранного объекта-источника вероятность нарушения ИПС (успешное порождение субъекта) не превосходит P_h. Итак, «качество» ИПС определяется свойствами хеш-функции H, а именно: величинами Th и P_h.

Обобщим приводимые выше рассуждения в методе «безопасной загрузки» или ступенчатого контроля. Он заключается в постепенном установлении неизменности компонент программно-аппаратной среды: сначала проверяется неизменность программ ПЗУ, при положительном исходе через проверенные на целостность программы ПЗУ считывается загрузочный сектор и драйверы операционной системы (по секторам) и их неизменность также проверяется, кроме того, проверяется целостность объекта, определяющего последовательность активизации компонент; через функции чтения проверенной ОС инициируется процесс контроля порождения процессов (реализация МБС); инициирование процесса контроля доступа к объектам завершает проектирование гарантировано защищенной КС.

Рассматривая вопросы программно-технической реализации ИПС, необходимо заметить, что мощность множества субъектов в некотором сегменте КС (выделенном по признаку принадлежности одной ЭВМ) с момента включения питания до момента запуска процессов пользователя, увеличивается. Первоначально активизируются субъекты аппаратно-программного уровня (программы ПЗУ), затем указанные субъекты порождают из объектов-источников данного уровня (это, как правило, сектора внешних носителей информации) субъектов уровня операционной среды.

Субъекты уровня операционной среды, как уже отмечалось, также делятся на два подуровня: нижний уровень – субъекты – первичные загрузчики ОС (работающие с информацией уровня секторов) и верхний уровень – субъекты-драйверы (порождаемые субъектами – первичными загрузчиками из объектов-секторов), работающие с объектами уровня «файл» (последовательности секторов). На этапе перехода от субъектов-загрузчиков к субъектам-драйверам происходит переход и к другой декомпозиции КС на объекты (от секторов к файлам). Указанная иерархия действует в любой известной на сегодняшний день КС и естественным образом предопределяет архитектуру, в рамках которой формируется и функционирует ИПС.

Например, аппаратная архитектура ПЭВМ типа IBM PC задает следующие этапы активизации различных субъектов КС. При включении питания ПЭВМ происходит тестирование ОП, инициализация таблицы векторов прерываний и поиск расширений BIOS. При их наличии управление передается на них. После отработки расширений BIOS в память считывается первый сектор дискеты или винчестера, и управление передается на него (образуется код загрузчика), затем код загрузчика считывает драйверы операционной системы, далее интерпретируются файлы конфигурации, подгружается командный интерпретатор и выполняется файл автозапуска.

При реализации ИПС на нее должна быть возложена функция контроля запусков программ и контроля целостности.

При описании методологии проектирования ИПС упоминалась проблема контроля реальных данных. Эта проблема состоит в том, что контролируемая на целостность информация может представляться поразному на разных уровнях.

Внедренный в систему субъект может влиять на процесс чтения-записи данных на уровне файлов (или на уровне секторов) и предъявлять системе контроля некоторые другие вместо реально существующих данных. Этот механизм неоднократно реализовался в STELS-вирусах. Однако верно утверждение.

Утверждение 7 (достаточное условие чтения реальных данных).

Если субъект, обслуживающий процесс чтения данных (т.е. указанный субъект инициируется запрашивающим данные субъектом и участвует в потоке), содержал только функции тождественного отображения данных на ассоциированные объекты-данные любого субъекта, инициирующего поток чтения, и целостность объекта-источника для этого субъекта зафиксирована, то при его последующей неизменности чтение с использованием порожденного субъекта будет чтением реальных данных.

Доказательство

Верность утверждения следует из определения тождественности субъекта и из условия утверждения, гарантирующего неизменность объекта-источника.

Необходимо и здесь сделать оговорку о вероятностном характере установления неизменности и говорить, что чтение реальных данных возможно с вероятностью, определяемой алгоритмом КЦ.

Метод ступенчатого контроля не противоречит утверждениям 4 и 5 и предусматривает разделение последовательности активизации компонент Z_L на подпоследовательности с одинаковым уровнем представления информации.

Реализация метода ступенчатого контроля целостности должна удовлетворять условиям утверждения 4.

Опишем практическую реализацию сформулированных методов.

3.3. Реализация гарантий выполнения заданной политики безопасности

Выше было сказано о том, что субъект контроля неизменности объектов, входящих в процедуры активизации КС и объектов, описывающих последовательность активизации компонент, должен быть активен уже на этапе работы субъектов аппаратно-программного уровня, но его объект-источник технически не может быть проверен на неизменность. В связи с этим подчеркнем весьма важный факт для любых реализаций ИПС.

Аксиома 5. *Генерация ИПС рассматривается в условиях неизменности конфигурации тех субъектов КС, которые активизируются до старта процедур контроля целостности объектов* O_z *и последовательности* Z_L*. Неизменность данных субъектов обеспечивается внешними по отношению к самой КС методами и средствами. При анализе или синтезе защитных механизмов свойства указанных субъектов являются априорно заданными.*

При решении практических вопросов генерации ИПС можно выделить три самостоятельных направления.

Первое из них связано с использованием внешних по отношению к КС субъектов (как правило, размещенных на внешнем носителе), целостность которых гарантируется методами хранения или периодического контроля. Предопределенность активизации субъектов, локализованных на внешних носителях, обеспечивается свойствами субъектов аппаратно-программного уровня (например, возможно установить такую аппаратную конфигурацию ПЭВМ, при которой будет происходить загрузка операционной системы с ГМД).

Второе направление связано с локализацией ИПС в рамках территориально ограниченного рабочего места (как правило, ПЭВМ) и использует аппаратную поддержку для задания предопределенной последовательности активизации субъектов. Данное направление, как правило, включает и аппаратную поддержку аутентификации пользователей.

Третье направление связано с реализацией метода доверенной загрузки операционной среды с использованием уже имеющихся в ней механизмов реализации и гарантирования ПБ.

Необходимо заметить, что в различные интервалы активности КС субъектами могут управлять различные пользователи, для которых множество разрешенных субъектов E различно, в связи с этим будем говорить о множестве E_i для i-го пользователя КС.

Будем также подразумевать, что перед установлением однозначного соответствия множества E_i пользователю i происходит процедура его аутентификации (см. главу 2).

Ниже будут кратко рассмотрены все способы реализации ИПС. Говоря о первом из них необходимо отметить, что в его рамках можно рассматривать конфигурацию ИПС в двух вариантах:

• при локализации всех объектов-источников для порождения ИПС в рамках одного или нескольких внешних носителей,

• при локализации части объектов-источников на внешнем носителе, а части – во внешней памяти рабочего места.

Вторая конфигурация характеризуется потенциальной возможностью нарушения изолированности, состоящей в том, что активизация субъектов из объектов-источников, не принадлежащих внешнему носителю, может производиться вне рамок ИПС. В качестве примера можно рассмотреть ситуацию, когда программы запускаются в рамках операционной среды, загруженной с дискеты. С другой стороны, запуск указанных программ возможен и при загрузке ОС с другого носителя (в частности, с носителей рабочего места) и при этом возможна активизация и тех модулей, которые находятся на дискете.

Следовательно, основной задачей при использовании внешнего носителя для генерации ИПС является обеспечение невозможности активизации любого субъекта из объекта-источника внешнего носителя вне рамок зафиксированной для этого носителя последовательности активизации компонент ИПС.

Наиболее ранний описанный способ проектирования ИПС в рамках подхода с использованием внешнего носителя получил название «невидимой дискеты». Этот способ заключается в том, что все объекты, принадлежащие множеству O_Z и объекты, описывающие последовательность Z_L, помещаются на внешний носитель, с которого может быть произведена загрузка операционной системы (обычно дискета). Неиз-

менность объектов обеспечивается физической защитой носителя от записи.

Кроме того, использование специальной технологии не позволяет использовать объекты (в том числе и обеспечить выполнение программ) без загрузки ОС именно с этой дискеты. Практически такая дискета выглядит достаточно нетривиально: будучи помещенной в дисковод ПЭВМ, она выглядит как неформатированная (или, в ином варианте, пустая). После загрузки с такой «пустой» дискеты пользователь сразу «погружается» в заданную программу и работает с ней, обращаясь, в том числе и к данным на винчестере и запуская программы с локальных несменяемых носителей рабочего места с предварительным контролем неизменности соответствующих им объектов источников (исполняемых файлов).

Предлагаемый способ позволяет исключить использование изготовленной дискеты без загрузки с нее. Дополнив загружаемую с такой дискеты операционную среду программами проверки целостности, можно добиться соблюдения всех требований изолированности программно – аппаратной среды.

Как следует из утверждения 5, одним из важнейших условий поддержания ИПС является невозможность изменения последовательности активизации компонент.

В данном случае целостность объектов, содержащих последовательность активизации компонент, гарантируется физическим запретом записи на дискету.

Важной проблемой является невозможность прерывания процесса активизации компонент. В ряде операционных сред для этого имеются штатные возможности, предусмотренные для обеспечения защиты от ошибок пользователя, сформировавшего некорректную последовательность активизации компонент ОС. В связи с этим должны быть приняты меры, гарантирующие пассивность органов управления в период отработки последовательности Z_L (например, аппаратная блокировка клавиатуры с момента активизации модифицированного BOOT до момента окончания активизации субъектов множества Sz).

Описанный метод позже был реализован во внешних носителях типа CD-ROM, которые позволили значительно (на два порядка) увеличить информационную емкость носителя и загружать с него развитые операционные среды типа OS/2. Однако однократность записи существенно снижает гибкость построения ИПС таким методом.

Неудобство использования загрузочной дискеты и ее быстрый износ обусловили возникновение следующего способа проектирования ИПС.

Откажемся от рассмотрения загрузочной дискеты и рассмотрим ПЭВМ с загрузкой ОС с устройства локального хранения (винчестера) и дополнительным аппаратным устройством изолирования среды.

Рассмотрим 2 этапа – этап установки ИПС и этап эксплуатации ИПС. Предположим существованием N пользователей, каждый i-й из которых характеризуется некоторой персональной информацией K_i, не известной другим пользователям и хранящейся на некотором материальном носителе (например, устройстве сенсорной памяти типа Touch Memory). Существует также администратор системы с ИПС, который знает все Ki и единолично проводит этап установки. Пользователи (владельцы K_i) же участвуют только в этапе эксплуатации.

Процесс установки ИПС состоит из следующих действий:

1. В ПЭВМ устанавливается аппаратный модуль, включающий в себя устройство и программы ПЗУ данного устройства (субъекты аппаратно-программного уровня), реализующие:

• операции сервиса аутентифицирующего носителя пользователя K_i (как минимум, его чтение),

• аутентификацию пользователя с номером i по введенному им K_i,

• чтение массива данных, содержащего множество доступных для пользователя i объектов-источников (исполняемых модулей) F_i1, F_i2, ..., F_im, составляющих O_Z, а также объект, содержащий Z_L,

• вычисление информации Mi1, Mi2, ... Mim, фиксирующей целостность объектов-источников F_i1, ..., F_im каждого объекта-источника (информация Mij должна удовлетворять требованиям к хеш-значениям и, возможно, зависеть от K_i), $Mij=H(K_i, F_j)$.

• блокирование устройств управления и предотвращение загрузки операционной среды с внешнего носителя.

2. Администратор определяет для пользователя i набор потенциально возможных для активизации субъектов E_i, $E_i=\{Pi1, ..., Pimi\}$, i=1, ..., N.

Create(Pik,F**j**)->Pij ,

mi – число разрешенных к запуску задач для i-го пользователя.

3. Администратор формирует (и заносит на носитель) или считывает с носителя для i-го пользователя его K_i и вычисляет значения для последующего контроля целостности $M_{ijr} = H(K_i, F_{jr})$, где H – функция КЦ (хеш-функция).

4. Администратор проделывает действия 2 и 3 для всех N пользователей.

5. Администратор устанавливает в КС МБС с объектом-источником $F_{ипс}$ и фиксирует его целостность. Установка модуля происходит с учетом условий утверждения 5.

6. Администратор фиксирует целостность объекта, содержащего Z_L.

Процесс эксплуатации состоит из следующих действий.

1. Включение питания и активизация аппаратного модуля:

а) Идентификация пользователя i по K_i.

При успехе выполняется п. б), при неудаче ПЭВМ блокируется.

б) Проверка целостности всех установленных в ПЭВМ ПЗУ.

При положительном исходе выполняется п. в), при неудаче ПЭВМ блокируется.

в) Чтение по секторам файлов операционной среды и проверка их целостности.

г) Чтение как файла $f_{ипс}$ (с помощью функций операционной среды) и проверка его целостности. Вариантом может быть чтение Fипс по секторам.

д) Активизация процесса контроля Рипс. ***Create***(S_x,Fипс)->Рипс. Активизация МБО.

е) Запуск избранной задачи i-го пользователя (может не выполняться).

2. Работа в ИПС.

Запуск каждого процесса Ps сопровождается проверками:

а) Принадлежит ли Fs к множеству разрешенных для i (E_i), если да, то выполняется п. б), иначе запуск игнорируется.

б) Совпадает ли $G=H(K_i, Fs,)$ с $M=H(K_i, Fs)$, вычисленной администратором.

в) При положительном исходе б) задача запускается, иначе запуск игнорируется.

Легко видеть, что условия изолированности среды выполнены.

Кроме того, в данном случае реализован механизм ступенчатого контроля, обеспечивающий чтение реальных данных.

При дополнении в ИПС реализации МБО и выполнении условий к субъектам, входящим в ИПС, сформированная программная среда будет гарантировано защищенной в рамках политики безопасности, реализованной в МБО.

Используя утверждение 4 об одинаковости состояний КС после активизации проверенных на неизменность субъектов в неизменной последовательности, можно описать метод доверенной загрузки компонент операционной среды (кратко «метод доверенной загрузки»).

Пусть предопределен порядок загрузки компонентов ОС (под загрузкой компонентов ОС понимается активизация различных субъектов ОС из соответствующих объектов-источников различного уровня иерархии). Процедуру загрузки ОС назовем доверенной, если:

• установлена неизменность **компонент ОС** (объектов), участвующих в загрузке (иными словами объектов, принадлежащих множеству O_Z), причем неизменность установлена до порождения первого субъекта из Z_L,

• установлена неизменность объектов, определяющих **последовательность активизации компонент ОС** (с учетом нескольких уровней иерархии), неизменность обеспечена в течение заданного интервала времени; состояние указанных объектов не может быть изменено никем, кроме предопределенного пользователя (пользователей) КС (это условие соответствует неизменности последовательности Z_L).

Легко видеть, что процедура доверенной загрузки обеспечивает одинаковое состояние КС после выполнения загрузки (согласно утверждению 4).

Основная техническая проблема при реализации доверенной загрузки состоит в доступе к объектам высшего уровня иерархии ОС (файлам) до загрузки ядра данной ОС (загружаемую ОС далее будем называть пользовательской). Однако при возможности генерации ИПС для какой-либо иной ОС (далее будем называть ее базовой) можно предложить итеративную реализацию доверенной загрузки с использованием ресурсов указанной ОС.

Рассмотрим реализацию доверенной загрузки ОС на основе генерации ИПС для одной из операционных сред вычислительной системы.

Предположим, что имеется базовая операционная система, для которой возможна полноценная генерация ИПС. Пусть в вычислительной системе существуют еще операционные системы OS_1, OS_2, ..., OS_n. Ставится задача доверенного запуска операционной среды OS_j. Пусть в базовой операционной системе имеется некоторое условно называемое «шлюзовое ПО» между базовой операционной системой и OS_j. Функции шлюзового ПО заключаются в обеспечении доступа к файловой системе операционной системы OS_j (т.е. объектам уровня R).

Пусть также пользователь i имеет физический доступ к комплекту технических средств (рабочему месту) сети (ЭВМ) Tm, на котором установлена операционная система OS_j. При использовании комплекта T_m пользователем i.

1. Происходит аутентификация пользователя i (по его индивидуальной информации).

2. Проверяются права пользователя по использованию аппаратной компоненты комплекта T_m.

3. Контролируется целостность (на основе информации пользователя K_i либо без нее) всех объектов базовой ОС, размещенных на некотором носителе локально или удаленно (через технические средства ЛВС) связанном с Tm.

4. Загружается базовая операционная система и контролируется целостность шлюзового ПО.

5. Загружается шлюзовое ПО (при этом становится доступным как минимум в режиме чтения файловая структура OS_j, размещенная локально на Tm).

6. Контролируется целостность объектов уровней, меньших Rj (Rj – максимальный уровень представления объектов в OS_j) для OS_j (см. выше).

7. Контролируется целостность объектов уровня Rj (файлов) OS_j.

8. Контролируется целостность объекта, задающего последовательность загрузки компонент.

9. Осуществляется принудительная загрузка (инициируется предопределенный в силу целостности объектов O_Z и последовательности Z_L порядок загрузки компонент ОС) проверенной на целостность OS_j.

Легко показать, что при дополнительных требованиях к доверенно загруженной ОС (а именно – гарантированном включении механизма контроля запуска задач и качественном (в смысле перехвата всех запросов на запуск ядром ОС) контроле запуска задач, а также при корректном управлении списком разрешенных задач для загруженной при помощи описанного алгоритма ОС) порождается изолированная программная среда.

Утверждение 8 (условия генерации ИПС при реализации метода доверенной загрузки).

Пусть ядро ОС содержит МБО и МБС, инициируемые в ОС субъекты попарно корректны и их объекты-источники принадлежат множеству проверяемых на неизменность в ходе доверенной загрузки, МБО запрещает изменение любого объекта-источника и выполнена процедура доверенной загрузки ОС. Тогда после инициирования ядра ОС генерируется ИПС.

Доказательство

Процедура доверенной загрузки по построению обеспечивает неизменность O_Z и Z_L, по условию утверждения для порождения субъектов разрешены только объекты-источники, принадлежащие O_Z, неизменность объектов-источников по условию гарантируется свойствами МБО. Следовательно, выполнены условия утверждения 5 и генерируется ИПС. Утверждение доказано.

3.4. Опосредованный несанкционированный доступ в компьютерной системе. Модель опосредованного НСД

Опосредованный несанкционированный доступ (ОНСД) понимается как действия (инициирование потоков) некоторого субъекта в рамках

поддерживаемой МБО политики безопасности, которые либо создают новые объекты в КС, либо изменяют ассоциированные объекты активных субъектов; существенно важно, что данные действия осуществляются указанным субъектом **автономно** (без управления каким-либо пользователем).

В дальнейшем с использованием измененных, либо порожденных субъектов пользователь через другие субъекты инициирует потоки, принадлежащие N, либо порождает субъекты, не принадлежащие множеству разрешенных. Указанный автономно работающий внедренный в КС субъект, реализующий ОНСД, называется разрушающим программным воздействием (РПВ) или программной закладкой.

Необходимые условия для осуществления ОНСД:

1. Наличие во множестве субъектов КС субъекта, который может изменить ассоциированные объекты (функционально ассоциированные или ассоциированные объекты-данные) других субъектов (модель действий – искажение), либо инициировать потоки от ассоциированных объектов некоторых субъектов к своим объектам (в смысле созданных или изменяемых данным субъектом) (модель действий – перехват).

2. Внедрение в систему измененных объектов-источников, которые могут породить субъекты со свойствами п.1.

Существует событийная модель воздействия РПВ на КС[1].

В КС возможны следующие комбинации несанкционированных действий, производимых субъектом РПВ – несанкционированное чтение (НСЧ) – поток от внешнего по отношению к РПВ объекту к ассоциированным объектам РПВ и несанкционированная запись (НСЗ) – поток от ассоциированных объектов РПВ к внешним объектам (заметим, что внешние для РПВ объекты могут являться ассоциированными объектами других субъектов. Объединим избранное множество субъектов в КС в один субъект (назовем его прикладной программой (ПП)). Рассмотрим также множество санкционированных действий прикладной программы по записи (СЗ) или считыванию (СЧ). Событийная модель описывается полной группой событий (событием считается поток) в КС в паре «РПВ – другие субъекты КС (ПП).

Таблица 3. Полная группа событий в системе «ПП-РПВ»

Ситуации	НСЧ	НСЗ	Действия	СЧ	СЗ
1	0	0	нет	0	0
2	0	0	нет	0	1
3	0	0	нет	1	0
4	0	0	нет	1	1
5	0	1	изменение (разрушение) кода прикладной программы в оперативной памяти	0	0
6	0	1	разрушение или сохранение выводимых прикладной программой данных	0	1

1. А.Щербаков. Разрушающие программные воздействия. М: Эдель – Киев: Век, 1993. – 64 с.

7	0	1	разрушение или сохранение вводимых прикладной программой данных	1	0
8	0	1	разрушение или сохранение вводимых и выводимых данных	1	1
9	1	0	нет	0	0
10	1	0	перенос выводимых прикладной программой данных в ОП	0	1
11	1	0	перенос вводимых в прикладную программу данных в ОП	1	0
12	1	0	перенос вводимых и выводимых данных в ОП	1	1
13	1	1	процедуры типа «размножение вируса» (действия закладки независимо от операций прикладной программы)	0	0
14	1	1	аналогично	0	1
15	1	1	ситуациям	1	0
16	1	1	6 – 8	1	1

Ситуации 1 – 4 соответствуют нормальной работе прикладной программы, когда закладка не оказывает на нее никакого воздействия.

Ситуация 5 может быть связана с разрушением кода прикладной программы в оперативной памяти (ОП) ЭВМ, поскольку санкционированных действий по записи и считыванию прикладная программа не выполняет, либо с сохранением уже накопленной в ОП информации.

Ситуация 6 связана с разрушением или с сохранением записываемой прикладной программой информации (искажение или сохранение выходного потока).

Ситуация 7 связана с сохранением считываемой прикладной программой информации (сохранение входного потока).

Ситуация 8 связана с сохранением информации закладкой при ее считывании или записи прикладной программой.

Ситуация 9 не связана с прямым негативным воздействием, поскольку прикладная программа не активна, а закладка производит только НСЧ (процесс «настройки»).

Ситуация 10 может быть связана с сохранением выводимой информации в оперативную память.

Ситуация 11 может быть связана с сохранением вводимой информации в оперативную память, либо с изменением параметров процесса санкционированного чтения закладкой.

Ситуация 12 может быть связана с сохранением как вводимой, так и выводимой прикладной программой информации в оперативную память.

Ситуация 13 может быть связана с размножением закладки, сохранением накопленной в буферах ОП информации или с разрушением кода и данных в файлах, поскольку прикладная программа не активна.

Ситуации 14 – 16 могут быть связаны как с сохранением, так и с разрушением данных или кода и аналогичны ситуациям 6 – 8.

Необходимо заметить, что условие взаимной корректности субъектов запрещает существование РПВ с возможностью записи в ассоциированные объекты не совпадающих с ним субъектов. С другой стороны, инициирование потока от ассоциированного объекта к ассоциированным объектам РПВ не противоречит условию корректности. Предположим, что в КС отсутствует субъект с указанными возможностями (чтение ассоциированных объектов). Тогда верно утверждение.

Утверждение 9 (условия невозможности опосредованного НСД в ИПС).

В ИПС с контролем целостности объектов-источников при порождении субъектов ОНСД невозможен.

Доказательство

По условию ИПС содержит только взаимно корректные субъекты, следовательно, изменение ассоциированных объектов у любых субъектов невозможно. Далее, порождение субъекта из измененного объекта-источника или порождение субъекта, не принадлежащего множеству субъектов ИПС также невозможно по условию. Необходимые условия ОНСД не выполнены, ОНСД невозможен. Утверждение доказано.

В условии сделано предположение об отсутствии субъекта чтения ассоциированных объектов. Для гарантий отсутствия произвольного ОНСД необходима коррекция начальных условий замыкания субъектов в ИПС, либо изменение алгоритмов работы МБО для предотвращения указанных выше действий РПВ.

Сделаем важное замечание. Перенос информации от ассоциированных объектов любого субъекта к ассоциированным объектам РПВ имеет смысл с точки зрения ОНСД только в том случае, когда эта информация будет отображена в объект внешнего хранения (иными словами, сохранена). В ином случае злоумышленные действия невозможны. С другой стороны, в реальных КС значительно число субъектов, участвующих в потоках информации (драйверы дисков и ЛВС и др.), которые переносят информацию от ассоциированных объектов, поэтому требования по отсутствию операций чтения ассоциированных объектов нереальны. Следовательно, необходимо противодействовать операциям сохранения. Как было сказано выше, данный вопрос можно решать на уровне анализа базового ПО, замыкаемого в ИПС (поиск и редуцирование действий по сохранению считанных данных), либо на уровне коррекции ПБ в части запрета записи в объекты внешнего хранения.

3.5. Новые подходы к созданию изолированных сред. Виртуализация[2]

Одной из основных функций операционной системы является унификация способа взаимодействия пользовательских программ с аппаратурой, в частности, создание такого способа взаимодействия с устройством, при котором все специфические операции скрываются от пользователя.

Таким образом, ОС дает прикладным программам возможность исполняться, предоставляя им некоторый набор API (application program interface – прикладных программных интерфейсов). При помощи этого

2. Параграф написан при участии А.Тормасова

набора программа прямо или опосредованно обращается к внешним устройствам и файловой системе, посылает и получает сетевые пакеты данных, создает и удалять процессы (субъекты – см. выше), обменивается с ними информацией и т. п. А чтобы предоставлять такие возможности, ОС должны уметь работать с физическими устройствами.

Другой функцией ОС является унификация способа взаимодействия пользовательских программ с аппаратной частью и при этом обычно применялась какая-либо реализация идеи виртуализации. Например, программа пользователя, работающая с файловой системой, размещенной на дисковом накопителе, видит его как некий обобщенный или виртуальный диск, не зная о технических деталях реализации того, что лежит “снизу”, на уровне аппаратной части компьютера – скажем, SCSI это или IDE-диск, а, может быть, вообще USB флэш-устройство.

Другой смысл термина “виртуальный” – несуществующий физически. Например, виртуальная память процесса противопоставляется физической. Впрочем, и первый пример с дисками можно охарактеризовать как имеющий отношение к “несуществующему” виртуальному устройству, с которым работает сервис ОС, обеспечивающий возможности файловой системы поверх виртуального, обобщенного блочного хранилища данных с прямым доступом. Такое содержание термина “виртуальный” отражает один из доминирующих сейчас принципов организации ОС.

До настоящего при реализации ОС виртуализация как подход наиболее активно использовалась на уровне именно устройств. Если рассмотреть традиционную иерархию понятий: нижний уровень – уровень прямого доступа к оборудованию, средний уровень – драйверов устройств, предоставляющих обобщенный интерфейс управления ими, и верхний уровень – набор сервисов ОС, предоставляемых уже приложениям пользователя, то можно отметить, что наиболее активно виртуализация применялась на среднем уровне. Когда появлялось новое устройство, то единственный способ его поддержки для приложений обычно сводился к тому, чтобы получить вместе с устройством некий “среднеуровневый” драйвер ОС, который непосредственно управлял самим устройством в “аппаратных” терминах (таких, как команды in/out и т. д.), предоставляя “наружу” возможность управления в терминах API (например, в виде графического драйвера, имеющего стандартный набор функций для рисования точек, линий, поверхностей и т. д.). Причем с точки зрения ОС этот интерфейс действительно был универсален для разных адаптеров устройств, т. е. в системе как бы было установлено одно или несколько “виртуальных” устройств, предоставляющих определенный сервис (в данном случае рисование изображения).

Тем не менее существовали и другие подходы, на других уровнях – верхнем и нижнем. Исторически сначала появился подход низкоуровневой виртуализации – когда ОС “подставлялось” виртуальное, а не физическое устройство.

Если рассматривать ОС как единый объект, обладающий некими “органами чувств”, т. е. средствами общения с внешним миром, то для нее единственным способом определить наличие и состояние какого-либо объекта остаются стандартные действия, описываемые обычно производителем аппратной части – начиная от детектирования подключения этого устройства (сейчас это обычно выполняется за счет специального

протокола Plug-and-Play) до возможности протестировать оборудование путем выполнения каких-то конфигурационных действий (например, можно отправить на последовательный порт, к которому, как предполагается, подключен модем, текстовую строку “AT” и ожидать, что, если модем работает, обратно придет текстовая строка “OK”).

В таких условиях появляется возможность создать своего рода «виртуальную реальность» для «органов чувств» ОС. Если сымитировать ответы физических устройств программным способом, то и получится “виртуальная реальность”, состоящая из виртуальных моделей физических устройств, которые дают вполне осмысленные ответы на запросы ОС и поведение которых максимально приближено к оригиналу. Иными словами, получится некий эмулируемый компьютер – виртуальная среда – набор виртуального оборудования, которое практически полностью эквивалентно такому же набору физического оборудования. Естественный следующий шаг – установка самой обычной ОС внутрь такого “виртуального устройства” и использование его для предоставления какого-либо сервиса.

Одной из первых по этому пути пошла компания IBM с мэйнфреймами, ОС OS/3x0 (последняя из них – OS/390), ОС VM и системой LPAR (Logical partitioning). Пользователь получал в свое распоряжение полноразмерный и полнофункциональный виртуальный компьютер, на который он мог поставить собственную версию ОС и установить собственное прикладное ПО. В этом компьютере имелась оперативная память, возможность использования ресурсов процессора, собственные виртуальные периферийные устройства (диск или сетевая карта) – практически все то, чем обладает обычный компьютер, только в виртуальном виде. Число виртуальных компьютеров, обслуживаемых на одном аппаратном компьютере, зависело от многих факторов, в частности, от лицензии, доступных ресурсов памяти или диска, возможностей центрального процессора и т. д. На мощной физической машине можно было запустить множество виртуальных; например, IBM утверждает, что на S/390 реально запустить более десятка тысяч копий виртуальных компьютеров с ОС Linux.

Упрощенную версию такого подхода (а может быть, и усложненную, это зависит от точки зрения) представляют собой эмуляторы ПК, распространенные на обычных IBM PC-совместимых компьютерах, – системы виртуальных машин (Virtual Machine, VM). Они реализуют аналогичный подход на традиционной архитектуре Intel x86.

Существует несколько проектов эмуляторов, использующих разные технологии и подходы и находящиеся на разных этапах своего жизненного цикла. Это и коммерческие реализации, такие, как решение, активно использующее аппаратную поддержку виртуализации компании Parallels (www.parallels.com), Microsoft Virtual PC/Virtual Server/Hyper-V (http://www.microsoft.com), EMC VMware (http://www.vmware.com), и открытые – например, проекты системы паравиртуализации Citrix/Xen (http://www.cl.cam.ac.uk/Research/SRG/netos/xen), Plex86 (http://www.plex86.org) и UML (http://usermodelinux.org), openvz.org для ОС Linux. Практически все они (кроме openvz) включают так называемый VMM – монитор виртуальных машин, который обслуживает запросы виртуальных машин, выполняет эмуляцию памяти, разделение доступа к ресурсам, изоляцию

и т. д. Сам VMM обычно загружается через так называемую Host OS (основную ОС) – ту копию ОС, которая сама непосредственно работает с физическими устройствами. Другим способом реализации того же подхода является использование так называемого «гипервизора» (являющегося синтезом мониторов объектов и субъектов), который управляет доступом к ресурсам и обычно использует VMM на другом уровне привилегий, «депривилегируя» даже основную (host) ОС (проект Xen, Parallels Server, ESX server и другие).

Эмулятор компьютера представляет собой некий процесс, внутри которого осуществляется моделирование набора стандартных устройств компьютера — памяти, сетевой платы, графической платы, клавиатуры, мыши и т.д. Особо стоит отметить, что кроме устройств эмулируется и BIOS, поэтому в этот процесс можно ставить полнофункциональную операционную систему, которая будет работать «не зная» о том, что она находится внутри эмулятора. Более того, на не-Intel платформах существовали и активно использовались до последнего момента системы, эмулирующие систему команд Intel — например, такой эмулятор входил в состав поставки Windows NT для процессора Alpha компании DEC (www.insignia.com/about/default.asp#1).

Принцип использования ресурсов основной ОС близок к идее разделов: такие ресурсы, как оперативная память компьютера и дисковое пространство, делятся между экземплярами запущенных эмуляторов, практически исключая совместное использование. Копия ОС, вернее, набора ее файлов, необходимых для функционирования ОС, у каждого экземпляра виртуального компьютера будет своя. Особая сложность – эффективное использование собственной оперативной памяти виртуального компьютера, поскольку ее практически никак нельзя разделить между разными экземплярами VM эмуляторов. Отдельную проблему в такого рода решениях создает и тот факт, что ОС внутри эмулятора ничего не знает о существовании внешней ОС и, например, считает своим долгом в случае отсутствия активности своих процессов загрузить процессор выполнением процесса типа idle (в Unix). Еще одна проблема – двойное кэширование данных: обращение к данным пытаются кэшировать обе ОС, что явно не повышает производительность. Надо отметить, что другие ресурсы, например, сетевая плата или процессоры, вполне корректно делятся между всеми экземплярами эмуляторов.

В результате работа одного ядра ОС внутри другого приводит к существенной потере производительности системы в целом. Такая реализация системы не позволяет запускать достаточно большое число экземпляров виртуальных машин на одном компьютере; практически допустимое их число лежит в диапазоне от одной до четырех-пяти на типовую рабочую станцию.

Технически реализация подобных VM-систем в архитектуре x86 – весьма непростая задача, так как при проектировании Intel не предусматривала подобных задач. В результате часть команд, которые могут выполнять ОС, не должны быть разрешены для исполнения внутри VM. Самым простым способом решения этой проблемы была бы возможность получать так называемые прерывания или исключения при попытке выполнения этих команд внутри VM, с тем чтобы обрабатывать их внутри VMM. Но проблема в том, что простого способа вызвать это

исключение для некоторого класса команд не существует. Они просто отработают внутри VM, дав при этом не тот результат, которого ожидали разработчики ОС, причем иногда выполнение такой небезопасной команды может вызвать сбой в основной ОС или даже зависание всего компьютера – например, команд ввода вывода in/out. Кроме собственно команд типа in/out, требующих особой обработки, требует особого внимания и команды, обращающиеся к заранее неизвестным областям памяти (скажем, возврата из функций по заранее неизвестному адресу в стеке или перехода по регистру), которые должны обрабатываться в процессе работы эмулятора. Существенную долю процессорного времени система в процессе исполнения так называемого «гостевого кода» (кода, находящегося внутри VM) тратит на разрешение подобного рода проблем. Они решаются разным образом – от бинарной трансляции исполняемого кода перед его реальным исполнением, как в технологии VMware, до обработки специальных отладочных прерываний или эмуляции команд, как в технологии компании Parallels, или путем модификации ОС, расположенной внутри VM (подход паравиртуализации Xen).

Недавно компания Intel анонсировала поддержку виртуализации инструкций, вернее, возможность специального способа исполнения инструкций, удобного для реализации таких эмуляторов, во всех типах своих новых процессоров архитектур IA-32 и IA-64 – как для серверов, так и для рабочих станций (технология под названием Intel Virtualization Technology VT). Все новые процессоры Intel будут выходить с этой поддержкой, начиная с 2006 года. Аналогичные планы анонсировала и компания AMD (технология под кодовым словом Pacifica или SVM – secure virtual machine). Подобная поддержка теоретически упростит создание аналогичных программных продуктов и повысит их эффективность (хотя вопрос о реальной степени ускорения процедур эмуляции архитектуры остается пока открытым). Тем не менее, этот анонс ведущего производителя процессоров говорит о том, что в использование технологий VM становится повсеместным, от ноутбуков до высокопроизводительных серверов. Большинство мировых аналитиков сходится в мнении что в той или иной степени виртуализация будет присутствовать абсолютно всюду, в любых вычислительных устройствах от самых простых до самых сложных.

Рассмотрим теперь другой уровень виртуализации – высокоуровневую виртуализацию, или виртуализацию API. Чтобы разобраться в ней, стоит выяснить, зачем вообще нужна виртуализация.

Если говорить о нижнем уровне, то цель виртуализации “физического оборудования” может быть разной, но обычно такие VM используются для упрощения установки и развертывания (ведь все виртуальные эмуляторы обычно имеют одинаковую конфигурацию), например, при тестировании. Иногда VM также применяют для консолидации серверов или для поддержки старых приложений, которые не могут работать с новыми версиями ОС (именно для таких целей Microsoft предлагает использовать свой Virtual Server). Виртуализация также удобна для отладки программ, опробования новых версий “заплаток” (не “сломают” ли они имеющиеся программы) и т. д.

Если взять средний уровень, там цель, казалось бы, очевидна – надо уметь использовать в рамках одной ОС максимальное разнообразие

физических устройств, а следовательно, предоставлять пользовательским и системным процессам унифицированный интерфейс взаимодействия.

Но если вдуматься, для пользовательских процессов ОС, – которые, собственно, и предоставляют сервисы пользователям, – наличие "виртуального оборудования", в общем-то, безразлично, их интересует только, чтобы те способы коммуникации, которые они применяют для связи с ОС, отрабатывались наиболее удобным и эффективным образом. Вот так и появляется "высокоуровневая виртуализация", которая обеспечивает для каждой среды исполнения (будем называть ее виртуальной средой – VE) свое собственное уникальное изолированное окружение – свои файлы и другие ресурсы (в том числе системные), свои сервисы, свои системные способы связи с внешним миром и т. д.

Иными словами, с точки зрения приложения оно запускается в собственном компьютере с основной ОС, и пользователь может делать с этой средой (VE) все, что обычно может делать администратор машины, – запускать и останавливать приложения и системные сервисы, устанавливать обновления приложений, конфигурировать сеть и сетевой экран, перезапускать VE. От оборудования, как видно из ***рис. 4***, среда пользователя (VE) отделена специальным "уровнем виртуализации", вводящим понятие VE в корневую (или основную) ОС. Кроме того, появляются дополнительные возможности, которые не так просто реализовать на физическом компьютере или внутри VM – например, быстрая и эффективная установка приложений внутри множества VE, быстрая миграция (со временем недоступности сервера 1-3 с или даже вообще без остановки сервисов и прерывания сетевых сессий), динамическое управление ресурсами системы, выделяемыми конкретному VE и его процессам.

Для реализации подхода компанией Parallels (продукт Parallels Virtual Containers) была выбрана идея разделения доступа путем изоляции пространства имен (namespace). Все объекты уровня ядра ОС имеют какие то идентификаторы, позволяющие их разделять. Обычно это или имя объекта в виде текстовой строки (пример – имя файла или ключа в Windows registry), специальные выделенные идентификаторы (типа pid – process identifier – идентификатор процесс, или fid – file identifier – идентификатор файла в файловой системе), просто какие то числа, являющиеся обычно номерами строк в специальных таблицах (пример – handle – «номер» графического объекта в программе ОС Windows), или просто адрес в виртуальном адресном пространстве процесса, возвращаемый при создании структуры данных для объекта. Существенным является тот факт, что в контексте вызова используемый идентификатор является уникальным, то есть отвечающим за один определенный объект, работу с которым и подразумевает тот, кто использует идентификатор. Таким образом, если ктото (например, процесс пользователя) хочет работать с объектом, то он использует для этого его идентификатор в качестве адреса.

Все объекты со своими адресами и связанными с ними идентификаторами объединяются в группы (обычно иерархические) – пространства имен (namespaces). Внутри пространства имя объекта обычно уникально, но, в разных пространствах, разные объекты могут иметь одно и тоже имя.

Современные операционные системы имеют развитую структуру именования объектов, на которую замкнуты практически все их возможности. Так, чтоб, например, послать сообщение процессу, завершить его или изменить приоритет необходимо знать его идентификатор pid, являющийся параметром вызова соответствующего системного сервиса.

Идея виртуализации и изоляции на уровне пространства имен состоит в том, что мы создаем более высокий уровень иерархии адресов объектов (чем имеющиеся уже уровни в ОС), объединяем пространство имен в группу (назовем ее VEn, где n – порядковый номер создаваемой среды) и создаем группу процессов, для которых любые системные вызовы содержат параметры из группы VEn. Например, если раньше программа обращения к ключу регистра Windows в качестве параметра указывала имя ключа как «HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update» и получала к нему доступ, то, теперь, если она находится в контексте группы 101, то при использовании того же параметра она получит доступ (скрытым от нее образом, «она об этом знать не будет» к ключу «\\Machine\VE101\HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update». Таким образом, две таких программы, запущенных в разных VE 101 и 102, обращаясь к одинаковому имени ключа «HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update» получат доступ к разным экземплярам этого ключа («\\Machine\VE101\HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update» и «\\Machine\VE102\HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update»), что и обеспечит их изоляцию на уровне имени. Более того, не будет никакого способа добраться до соответствующего ключа второго VE из контекста первого (скажем, если программа попытается явно открыть «\\Machine\VE101\HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update» из контекста VE 102, то в реальности она будет пытаться добраться до ключа «\\Machine\VE102\Machine\VE101\HKEY_LOCAL_MACHINE\SOFTWARE\Apple Inc.\Apple Software Update»). Аналогичным образом осуществляется изоляция доступа и для других типов идентификаторов.

Предложенная модель виртуализации существенно отличается от существующих стандартных моделей низкоуровневой виртуализации оборудования (виртуальных машин – VM) тем, что предоставляя практически такие же возможности, как и VM, позволяет добиться существенно более высокой «плотности» размещения виртуальных серверов на оборудовании (в сотни раз выше), не теряя в уровне безопасности и изоляции, и более эффективна для запуска и миграции вычислительных процессов.

Существенным отличием предложенной модели виртуализации является разделение всеми VE одного ядра ОС, что накладывает дополнительные требования на технологии управления ресурсами ОС. Как показал опыт, даже в высокоразвитых коммерческих ОС имеющихся средств управления ресурсами недостаточно для групповых гарантий и лимитов выделения ресурсов потребителям, и особенно для осуществления взаимной изоляции виртуальных серверов по производительности (в частности, для защиты от случайных или намеренных атак типа «отказа в обслуживании»). Особо следует отметить, что использованные в существующих ядрах ОС алгоритмы управления ресурсами пред-

назначены для управления сотнями, максимум тысячами объектов, тогда как, например, для 25 тысяч виртуальных серверов, запущенных на одной машине (такой эксперимент проводился на стандартном сервере архитектуры Intel), общее количество процессов и потоков превысило 200 тысяч.

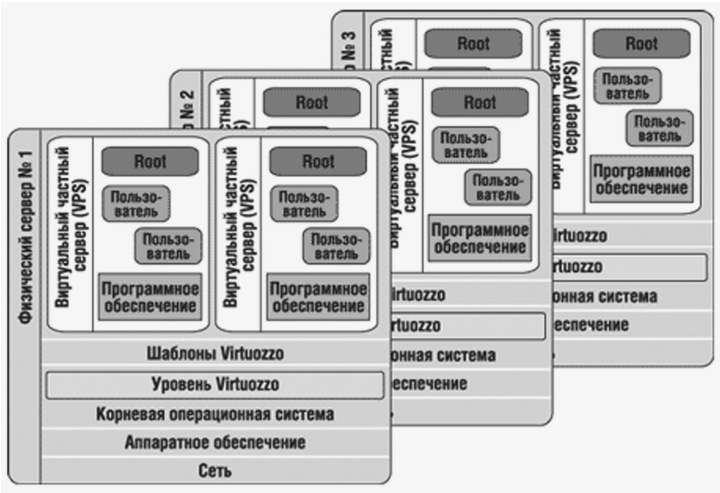


Рис. 4. Virtuozzo – реализация технологии виртуального приватного сервера VE

Технология VE находится "выше" технологии VM, их различие по отношению к архитектуре иллюстрирует следующий ***рис. 5.***

Рис. 5. Уровни реализации Virtual Machine и Virtual Private Server (на примере технологий VMware и Virtuozzo)

Какие же задачи можно решать при помощи виртуальной среды (VE) или виртуальной машины (VM)? Технически VM и VE представляют собой практически полноценный компьютер, т. е. их можно использовать

для всего того, для чего используют обычный компьютер. Наверное, проще будет описать для каждого способа те задачи, которые он решать не может или решает неэффективно.

Начнем с VM. Чем все же отличается VM и запущенная в ней программа от обычного компьютера? В первую очередь VM потребляет больше ресурсов, так как у нее есть накладные расходы (и немалые) на обслуживание собственно самой VM, VMM, Host OS и т. д. Это означает, что приложение, запущенное в VM, получит в свое распоряжение меньше ресурсов, чем такое же приложение, запущенное в такой же среде, установленной не внутри VM, а внутри обычного "железного" компьютера. Потери могут быть довольно велики – до 20-30% ресурсов процессора (в зависимости от нагрузки), минимум 20% физической памяти (обычно больше); может также быть ограничено число процессоров, видимых внутри VM, а при наличии сколько-нибудь экзотического оборудования – доступ к нему. Например, если на вашем компьютере оказалась карта со специальным видеоускорителем или современная сложная карта SAN, с большой вероятностью использовать их внутри VM так, как хотелось бы, вам не удастся.

Естественное следствие таких ограничений – невозможность массового использования множества VM на одном физическом компьютере. Обычно допустимое число VM составляет от 2-4 на обычной машине до 50 на высокопроизводительных серверах (и то только в одной определенной версии VM-системы, в типовом случае сервер может обслужить десяток машин).

Другим, достаточно неочевидным, ограничением может стать потребность в лицензиях. Многие приложения и системы сейчас лицензируются для определенного числа копий бинарных файлов, загруженных в память (такова, например, ситуация с Microsoft Windows Server 2003), и для каждой их копии в VM необходимо купить отдельную лицензию – а их общая цена может многократно превышать стоимость оборудования, использованного для данного физического сервера!

Ограничения на использование технологии VE – несколько другого рода и связаны обычно со способом ее реализации. Например, технология Virtuozzo подразумевает, что каждый запущенный на машине VE использует одно и то же ядро и изнутри VE нельзя менять существенные компоненты или версию ядра. Впрочем, это не означает, что нельзя применять, скажем, разные "заплатки" на разные VE, если они не затрагивают собственно ядра или модулей в Linux либо существенных библиотек или исполняемых файлов ядра Microsoft Windows. Других существенных ограничений, в общем-то, нет (за исключением прямого доступа к оборудованию, который опять-таки ограничен изнутри VE соображениями безопасности).

Есть еще некоторые ограничения, присущие той или другой реализации, – например, в VE в любой момент обычно возможен прямой административный доступ к файлам из основной ОС, тогда как в типовых VM-реализациях он возможен только по сети, если это сделал администратор соответствующей VM; и наоборот, в VM есть возможность сделать "снимок" состояния всей системы для дальнейшего возвращения к этому состоянию (snapshot), а в VE обычно это делается по-другому.

На самом деле с точки зрения пользователя виртуализация сама по себе является чисто технической составляющей решения. Конечному

пользователю от вычислительных систем нужно обслуживание, то есть набор сервисов, плодами работы которых он мог бы пользоваться. В реальности человека чаще всего интересует не возможность, например, пользоваться MS Word, а возможность получения текстового документа, который можно отправить по почте или напечатать, а адресат сможет его прочитать.

Виртуализация помогает решению такого рода задач, упрощая конфигурирование, развертывание и обслуживание реальных сервисов.

Используя виртуализационный подход можно отделить «суть» от «техники реализации», поместив сервисы, оказывающие услуги пользователям, внутри виртуализированного окружения, и отодвинув машинно- и коммуникационно- зависимые части вовнутрь средств поддержки функционирования виртуализации.

Именно это и является причиной появления и активного использования средств виртуализации – решающим здесь оказывается «экономический эффект», который можно получить от виртуализованных систем – дающий как некоторые новые возможности (например, по мобильности сервисов путем переноса их с одного физического компьютера на другой), так и уменьшение общей стоимости обладания системы.

Тем не менее, виртуализация, конечно, не является «панацеей» и полным решением задачи предоставления сервисов для конечных пользователей.

Промышленная реализация технологии для основных коммерческих ОС (технология Virtuozzo® компании Parallels – для Microsoft Windows, Linux, Unix) подтвердила, что уровень накладных расходов по сравнению с виртуальными машинами оказался в 3-10 раз ниже из за высокой эффективности разделения ресурсов ядром ОС и управления ресурсами.

Типичная IT организация сейчас должна уметь управлять и наращивать сложной гетерогенной средой, состоящей из множества компьютеров с Windows, Linux, UNIX и др. Массовый переход на использование многоуровневых приложений обнаружил серьезную проблему – большинство приложений сейчас устанавливается с по принципу «одно приложение на один сервер в один отдел» (далее «принцип 1-1-1»). Более того, развертывание решений на так называемых «тонких клиентах» типа предоставляемых компанией Citrix или Microsoft (Microsoft Terminal Server), только увеличивает количество выделенных серверов которыми должен управлять IT отдел. Эти условия осложняются часто ситуативным увеличением количества приложений необходимых для работы компаний.

С ростом сложности и объема, стоимость управления такой структурой растет буквально экспоненциально. В результате, организации становятся обладателями дорогой, негибкой и сложной инфраструктуры.

Несмотря на развитие аппаратных средств, цена оборудования остается практически одинаковой. Аппаратура становится все более мощной, но объем затрат на нее почти не меняется. Казалось бы, что собственно цена оборудования, не является большой проблемой для предприятий, однако увеличение уровня недогрузки и стоимость обслуживания все же начинают быть все более и более обременительными.

Стоимость программного обеспечения. Типичный сервер нуждается в лицензиях и поддержке со стороны производителя операционной системы и со стороны каждого производителя приложений. Обычно, общая стоимость программного обеспечения превышает стоимость оборудования сервера, на котором оно запущено, более чем в несколько раз.

Цена управления. Эта статья расходов является наиболее весомой для типичной IT организации. Персонал IT отдела корпорации должен обновлять сбоящее ПО, чинить аппаратуру и осуществлять резервное копирование данных. Отделу IT приходится следовать за увеличивающимся потоком запросов, хочет он этого или нет.

Можно утверждать, что типичная инфраструктура современного IT предприятия испытывает проблемы, связанные со следующим набором факторов.

Высокой сложностью. Большинство отделов IT поддерживает множество ОС. Только Windows широко распространена в 4 основных выпусках и 4 разных изданиях, сервера на базе Linux и UNIX так же создаются с использованием множества типов и версий, предприятия обладают тысячами физических серверов.

Неполной информацией о состоянии сети. Ситуативность принятий решений об установке новых серверов и руководство принципом 1-1-1 приводит к непредсказуемости времени закупок оборудования, а специализированные базы данных, отражающие информацию об используемом оборудовании, его статусе и принадлежности практически не ведутся.

Низким уровенем сервиса и высоким временем простоя серверов. Сейчас единственный способ гарантировать затребованный приложением ресурс – рассмотреть «худший случай» наибольшей нагрузки как основу для выбора покупаемого оборудования и надеяться на то, что в ПО нет критических ошибок, ведущих к утечке ресурсов. С другой стороны такой подход приводит к увеличению числа недоиспользуемых серверов, установленных по принципу 1-1-1.

Низкой гибкостью. Любые изменения в существующей сети приводят к отказу или плохой работе сервисов. Необходимость переноса сервера с приложениями, вызванное потребностями компании, часто приводит к весьма существенному простою. Создание нового сервера может занимать недели. Низкая гибкость обычно является прямым следствием высокой стоимости и высокой сложности.

Из изложенного выше следует, что наибольшего снижения стоимости IT инфраструктуры можно добиться снизив в первую очередь расходы на управление и оптимизировав политику приобретения ПО. При этом, рост стоимости оборудования менее чем в 2-3 раза будет приемлемым. Обычно размещение серверов в нескольких выделенных местах (центрах данных) оказывается первым логичным шагом к улучшению уровня управляемости инфраструктурой. Наряду с путем простой централизации, существуют и другие подходы к решению указанных проблем. Один из них – виртуализация.

Явным преимуществом виртуализации является высокая стандартизация и унификация «виртуальной среды», в которой функционирует гостевая ОС. Если на различных платформах есть одинаковый эмулятор, то с точки зрения гостевой ОС – эти компьютеры эквиваленты

и отличаются только такими параметрами как объем диска, скорость процессора, объем оперативной памяти. В таких условиях, однажды созданный и настроенный образ гостевой ОС с установленным ПО может быть использован везде, и, в конечном счете, даже намного пережить тот физический компьютер, на котором он был создан. Очевидно, что проблема обновления ПО в такой ситуации сводится к обновлению одного и только одного образа, который затем просто копируется везде, где он нужен. Значительно облегчается установка и запуск новых систем, восстановление после аварийных ситуаций, резервное копирование. Это, естественно, снизит затраты на управление, которые являются самыми весомыми из составляющих стоимости IT инфраструктуры.

Однако, у подобного подхода есть существенные недостатки – кроме собственно технических ограничений (например, не вся мощность нового компьютера может быть доступна немедленно гостевой системе, поддержка нового сложного оборудования требует не просто драйвера, но переписывания эмулятора и т.д.), существуют еще и ограничения в снижении стоимости обслуживания, так как каждый используемый образ гостевой ОС должен автономно управляться, т.е. линейно растет количество объектов управления.

Существует множество способов использования виртуализации в реальной среде и существующих центрах данных. Для лучшего понимания посмотрим на несколько типовых сценариев, в которых эта технология раскрывает свои преимущества для корпоративных пользователей:

- Консолидация серверов
- Автоматизация управления обновлениями.

Следует отметить, что существует еще множество высокоэффективных сценариев использования технологий виртуализации, приносящих высокую отдачу (особенно в области общей стоимости управления) – например, для управления ростом критических приложений и уменьшении времени простоя, обслуживание сложных децентрализованных запросов отделов или запросов на сложные конфигурации, создание легких и эффективных сред разработки и установки систем и др.

Итак рассмотрим консолидацию серверов. Стоимость серверного оборудования существенно падает, и большинство IT администраторов предпочитают использовать принцип 1-1-1. В результате обнаруживается, что стоимость управления и программного обеспечения много больше того, чем представлялось в начале. Эксперты оценивают, что стоимость оборудования обычно составляет что то около 10% от стоимости сервера приложения, причем собственно ПО в цене практически равно цене его администрирования – и все это растет пропорционально числу серверов. Консолидация серверов – это первый шаг к централизованному управлению инфраструктурой.

Консолидация серверов может быть интерпретирована по-разному. С точки зрения оборудования, для упрощения обслуживания стоит пытаться держать все «железо» в одном месте с высокой плотностью размещения. Именно для таких целей созданы так называемые blade-серверы (blade – “лезвие” в переводе). Но даже сложные системы управлением развертки серверов не могут дать существенной экономии в общей стоимости обслуживания – цена управления этими серверами все же остается существенной.

Технологии динамической виртуализации типа Virtuozzo™ могут дать значительно больше чем технологии консолидации оборудования, так как она предоставляет консолидацию на уровне ПО. Каждый VE предоставляет полный сервис, который ожидается от ОС обычного компьютера, то есть в VE можно запустить любое не модифицированное приложение, работающее в основной операционной системе – то есть, процесс перехода полностью прозрачен для конечного пользователя и не требует обращения его к IT персоналу. Каждый физический компьютер может нести сотни таких VE, и это предоставляет дополнительные возможности по консолидации логических серверов. Комбинация аппаратной и программной консолидации дает существенное уменьшение TCO, увеличивая уровень полезной загрузки серверов от типичных 5-15% до 60-80%, и все это без принесения в жертву качества обслуживания и внесения дополнительных потерь на обслуживание виртуализации.

При автоматизация управления обновлениями ПО консоль управления, созданная для Virtuozzo – VZMC. позволяет управлять сотнями VE, расположенными на одном физическом сервере, так же легко как и одним сервером, и множество физических серверов также может управляться централизованно. VZMC заменяет такие традиционные «примитивные» средства управления процессом установки и обновлений как обычный спредшит, на единый централизованный список версий ПО. Новые приложения могут быть установлены на все VE буквально несколькими нажатиями мышки. Темплиты ОС и приложений могут быть обновлены или наложены заплатки опять таки путем централизованного приложения темплитов к VE и серверам. Влияние таких изменений может быть предварительно опробовано путем создания копий VE, выполнения обновления, тестирования результата и мигрирования VE обратно в рабочее использование как только он будет к этому готов.

Суммируя все вышесказанное можно указать очевидные технические преимущества виртуализации, использующей подход, реализованный в технологии Virtuozzo:

• Повышение уровня используемости ресурсов сервера с обычных 5-10% до 80% наряду с очень низкой потребностью в ресурсах самого виртуализационного слоя.

• Немедленное создание и предоставление пользователям VE.

• Немедленное добавление ресурсов к уже существующим VE, вплоть до возможности получения всех ресурсов машины внутри одного VE.

• Упрощенное и ускоренное восстановление сервисов после сбоя.

• Практически нулевое запланированное и близкое к нулю незапланированное время простоя ресурсов.

• Однотипное представление и предоставление сервисов и ресурсов, мониторинга и управления на множестве платформ.

• Выполнение сложных запросов на изменения за секунды.

• При популярном сегодня принципе 1-1-1 консолидированные сервера для таких приложений могут использовать только одну лицензию на всех, т.к. физически будут исполняться на одном компьютере.

• Появляется смысл и даже потребность в хорошей, что зачастую синоним очень дорогой, аппаратной базе для работы консолидированных серверов, что означает снижение частоты сбоев аппаратуры из-за

низкого качества комплектующих. Как следствие, это приведет к значительному снижению стоимости обслуживания.

Резюмируя, заметим, что сфера категорий в общественном и научном сознании вокруг понятия «персональный компьютер» в настоящее время включает как минимум три важнейших аспекта:

- персональный компьютер (ПК) как системообразующий элемент организации личного и рабочего пространства;
- ПК как элемент вовлеченности в сетевое общество;
- ПК как «единство форм» – т.е. существующий в виде различных мобильных, настольных и стационарных устройств с общим набором функций и сервисов.

С точки зрения компьютерной безопасности данная ситуация является совершенно новой с точки зрения постановок задач и обладает рядом существенных отличий от классических моделей безопасности.

В первую очередь – это нестационарность субъектной среды компьютерной системы, т.е. возможность применения пользователем совершенно разных программ для обработки своей личной информации. Нестационарность субъектной среды порождает **принципиальную невозможность** гарантированного выполнения любой политики безопасности в КС.

Этот тезис вполне очевиден и обоснован выше. Кратко повторим – появление в программной среде априорно любой программы делает возможным две ситуации – изменение алгоритмов работы монитора безопасности, отвечающего за реализацию политики безопасности, либо реализацию потоков информации в обход монитора безопасности.

Для гарантированного выполнения политик безопасности используется описанные выше концепция и технологии создания изолированной программной среды, заключающаяся в том, что в компьютерной системе разрешается существование только таких программ, которые не имеют вышеперечисленных функций воздействия на монитор безопасности.

Однако для персонального компьютинга, для «домашнего компьютера» пользователя концепция изолированной среды несостоятельна, поскольку требует от пользователя не свойственного обычной технологии работы на ПК «добровольного самоограничения», либо присутствия некоторого администратора, который разрешает или запрещает использовать некоторые программы. И то, и другое достаточно утопично.

Синтез совершенно необходимой для обеспечения безопасности изолированной программной среды с привычной свободой использования приложений и сервисов, а также свободой использования сетевого ресурса приводит к необходимости формулирования новой концепции на уровне архитектуры построения сети и организации обработки данных.

Итак, необходимо сохранить полный набор привычных сервисов для обработки личной пользовательской информации. При этом на том же ПК должна быть предусмотрена возможность выполнения регламентированных некоторой корпоративной политикой безопасности действий, т.е. безопасной обработки некоторой служебной информации.

Из этих требований следует в первую очередь, что области личных данных пользователя и области корпоративных (защищенных) данных не должны пересекаться. Это легко доказать от противного, поскольку в

случае пересечения возможна активизация из личных данных программы в защищенной корпоративной среде и, как результат, нарушение заданой плитики безопасности.

Далее, личный сеанс и корпоративный сеанс работы на ПК должны быть разделены во времени и пользователи во время этих сеансов должны быть аутентифицированы по-разному.

Кроме того, поскольку ПК объединены в сеть и в некоторый интервал времени в сети могут работать как пользователи, занимающиеся личным задачами, так и пользователи, выполняющие служебные задачи, необходима криптографическая защита сетевого трафика, обеспечивающая его конфиденциальность, т.е. шифрование трафика сети.

Необходимо отметить, что, поскольку мы оперируем двумя сущностями в КС, а именно: личными и корпоративными данными, должна быть дополнительная политика безопасности, описывающая регламент обмена между этими данными. В вырожденном случае этот обмен полагается невозможным, однако, такой случай, по-видимому, на практике будет встречаться редко.

Таким образом, можно резюмировать, что для гарантий выполнения корпоративных политик безопасности необходима новая архитектура сети, включающей ПК, при которой:

- области личных данных пользователя и области корпоративных (защищенных) данных не пересекаются;
- личный сеанс и корпоративный (защищенный) сеанс работы на ПК разделены во времени и пользователи во время этих сеансов аутентифицированы по-разному;
- обеспечена криптографическая защита сетевого трафика, гарантирующая его конфиденциальность;
- сформулирована и реализована дополнительная политика безопасности, описывающая регламент обмена между личными и защищенными данными.

Отметим еще два важных свойства – шифрование трафика не должно быть подвержено влиянию ни из личной сессии работы ПК, ни из защищенной, и обмен между личными и защищенными данными как минимум должен регистрироваться.

4. БЕЗОПАСНОЕ СУБЪЕКТНОЕ ВЗАИМОДЕЙСТВИЕ И ИНФРАСТРУКТУРНЫЕ ВОПРОСЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

4.1. Введение

При рассмотрении моделей реализации политики безопасности считалось, что МБО имеет исчерпывающую информацию о текущем пользователе КС, либо субъекте, который инициирует потоки к объекту. В соответствии с этой информацией реализуется заданная при проектировании КС политика безопасности для конкретного пользователя системы. Следовательно, для обеспечения работы модулей реализации ПБ (МБО) в него необходимо передать информацию о текущем пользователе системы. Процесс «представления» пользователя (или субъекта) системе состоит из двух стадий: идентификации – пользователь сообщает свое имя (идентификатор) и аутентификации – пользователь подтверждает идентификацию, вводя уникальную, не известную другим пользователям информацию о себе (например, пароль). Очевидно, что для проведения процедур идентификации и аутентификации необходимо наличие соответствующего субъекта, затем необходима передача информации, касающейся безопасности между субъектами, отвечающими за безопасность (от модуля аутентификации к МБО). Кроме того, необходимо наличие соответствующих объектов, хранящих информацию, необходимую для аутентификации.

Важно заметить, что создание защищенной КС может происходить с учетом существования части механизмов безопасности, в связи с чем возникает задача сопряжения субъектов обеспечения ПБ и поддержания гарантий ПБ в различных элементах КС. Выше было показано, что для создания КС с гарантированно выполненной ПБ, необходимо реализовать МБО (поддерживающий эту ПБ), и МБС (гарантирующий ПБ) и замкнуть каким-либо описанным выше образом субъекты КС в ИПС. Практически в любой КС этого можно достигнуть, если субъекты (МБО и МБС) спроектированы и реализованы в виде исполняемых модулей конкретной операционной среды (ОС). Однако весьма часто в ОС уже присутствуют средства типа МБО с реализацией некоторой ПБ, несколько реже присутствует и МБС. С точки зрения оптимизации трудозатрат на реализацию защитных механизмов целесообразно максимально использовать средства, которые уже реализованы в КС, в необходимых случаях усиливая и дополняя их.

Итак, в данной части будут рассмотрены следующие группы важных теоретических проблем компьютерной безопасности.

1. Формализация процедуры аутентификации пользователей КС и описание ее характеристик.

2. Формализация процедур сопряжения субъектов (для решения задач передачи параметров от модулей аутентификации к модулям реализации и поддержания ПБ).

3. Формализация и описание процедур использования внешних субъектов для поддержания защищенности КС.

4. Методика анализа попарной корректности субъектов.

5. Защищенные хранилища аутентифицирующей и ключевой информации и инфраструктуры распределения такой информации.

4.2. Процедура идентификации и аутентификации

Учитывая, что пользователь КС только опосредованно работает с объектами и субъектами КС (через средства ввода и отображения информации), постулируем наличие как минимум двух аутентифицирующих пользователя объектов – внешнего аутентифицирующего объекта, не принадлежащий КС, и внутреннего, принадлежащего КС, в который переносится информация из внешнего объекта. Будем полагать, что внешний и внутренний аутентифицирующий объекты семантически тождественны – т.е. могут быть путем детерминированной процедуры приведены к тождественному виду (в виде слов в одном языке). Кроме того, полагаем наличие субъекта переноса информации от внешнего к внутреннему объекту (например, драйвер клавиатуры). Например, символьный пароль для входа в систему находится в «памяти пользователя», затем он путем набора на клавиатуре переносится в буфер программы запроса пароля (объект оперативной памяти ПЭВМ).

Опираясь на допущение о тождестве внешнего и внутреннего объектов далее будем изучать один из них – внутренний, возникающий после переноса информации извне КС.

Поскольку предполагается выполнение таких процедур, как идентификации и аутентификации, предположим, что i-й аутентифицирующий объект содержит два информационных поля IDi – неизменяемый идентификатор i-го пользователя, который является аналогом имени и используется для идентификации пользователя, и K_i – аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации.

На самом деле IDi может соответствовать разным пользователям, например, носитель сенсорной памяти Touch Memory (TM) содержит 8 байт неизменяемого идентификатора носителя, но при этом TM может быть передана разным пользователям.

Совокупную информацию в аутентифицирующем объекте будем называть ***первичной аутентифицирующей информацией*** i-го пользователя. Описанная структура соответствует практически любому устройству, служащему для опознания пользователя, например, упомянутый носитель сенсорной памяти типа Touch Memory (TM) имеет 8 байт неперезаписываемого неповторяющегося серийного номера, который однозначно характеризует конкретную TM и некоторый объект перезаписываемой памяти, который может содержать K_i. Аналогично для носителей типа пластиковых карт выделяется неизменяемая информация первичной персонализации пользователя, соответствующая IDi и объект в файловой структуре карты, содержащий K_i.

Очевидно, что внутренний аутентифицирующий объект не должен существовать в КС длительное время (большее времени работы конкретного пользователя). Далее, для постоянного хранения необходимо

использовать некую преобразованную информацию от первичной. Рассмотрим типовые схемы аутентификации.

Схема 1

В КС выделяется объект следующей структуры – эталон для идентификации и аутентификации (положим, что в системе зарегистрировано n пользователей)

Таблица 1. Объект-эталон для схемы 1

	Информация для идентификации	**Информация для аутентификации**
1	ID_1	E_1
2	ID_2	E_2
...	...	...
n	ID_n	E_n

где, $E_i=F(ID_i, K_i)$,

F – функция, для которой можно качественно описать свойство «невосстановимости» K_i по E_i и ID_i. В качестве такой функции может быть выбрана функция зашифрования (см.выше).

«Невосстановимость» K_i описывается некоторой пороговой трудоемкостью T_0 решения задачи восстановления аутентифицирующей информации по E_i и ID_i, ниже которой не должна опускаться ни одна оценка трудоемкости нахождения K_i для всех известных алгоритмов решения данной задачи.

Кроме того, для пары K_i и K_j возможно совпадение соответствующих значений E. В связи с этим вводится вероятность ложной аутентификации пользователя $P_{ла}$, которая вычисляется как условная вероятность события «совпадение E_i и Ej при условии нетождественности K_i и Kj». Эта вероятность не должна превосходить некоторой предельной величины P_0.

На практике задают $T_0 = 10^{20} – 10^{30}$, $P_0 = 10^{-7} – 10^{-9}$.

Алгоритм идентификации и аутентификации:

1. Пользователь предъявляет свой идентификатор (имя) ID.

2. Если ID не совпадает ни с одним IDi, зарегистрированным в КС, то идентификация отвергается – пользователь не допущен к работе (в смысле того, что он не может инициировать ни один субъект), иначе (существует IDi=ID) устанавливается факт «пользователь, назвавшийся пользователем i, прошел идентификацию».

3. У пользователя субъектом аутентификации запрашивается аутентификатор K.

4. Субъектом аутентификации вычисляется Y=F(IDi, K).

5. Субъектом аутентификации производится сравнение E_i и Y. При совпадении фиксируется событие «пользователь успешно аутентифицирован в КС», информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных (см. глава 1), в противном случае аутентификация отвергается – пользователь не допущен к работе (в смысле – см. выше).

Данная схема может быть модифицирована.

Схема 2

В КС выделяется объект следующей структуры (положим, как и в схеме 1, что в системе зарегистрировано n пользователей).

Таблица 2. Объект-эталон для схемы 2

	Информация для идентификации	**Информация для аутентификации**
1	ID_1, S_1	E_1
2	ID_2, S_2	E_2
...	...	...
n	ID_n, S_n	E_n

где, $E_i=F(S_i, K_i)$,

S_i – случайный вектор, заданный при создании пользователя (т.е. при создании строки, необходимой для идентификации и аутентификации пользователя),

F – функция, для которой можно качественно описать свойство «невосстановимости» K_i по E_i и S_i.

«Невосстановимость» K_i понимается как и для схемы 1.

Алгоритм идентификации и аутентификации:

1. Пользователь предъявляет свой идентификатор (имя) ID.

2. Если ID не совпадает ни с одним IDi, зарегистрированным в КС, то идентификация отвергается – пользователь не допущен к работе (в смысле не может инициировать ни один субъект), иначе (существует IDi=ID) устанавливается факт «пользователь, назвавшийся пользователем i, прошел идентификацию».

3. По IDi выделяется S_i.

3. У пользователя субъектом аутентификации запрашивается аутентификатор K.

4. Субъектом аутентификации вычисляется $Y=F(S_i, K)$.

5. Субъектом аутентификации производится сравнение E_i и Y. При совпадении фиксируется событие «пользователь успешно аутентифицирован в КС», информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных (см. глава 1), в противном случае аутентификация отвергается – пользователь не допущен к работе (в смысле – см. выше).

Вторая схема аутентификации применяется в ОС Unix. В качестве идентификатора применяется имя пользователя (запрошенное по Login), в качестве K_i – пароль пользователя (запрошен по Password), функция F представляет собой алгоритм шифрования DES. Эталоны для идентификации и аутентификации содержатся в файле Etc/passwd.

Докажем важное утверждение о свойстве объекта-эталона.

Утверждение 1 (о подмене эталона).

Если пользователь имеет доступ на запись к объекту хранения эталона, то он может быть идентифицирован и аутентифицирован (в рамках рассмотренных схем) как любой пользователь.

Доказательство

Пусть имеется пользователь i. Покажем, что он может выдать себя за любого пользователя j. Возможность записи в объект, содержащий

эталоны, означает возможность замены любой записи на произвольную. Пользователь i меняет j-ю запись на свои параметры IDi и E_i (или дополнительно и S_i). При следующей процедуре идентификации он будет опознан как пользователь j (по построению и описанию схем). Утверждение доказано.

Смысл данного утверждения состоит в том, что доступ на запись к объекту хранения эталонов должны иметь только субъекты специально выделенного пользователя КС, отвечающего за управление безопасностью.

Заметим, что необходимым требованием устойчивости схем аутентификации к восстановлению K_i аналитическими методами является требование случайного равновероятного выбора K_i из всего множества возможных. Это требование автоматически снижает ценность систем парольной аутентификации, поскольку в них выбор аутентифицирующей информации происходит из небольшого множества осмысленных слов, мощность которого определяется энтропией соответствующего языка.

4.3. Использование внешних субъектов при реализации и гарантировании политики безопасности

Выше упоминалось, что при проектировании механизмов генерации ИПС необходимо предусмотреть процедуру проектирования субъекта, который реализует алгоритм вычисления хеш-функций. Этот субъект может быть интегрирован в состав МБС, но может существовать и отдельно. В связи с этим можно утверждать, что защитная подсистема КС, включающая механизмы генерации ИПС и субъекты МБО, нуждается в реализации ряда общих функций, связанных с логическим преобразованием содержания объектов (функции логической защиты). К таким функциям относятся алгоритмы контроля целостности объектов КС, алгоритмы аутентификации или авторизации субъектов или пользователей, которые управляют субъектами, алгоритмы поддержания конфиденциальности содержания объектов (например, объекта вторичной аутентификации пользователей (см. предыдущую главу)).

Международные стандарты описывают ряд хорошо изученных функций защитного характера, в частности, алгоритмы хеширования MD2 и MD5, ГОСТ Р 34.11, алгоритмы генерации и проверки ЭЦП DSS и ГОСТ Р 34.10. Все эти алгоритмы имеют различную спецификацию вызовов (в частности, различную длину аргументов) и, естественно, не совместимы между собой.

Выше рассматривалась задача разработки универсальных методов проектирования защиты в КС (с применением метода генерации ИПС). В связи с этим необходимо заметить, что описанные методы генерации ИПС и связанные с ним алгоритмы работы с объектами не используют какого-либо конкретного алгоритма контроля целостности или поддержания конфиденциальности (или в случае использования аппаратной поддержки аутентификации, конкретного алгоритма аутентификации). С другой стороны, реализации ИПС должны сопрягаться с различными общепринятыми стандартами реализации логических функций защиты.

При организации территориально распределенных КС в различных локальных сегментах могут использоваться функционально одинако-

вые, но семантически разные функции логической защиты (вычисление функций КЦ может производиться с применением различных алгоритмов). Следовательно, субъекты различного уровня, участвующие в создании ИПС, должны сопрягаться с существующими решениями в области логической защиты. Особенно актуальна эта проблема относительно стандартизированных и сертифицированных аппаратных модулей типа FORTEZZA или Crypton.

При сопряжении с различными средствами защиты по-прежнему важным остается вопрос о свойствах защищенности КС при добавлении какого-либо субъекта к множеству существующих. Кроме задач семантического сопряжения (по перечню, типу и длине аргументов) или стандартизации функций возникает задача проверки корректности работы субъекта реализации защитных функций, понимаемая несколько шире, чем сформулированная выше корректность субъектов относительно друг друга. Необходимость более широкого взгляда на проблему межсубъектного взаимодействия следует из того, что при передаче информации от субъектов, обеспечивающих гарантии ПБ (МБС) или субъектов реализации ПБ (МБО) к субъектам, реализующим логические функции защиты, происходит в общем случае изменение ассоциированных объектов рассматриваемых субъектов. Несмотря на то, что речь идет об изменении ассоциированных объектов-данных, в случае локализации в рамках одного адресного пространства и функционально ассоциированных объектов возможно потенциальное нарушение корректности.

Цель изложения данного параграфа можно сформулировать следующим образом: в условиях априорного заданного при проектировании разделения системы безопасности на защитные субъекты (МБО и МБС) и субъекты реализации функций логической защиты необходимо в ходе проектирования решить следующие задачи: определить условия корректного взаимодействия этих субъектов, определить условия (в случае их существования), в которых поддерживается гарантированная защищенность всей КС, определить структуры данных и их свойства для организации взаимодействия защитных субъектов с субъектами реализации логической защиты.

Задача использования внешних функций логической защиты актуальна не только для защитных субъектов, но и для произвольного субъекта, входящего в КС (например, использование субъекта вычисления КЦ для фиксации целостности информации, передаваемой во внешнюю сеть). В связи с этим вопрос взаимодействия с функциями логической защиты будет изучаться без привязки к конкретным функциям субъекта (исключения будут отмечаться особо).

4.4. Понятие внешнего разделяемого сервиса безопасности. Постановка задачи

Субъекты КС, связанные с выполнением защитных функций (например, субъекты МБО и МБС, либо субъекты прикладного уровня), как было сказано выше, могут использовать некоторое общее подмножество функций логического преобразования объектов (в частности, алгоритмы контроля целостности объектов). При проектировании и реализации субъектов КС исторически сложившийся подход относительно

использования общего ресурса связан с использованием разделяемых субъектов, выполняющих общие для некоторого подмножества внешних по отношению к данному субъектов функций (например, динамически загружаемых библиотек – типа DLL для ОС Microsoft Windows). Логично распространить данный подход на функции реализации защиты от НСД. Среди общего множества функций, относящихся к защитной компоненте КС, можно выделить три класса: *функции, связанные с работой средств идентификации и авторизации пользователей, субъектов и объектов; функции, связанные с контролем неизменности объектов; функции, связанные с логическим преобразованием объектов КС и поддержанием функций конфиденциальности* (криптографические функции). Иногда отдельно выделяют функции *генерации случайных последовательностей*, необходимых, в частности, для формирования индивидуальных аутентификационных признаков пользователей (K_i).

Далее будем говорить о защитных функциях КС, объединяя в данном термине все три класса функций и уточняя по мере необходимости те особенности при их реализации, которые необходимо следуют из их свойств.

Проблему проектирования субъектов, реализующих функции логической защиты, можно рассматривать в нескольких аспектах:

• проблему их ***оптимальной реализации*** в рамках некоторого субъекта КС, к которому обращаются остальные субъекты за выполнением соответствующих функций (в данном случае речь идет о задаче оптимизации параметров «быстродействие-память» при реализации защитных функций),

• проблему ***мобильности*** субъектов, использующих защитные функции при изменении внутреннего наполнения, реализующего защитные функции субъекта (имеется в виду задача максимальной переносимости субъектов, использующих защитные функции на иные алгоритмы, например, иной алгоритм контроля целостности),

• проблему ***корректного использования*** субъекта, реализующего защитные функции, со стороны вызывающих его модулей (проблема корректного использования, как отмечалась, несколько шире, чем просто корректность субъектов, поскольку передача информации к ассоциированным объектам-данным при вызове функций логической защиты (либо при возврате управления) подразумевает влияние на ассоциированные объекты вызывающего субъекта).

Сосредоточимся на изучении третьей проблемы и при формулировании утверждений будем пояснять связь тех или иных свойств с проблемами оптимальности и мобильности.

Введем понятие разделяемой технологии применения функций логической защиты.

Разделяемая технология применения функций логической защиты – такой порядок использования средств логической защиты информации в КС, при котором:

• не требуется изменений в программном обеспечении (в содержании и составе субъектов КС) при изменении алгоритмов защиты,

• в КС однозначно выделяется модуль (программно-техническая реализация объекта-источника для порождения соответствующего субъекта) реализации защитных функций (модуль реализации защитных функций, МРЗФ).

Открытым интерфейсом (ОИ) МРЗФ назовем детальное специфицирование функций, реализованных в МРЗФ, позволяющее организовать исполнение этих функций из внешних субъектов.

«Открытость» интерфейса понимается как его полное описание для использования реализуемых в МРЗФ функций внешними субъектами. Для проектирования схем информационных потоков КС, в том числе имеющих отношение к обеспечению безопасности, такое описание имеет ключевую роль.

Рассмотрим схему функционирования МРЗФ (***рис. 1***). Пусть имеется вызывающий субъект S_i и его ассоциированные объекты: O_k (объект для передачи информации к МРЗФ) и O_f (объект возврата информации от МРЗФ после ее преобразования). Для субъекта МРЗФ выделим ассоциированные объекты: O_m для приема информации от вызывающего модуля и O_r для размещения информации перед ее транспортировкой к вызвавшему субъекту S_i.

Рис. 1. Схема взаимодействия МРЗФ с МБО И МБС

4.5. Понятие и свойства модуля реализации защитных функций

Заметим, что МРЗФ является участником межсубъектного взаимодействия в КС.

Сформулируем некоторые свойства МРЗФ.

1. Существует хотя бы один из потоков $Stream(S_x,(O_k))->O_m$ и $Stream(S_x, O_r)->O_f$. Первый поток описывает передачу информации для обработки в МРЗФ, второй – возврат результата преобразования к ассоциированным объектам вызвавшего субъекта. Обозначим через F функцию преобразования ассоциированных объектов МРЗФ, обобщая в ней

все функции логической защиты, реализуемые конкретным субъектом МРЗФ.

Субъект S_x в данном случае может быть тождественен либо S_i, либо МРЗФ и, по определению потока, устанавливает субъект, инициирующий поток.

Возможно четыре ситуации инициирования входного и выходного потока.

1. Поток передачи информации инициируется МРЗФ (процесс субъекта S_i устанавливает адрес объекта O_k и активизирует прямо или опосредованно (через субъект запуска задач или загрузки библиотек – обычно, субъект-ядро ОС) субъект МРЗФ, передавая адрес объекта O_k). Субъект МРЗФ организует поток от O_k к своим ассоциированным объектам O_m. В данном случае потоков, влияющих на ассоциированные объекты S_i, не возникает.

2. Поток передачи информации инициируется S_i (как правило, в случае ждущего режима работы МРЗФ, который фиксирует создание или изменение объекта O_m). В данном случае за счет существования потока Stream(S_i,O_k)-> O_r существует возможность нарушения корректности межсубъектного взаимодействия за счет влияния на ассоциированные объекты МРЗФ.

3. Поток возврата инициируется S_i. После выполнения логического преобразования $F(O_k)$->O_r МРЗФ передает управление вызвавшему субъекту S_i. Вызвавший субъект инициирует поток Stream(S_i,O_r)->O_f. Угрозы корректности межсубъектному взаимодействию нет.

4. Поток возврата инициируется МРЗФ. Существует угроза корректности.

Из рассмотренных четырех возможностей две ситуации безопасны с точки зрения взаимной корректности субъектов, а две опасны. Легко видеть, что опасность некорректного взаимодействия возникает в случае, когда передача информации к субъекту производится другим субъектом (ситуации 2 и 4).

С точки зрения передачи и возврата параметров наилучший случай, когда поток передачи параметров инициируется МРЗФ (на практике так обычно и бывает), а поток возврата инициируется вызывающим субъектом S_i (обычно это реализовано не так: вызывающий субъект передает адрес объекта O_f как своего ассоциированного объекта O_f (объекты O_r и O_f могут совпадать) и одновременно с вычислением результата $F(O_k)$->O_r инициируется поток к ассоциированному объекту S_i). В данном случае обеспечение корректного возврата результата может быть обеспечено двумя путями:

• свойствами самого МРЗФ (должен быть обеспечен возврат результата именно в объект O_f),

• возвратом вызвавшему субъекту адреса объекта O_r и инициированием потока Stream(S_i,O_r)->O_f уже вызвавшим субъектом, что не несет угрозу корректности.

Формализуем изложенные положения в нескольких определениях.

Определение 1. Корректным преобразованием информации в МРЗФ в период времени T называется такой порядок существования субъекта МРЗФ в КС, при котором функция преобразования ассоциированных объектов F соответствует описанию ОИ и не изменяется в течение всего времени T.

Определение 2. Корректным вызовом МРЗФ называется такой порядок реализации потока Stream (S_x, O_k)->O_m, где O_m ассоциированный объект МРЗФ, когда объекты O_k и O_m тождественны.

Определение 3. Корректным возвратом результата от МРЗФ называется такой порядок реализации потока Stream (S_x, O_r)-> O_f, при котором объекты O_r и O_f тождественны.

Определение 4. Корректной работой МРЗФ в КС в период времени Т назовем такой порядок его использования, при котором обеспечивается корректное преобразование, корректный вызов и корректный возврат результата.

Покажем, что в общем случае всегда существует субъект, относительно которого происходит некорректный возврат результата.

Утверждение 6 (о потенциальной возможности некорректного возврата результата из МРЗФ)

Относительно любого МРЗФ всегда существует субъект, относительно ассоциированных объектов которого происходит некорректный возврат результата.

Доказательство

Для доказательства утверждения необходимо описать хотя бы один субъект, имеющий свойство преобразования ассоциированного объекта O_f, относительно которого происходит возврат результата преобразования: Stream(МРЗФ, O_r)->O_f. Таким субъектом будет любой S_i, который определяет O_f либо как неассоциированный с самим собой объект, либо выполняет над объектом Of дополнительное преобразование, не равное тождественному отображению.

Из данного утверждения следует, что:

• субъект, вызывающий МРЗФ должен содержать в себе корректную реализацию получения результата,

• корректная реализация вызова должна поддерживаться в течение заданного времени Т промежутка активности КС.

Аналогичное утверждение можно сформулировать и для операции вызова МРЗФ.

Утверждение 7 (о потенциально возможном некорректном вызове МРЗФ)

Относительно любого МРЗФ всегда существует субъект, относительно ассоциированных объектов которого происходит некорректный вызов МРЗФ.

Доказательство

Аналогично предыдущему утверждению, необходимо описать субъект S_i, для которого Stream(S_x, O_k)->O_m. Таким субъектом будет любой S_i, который определяет O_m как неассоциированный с МРЗФ объект. При этом очевидно, что объекты O_k и O_m будут в общем случае нетождественны, это означает, что происходит отображение объектов, не равное тождественному отображению.

Приведенные утверждения позволяют сделать вывод о том, что проблема обеспечения корректной работы МРЗФ в КС должна решаться исходя как из свойств МРЗФ, так и вызывающих его субъектов.

Если обеспечить на этапе разработки механизмы корректного вызова и корректного возврата результата, то для поддержания указанных свойств корректности необходимо:

• обеспечить замкнутость программной среды, т.е. возможность активизации только субъектов из числа проверенных на корректность вызова и возврата,

• обеспечить неизменность объектов-источников для порождения вызывающих субъектов и самого объекта-источника МРЗФ.

Невыполнение хотя бы одного из указанных условий может привести к некорректному вызову или возврату (поскольку вызывающий субъект может быть произволен либо за счет привнесения в КС нового объекта-источника, либо за счет изменений объектов-источников уже имеющихся в КС (например, самого МРЗФ)).

Практически указанные необходимые требования означают использование МРЗФ в режиме генерации ИПС.

4.6. Проектирование модуля реализации защитных функций в среде гарантирования политики безопасности

Из указанных выше положений следует, что достаточным условием корректной работы системы с МРЗФ является замкнутость всех субъектов, включая МРЗФ в ИПС с контролем неизменности объектов-источников.

Докажем это положение.

Утверждение 8 (достаточные условия корректного использования МРЗФ)

Если в изолированной (абсолютно изолированной) КС существуют только субъекты, которые корректно вызывают МРЗФ и обеспечивают корректный возврат результата. Существует также собственно МРЗФ и действует контроль порождения субъектов с контролем неизменности объектов-источников, а также исключено существование потоков к O_m или O_r стороны любого субъекта S_i от начала Stream (S_x, O_k)->O_m до завершения Stream (S_x, O_r)->O_f, то МРЗФ вызывается корректно в течение всего времени активности КС (S_x по-прежнему тождественен либо S_i, либо МРЗФ).

Доказательство

Для доказательства утверждения необходимо показать, что:

1. МРЗФ обеспечивает корректное преобразование информации, т.е. функция F преобразования информации в МРЗФ неизменна,

2. обеспечивается корректный вызов со стороны любого субъекта,

3. обеспечивается корректный возврат результата к любому субъекту.

Докажем первое положение.

В КС действует МБС, производящего порождение субъектов с контролем неизменности объектов, это означает, что для различных t1 и t2 операции

Create(S_i,Объект МРЗФ[t1-1])->МРЗФ[t1] и

Create(S_i,Объект МРЗФ[t2-1])->МРЗФ[t2] порождает тождественные субъекты МРЗФ[t1] и МРЗФ[t2].

Поскольку все субъекты КС по условию изолированности как минимум попарно корректны и корректны относительно МРЗФ, то отсутствуют потоки к ассоциированным объектам МРЗФ, которые могли бы изменить функцию F.

Следовательно, корректность функционирования МРЗФ доказана.

Докажем второе положение.

Поскольку любой субъект обеспечивает корректный вызов и возможно порождение только тождественных субъектов в различные моменты времени, то поток Stream(S_i,O_k)->O_m есть тождественное отображение, если не существует потоков к объектам O_k или O_m от других субъектов. Отсутствие потоков к O_k гарантируется корректностью субъектов. Поток к O_m может быть инициирован как вызов МРЗФ со стороны иного субъекта, однако это запрещено по условию теоремы.

Аналогично доказывается третье (корректный возврат результата).

Утверждение доказано.

Условие невозможности изменения ассоциированных объектов МРЗФ, доступных для передачи параметров (т.е. запрет вызова МРЗФ до завершения операций с вызвавшим субъектом), может реализоваться двумя путями.

Первый путь заключается в порождении при вызове каждым субъектом нового субъекта МРЗФ (реализован в механизме загрузки динамических библиотек – DLL).

Второй путь заключается в блокировании потока от любого субъекта до завершения операции с МРЗФ, т.е. до завершения потока возврата результата.

Относительно некоторого множества субъектов, использующих МРЗФ, можно говорить о полноте функций МРЗФ. При этом удобнее оперировать с формально описанными функциями ОИ (которые однозначно соответствуют функциям МРЗФ), поэтому далее будем говорить о полноте функций ОИ. Полнота функций ОИ может быть функциональной и параметрической.

Функциональная полнота ОИ – свойство, заключающееся в реализации всех функций защиты, инициируемых фиксированным набором субъектов КС.

Из данного определения следует, что функциональная полнота понимается относительно заданного множества программ, использующих функции ОИ.

Параметрическая полнота ОИ – свойство, заключающееся в возможности инициирования всех функций ОИ со стороны фиксированного множества субъектов с некоторым набором параметров, не приводящих к отказу в выполнении запрошенной функции.

Как о функциональной, так и о параметрической полноте можно говорить относительно конечного множества субъектов, следовательно, понятие полноты ОИ также, как и корректное использование МРЗФ, имеет смысл только в ИПС.

4.7. Передача параметров при составном потоке

Остановимся теперь на ситуации составного потока передачи и возврата управления. Рассмотрим ситуацию, когда существует объект Op такой, что:

1. Stream(S_x,O_k)->Op и Stream(S_x, Op)->O_m
2. Op не ассоциирован ни с S_i, ни с МРЗФ,

когда в потоке участвует некоторый неассоциированный с S_i и МРЗФ объект.

При этом данный субъект может быть ассоциирован с каким-либо другим субъектом. Практически данная ситуация повсеместно возникает в реальной КС в случае передачи параметров через внешнюю память (файл), либо при взаимодействии вызывающего субъекта и МРЗФ через сетевую транспортную среду (необходимо уточнить, что в этом случае ситуация усложняется тем, что существует не один объект Op, а, как минимум, два территориально несовпадающих объекта Op1 и Op2, между которыми существуют потоки Stream(S_x, O_k)->Op1, Stream(Sl, Op1)->Op2 и Stream(Sy,Op2)->O_m (и аналогично при возврате результата). Для простоты рассмотрения будем говорить только об одном объекте, включенном в поток, поскольку можно говорить, об одном объекте Op, но измененном субъектом Sl.

В приводимой таблице рассмотрим свойства подсистемы КС, состоящей из объектов O_k, Op, O_m и субъектов S_i, Sl и МРЗФ.

Будем учитывать свойство по инициированию потока Stream (S_x,O_k)->Op и Stream(S_x, Op)->O_m, а также функцию F(Op)->Op преобразования объекта Op, локализованную в Sl, и существование потоков Stream(Sl,Op)->Oy, где Oy – некоторый объект. Рассматриваемые свойства взаимодействия отражают все возможные ситуации, возникающие при участии субъекта Sl в потоке:

1. Свойства по преобразованию проходящей информации.

2. Свойства по отображению проходящей информации в другой объект.

Действительно, с точки зрения опосредованного НСД возможны либо изменения состояния объекта, либо отображение его содержания в другой объект. Учитывая то, что ранее рассматривалось тождественное отображение объектов при передаче параметров и возврате результата, будем рассматривать функцию F либо как тождественное отображение, либо как нетождественное. В таблицу введен также существенный параметр – каким субъектом порожден поток к промежуточному объекту – вызывающим субъектов (S_i), либо МРЗФ (обозначения S_x=S_i означает, что субъектом S_x является вызывающий субъект).

Таблица 4. (Свойства составного потока при использовании МРЗФ)

	S_x=S_i	S_x=МРЗФ	Отображение F	Существует поток Stream(Sl,Op)->Oy	Примечание
0	Нет	Нет	~E	Нет	Не рассматривается
1	Нет	Нет	~E	Да	Не рассматривается
2	Нет	Нет	E	Нет	Не рассматривается
3	Нет	Нет	E	Да	Не рассматривается
4	Нет	Да	~E	Нет	Искажение промежуточного объекта

5	Нет	Да	~E	Да	Искажение промежуточного объекта и поток к постороннему объекту
6	Нет	Да	E	Нет	Корректная передача, инициированная МРЗФ
7	Нет	Да	E	Да	Корректная передача, но с организацией постороннего потока
8	Да	Нет	~E	Нет	Аналогично 4
9	Да	Нет	~E	Да	Аналогично 5
10	Да	Нет	E	Нет	Аналогично 6
11	Да	Нет	E	Да	Аналогично 7
12	Да	Да	~E	Нет	Некорректное использование промежуточного объекта
13	Да	Да	~E	Да	Некорректное использование промежуточного объекта
14	Да	Да	E	Нет	Некорректное использование промежуточного объекта
15	Да	Да	E	Да	Некорректное использование промежуточного объекта

Случаи 0-3 не рассматриваются, поскольку участвующие в сложном потоке субъекты не совпадают, ни с вызывающим субъектом, ни с МРЗФ. Априорно можно также считать некорректными случаи 12-15, поскольку одновременное наличие потока к промежуточному объекту заведомо порождает некорректный вызов или возврат результата.

Итак, содержательному анализу подвергнем ситуации 4-11.

Ситуация 4 описывает поток с искажением промежуточного объекта.

Ситуация 5 кроме искажения промежуточного объекта содержит еще и поток к постороннему объекту.

Ситуация 6 описывает полностью корректную передачу к промежуточному объекту.

Ситуация 7 описывает передачу параметров без изменения, но с организацией потока к другому объекту.

Ситуации 8 и 9 описывают функционально аналогичные ситуациям 4 и 5, но относительно другого инициирующего поток субъекта.

Аналогично рассматриваются и ситуации 10 и 11.

Рассмотрев различные ситуации, связанные с передачей и возвратом параметров через промежуточный объект, можно сформулировать требования к субъекту, управляющему потоками информации через данный объект.

1. В субъекте Sl должно быть реализовано только тождественное отображение объекта Op.

2. Субъект Sl не должен инициировать потоки, отображающие промежуточный объект на другие объекты.

Данные свойства должны быть предметом содержательного анализа при проектировании защиты в КС, где существуют субъекты – участники составных потоков.

4.8. Методика проверки попарной корректности субъектов при проектировании механизмов обеспечения безопасности с учетом передачи параметров

Рассмотренная выше модель передачи параметров между субъектами позволяет уточнить понятие попарной корректности, изложенное выше. Общие подходы к изучению субъектного наполнения КС хорошо изложены в литературе[1].

Под ***некорректным программным субъектом*** понимается процедура (в терминах языка высокого уровня), выполняющая хотя бы одну из операций:

- выделение в теле процедуры динамической памяти и неосвобождение ее при выходе из процедуры;
- обращение по чтению/записи к динамически выделенной памяти после ее освобождения;
- обращение по чтению/записи за границы как динамически, так и статически (стек, глобальные переменные) выделенной памяти;
- запись в сегмент кода программного модуля;
- запись в область переменных, задающих порядок возврата управления при выходе из процедуры.

Под ***контекстуально некорректным программным субъектом*** понимается некорректный в некотором (хотя бы одном) контексте вызова программный субъект.

Под ***контекстом вызова*** программного субъекта понимается совокупность объектов данных известной размерности (или их адресов), передающихся ему в качестве параметров вызова.

Контекстуальная некорректность проявляется при вызове одних программных субъектов другими и заключается в некорректных операциях с объектами, адреса которых передаются в качестве параметров в процедуру. Выявить такого рода некорректность можно только в контексте вызова одной процедуры другой, поскольку передача в качестве параметра указателя на некоторый объект (например, типа массив) не дает информации, достаточной для осуществления контроля за правильностью операций над данным объектом.

1. Ховард М., Лебланк Д. Защищенный код/ Пер. с англ. – М.: Изд-во «Русская редакция», 2003 – 704 с., ил.

Таким образом, данная классификация отражает предмет **статического анализа** программного кода по исходным текстам на языке высокого уровня.

Под ***программным субъектом*** понимается выделенная процедура: фрагмент кода на языке высокого уровня, ограниченный точкой входа (и, возможно, передачи параметров) и точкой (точками) возврата управления (выходом из процедуры).

Попарная корректность процедур P1 и P2 понимается как отсутсвие следующих операций

• по записи в области памяти, содержащие код как процедуры P1, так и процедуры P2 из процедур P1 и P2;

• по записи в область данных процедуры P1 из процедуры P2 и наоборот (кроме передачи параметров при взаимном вызове);

• по записи в область передачи параметров при вызове процедуры P2 из процедуры P1 (или наоборот), превышающую размерность объекта для передачи параметров;

• по записи в область переменных, задающих порядок возврата управления и передачи параметров (стека).

Динамические методы анализа направлены на выявление некорректных ситуаций в динамике выполнения программного кода, реализующего целевую функцию ПО. К динамическим методам анализа относятся:

• исполнение ПО (фрагментов ПО) под управлением отладчика. При этом имеется возможность использования условных точек останова (breakpoints) для различных классов событий (обращение к участкам памяти по чтению/записи, обращение к портам ввода/вывода, возникновение исключений (exceptions) в работе ПО), а также пошаговое исполнение наиболее критичных участков программного кода.

• написание тестовых макросов, которые вносятся в исходные тексты ПО перед его компиляцией в исполняемые модули, выполняющих динамическую проверку некоторых условий на этапе исполнения программного кода (например, на входе процедуры тестовый макрос сохраняет сегмент стека (stack frame) на некоторую глубину и проверяет его неизменность на выходе из процедуры, или тестовый макрос на входе процедуры вычисляет контрольную сумму программного кода и проверяет его неизменность на выходе процедуры и т.д.) и сигнализирующих в случае их невыполнения.

• написание и исполнение тестовых программ, создающих стрессовые или нештатные условия эксплуатации ПО и дальнейший анализ реакции ПО на данные ситуации.

Статическими являются методы анализа по исходным текстам на языке высокого уровня.

Анализ одиночной процедуры

1. Выявление прямых операционных конструкций:
   1.1. Вызовов других процедур.
   1.2. Циклов for, do while, do until.
2. Выявление косвенных операционных конструкций
   2.1. Изменение объектов стека.
   2.2. Косвенные циклы через if-else.
   2.3. Процедур типа longjump.

3. Анализ выявленных конструкций:

3.1. Происходит ли перемещение данных внутри процедуры, каковы параметры операций, возможно ли повреждение (запись) в области собственного кода или кода другой процедуры.

3.2. Происходит ли перемещение данных непосредственно во вне процедуры (в глобальные переменные), и возможно ли повреждение кода других процедур.

3.3. Происходит ли перемещение данных в область стека данной процедуры или других процедур.

3.4. Происходит ли сквозная передача параметров для вызовов из данной процедуры, существует ли возможность повреждения кода и данных данной процедуры при возврате управления.

Пример сквозного вызова:

```
int fun1 (char *buf, int len)
 {
 ...
 fun2(buf,15);
 }
```

Пример повреждения при возврате:

```
int fun1 (...)
 {
 char buf[32];
 ...
 fun2(buf);
 }

void fun2(char *p)
 {
 ...
 for(i=0;i<64;i++) p[i]=i;
 }
```

Целесообразно выделить процедуры по прямому перемещению объектов типа memcpy, strcpy и т.д.

Для каждой процедуры составляется формуляр, в котором указываются все вызываемые процедуры с указанием результатов анализа по пп.3.1 – 3.4, указывается интегральный вывод по свойствам операционных конструкций (типа «перемещение данных в области памяти других процедур не найдено (или найдено)»; при нахождении указывается конкретная локализация в процедуре с указанием параметров, при которых реализуется то или иное негативное свойство.

Должны быть проанализированы все процедуры, принадлежащие к полному множеству.

Множество процедур ПО называется полным, если во всех принадлежащих этому множеству процедурах все вызываемые процедуры также принадлежат данному множеству.

Далее будет рассмотрена конкретизированная в виде алгоритма методика анализа ПО на предмет наличия (отсутствия) как некорректных, так и контекстуально некорректных программных субъектов. Под ошибкой типа buffer overflow понимается потенциально возможная ситу-

ация, связанная с записью в область памяти за объявленной границей объекта.

Описание алгоритма.

1. Фиксируется список {F} всех процедур комплекса ПО. Для каждой процедуры описываются ее параметры (с указанием размерности) и возвращаемое значение. В том случае, когда размерность объекта данных (либо непосредственно выступающего в качестве параметра, либо опосредованно через указатель на него) не может быть определена без контекста вызова данной процедуры, указывается «размерность не определена» (unknown). Такие функции заносятся также в список {Fc} процедур, являющихся объектом анализа на предмет контекстуальной корректности. Также создается список {P} глобальных переменных. Для каждой переменной указывается ее размерность.

2. Для любой глобальной переменной P из списка {P} производится ее поиск во всех файлах с исходными текстами на языке C:

• переменную P слева от оператора присваивания (включая вхождения в операционные конструкции типа for(), while(),..); при этом размерность присваиваемого значения не должна превышать размерность переменной P,

• переменную P в качестве первого параметра в вызове функции strcpy(char *dest, char *src); при этом размерность присваиваемого значения не должна превышать размерность переменной P,

• переменную P в качестве второго параметра в вызове функции bcopy(char *src, char *dest, size_t n); при этом значение n не должно превышать размерности переменной P,

• переменную P в качестве второго параметра в вызове функции bzero(char *s, size_t n); при этом значение n не должно превышать размерности переменной P,

• переменную P в качестве первого параметра в вызове функции memcpy(void *dest, void *src, size_t n); при этом значение n не должно превышать размерности переменной P,

• переменную P в качестве первого параметра в вызове функции memset(void *s, int c, size_t n); при этом размерность присваиваемого значения не должна превышать размерность переменной P,

• переменную P в качестве параметра в вызове функций gets(char *s).

Во всех случаях выявляется потенциальная возможность ошибки типа «buffer overflow»

• конструкцию типа P=getenv(«SOME_ENVIRONMENT»)

• для переменной P типа массив операция индексирования не должна приводить к выходу за границу массива.

3. Для любой процедуры F() из списка {F}:

• для всех локальных переменных провести анализ по предыдущему пункту.

В случае, если среди параметров присутствуют указатели на некоторые объекты данных, размерность которых не может быть установлена без контекста вызова данной процедуры, то она является также объектом анализа на предмет контекстуальной корректности (см. далее) и заносится в соответствующий список.

• для каждой функции динамического распределения памяти (malloc(), ...) производится поиск соответствующей ей функции освобождения памяти (free(), ...).

После освобождения памяти не должно быть операционных конструкций, ссылающихся на нее.

• не должно быть операций записи по адресам любых меток и имен объектов (процедур и т.д.) в сегменте кода;

• не должно быть операций записи в область переменных, задающих порядок возврата управления при выходе из процедуры;

• необходимо выявить вызовы всех других процедур из данной. Для каждой из вызываемых процедур производим поиск в списке {Fc}. Для каждой найденной в этом списке определяем из контекста вызова размерность параметров и рекурсивно проводим анализ вызываемой процедуры по данному алгоритму.

4.9. Понятие защищенного хранилища

Для корректной реализации защиты КС необходимо обеспечить ***надежную систему хранения персональной информации***, включая в первую очередь информацию для аутентификации, секретные ключи пользователя (ключи подписи), ключ контроля целостности, либо образцовые хеш-значения, личные ключи, служащие для формирования открытых ключей, сертификатов, подписания электронных документов и формирования маркеров подлинности. Два последних мы подробнее рассмотрим ниже.

Введем следующие понятия.

Защищенное хранилище пользователя (ЗХП) – это аппаратно-программный модуль (самостоятельный или совмещенный с другим устройством, например, смартфоном), предназначенный для хранения информационной собственности пользователя (КСП) и/или прав информационной собственности пользователя (ПКСП) и обладающий заданным набором свойств.

Информационная собственность пользователя – объект КС специального вида, ассоциированный с физическим лицом (пользователем) и являющийся его собственностью.

Права информационной собственности пользователя – совокупность данных, ассоциированных с физическим лицом (пользователем) и однозначно устанавливающих права пользователя на объекты КС, представленные в виде цифровых информационных массивов.

Защищенный обмен между ЗХП – процесс обмена КСП и/или ПКСП, содержащимися в хранилищах.

Информационной собственностью пользователя могут являться следующие специальные объекты КС: секретный (закрытый) ключ пользователя, электронная банкнота, маркер владения, маркер доступа.

Маркер владения – объект КС, содержащий информацию о предоставленном владельцу ЗХП праве владения некоторыми информационными массивами. Данный объект определяет ПКСП.

Право владения может включать возможность пользования программными продуктами, видео и аудиоинформацией, а также другими информационными ресурсами на материальных носителях, определенными в рамках договорных отношений с владельцем этих массивов. Передача права владения в ЗХП пользователя будет означать принятие условий договора.

Публичное представление пользователя – информация о пользователе, публикуемая с его разрешения в составе открытых информационных массивов. Публичное представление пользователя содержит как минимум его сертификат, однозначно связанный с закрытым ключом пользователя. Публичное представлением может содержать другую информацию, в обязательном порядке заверенную ЭЦП.

Доверенный оператор публичных представлений (ДОПП) – уполномоченное (аккредитованное) предприятие, содержащее Удостоверяющий центр по работе с сертификатами пользователей и базы данных с публичным представлением.

ЗХП должен обладать следующими группами свойств:

- гарантии хранения информации;
- гарантии невозможности доступа к информации инженерными методами;
- гарантии стирания при попытках несанкционированного доступа;
- гарантии надежности (включая процедуры тестирования);
- свойства доступа к областям памяти;
- гарантии защищенного взаимодействия между ЗХП;
- свойства операций с сущностями пользователя;
- гарантии архитектуры, проектирования и производства.

Принципиальным для ЗХП является наличие защищенной памяти, которая стирается при попытках проникновения инженерными методами, а также наличие собственного вычислителя, предназначенного для выработки и проверки ЭЦП под сущностями пользователя, а также датчика случайных чисел. Эти свойства будут обязательными для всех классов.

Предлагается объекты, хранимые в ЗХП, идентифицировать по символьному имени, типу и свойствам данного объекта, включая его содержание. Например, типом объекта может являться то, что он является электронной банкнотой, маркером владения или маркером прав, а свойствами – возможность быть *делегируемым* (т. е. объект может быть передан в другой ЗХП, а исходный объект (сущность) может остаться в ЗХП-источнике), либо *отчуждаемым* (объект при передаче в другой ЗХП уничтожается в источнике). Так, электронная банкнота всегда будет отчуждаемой сущностью.

Защищенный обмен между ЗХП предполагается в основном трехсторонним с друхуровневой защитой.

Смысл трехстороннего обмена состоит в участии некоторого пассивного посредника (устройство типа кассовой машины), который обеспечивает гарантии взаимодействия и учет операций с ЗХП для уполномоченных органов. Например, при передаче маркера владения происходит сначала его перемещение в устройство пассивного посредника (УПП), а затем перемещение к новому владельцу. При этом в УПП ведется журнал, в котором отражаются данные по операциям передачи свойств (имена владельцев обмена, дата, время и т. д.).

Основное преимущество трехстороннего обмена состоит в том, что для обмена ЗХП – ЗХП невозможно точно гарантировать факт переноса сущности в другой ЗХП с одновременным стиранием в ЗХП-источнике. Это связано с тем, что стирание в ЗХП-источнике должно происходить только после записи и проверке перемещенной сущности в ЗХП-прием-

нике. Для этого должен быть создан канал обмена, по которому передается квитанция «запись в приемник произведена». Однако имеется принципиальная возможность блокирования обратного канала квитирования и, соответственно, нестирание сущности в ЗХП-источнике. Наличие УПП позволяет сначала перенести сущность в него, а затем в ЗХП-приемник. Другим преимуществом является доступ уполномоченных органов к движению информационной собственности, например, оплата электронными банкнотами.

Двухуровневая защита (двухуровневый протокол) подразумевает наличие двух рубежей защиты:

• создание защищенного канала взаимодействия с участием сертификатов владельцев ЗХП и их секретных ключей и обменом сеансовым ключом (ключами), на котором осуществляется шифрование и защита целостности передаваемых элементов протокола;

• обмен в рамках защищенного канала сущностями, каждая из которых защищена ЭЦП.

При этом трехсторонний обмен не исключает двухсторонний. Возможно, в ряде ситуаций будет возможен обмен ЗХП – ЗХП напрямую.

Делегируемый обмен в ряде случаев должен подразумевать изменение некоторых счетчиков. Однако это связано с тем, что первично полученный объект будет изменяться, и ЭЦП под ним будет в этом случае неверна. В связи с этим предлагается концепция неизменного хранения делегированных прав и аудит передачи прав.

Сущность этой концепции состоит в том, что объекты в свойствах имеют пометку о том, делегируемые они или отчуждаемые, а движение объекта отмечается во внутреннем журнале ЗХП, в соответствии с которым происходят ограничение делегирования при исчерпании возможных количеств делегирования либо стирание объекта в случае перемещения отчуждаемого объекта.

Таким образом, в ЗХП должен быть реализован журнал аудита передачи прав, защищенный от модификаций и доступный для извлечения отдельных записей с ЭЦП владельца ЗХП.

Для ЗХП возможны следующие субъекты, производящие доступ к данным внутренней памяти: программы ядра ЗХП, программы пользователя (внешние модули, если ЗХП интегрирован в некое устройство), внешние программы, инициирующие информационный обмен. Программы ядра ЗХП должны иметь полный доступ к памяти ЗХП.

Для внешних и пользовательских модулей возможны такие варианты:

Внешние		**Пользователь**		**Свойства**
R	W	R	W	
0	0	0	0	Нет доступа
0	0	0	1	Неэкспорируемые сущности, создаваемые пользователем (секретный ключ)
0	0	1	0	Неэкспортируемые сущности, читаемые пользователем
0	0	1	1	Неэкспортируемые сущности с полным доступом пользователя

0	1	0	0	Импортируемые сущности, недоступные пользователю
0	1	0	1	Импортируемые сущности, модифицируемые пользователем
0	1	1	0	Импортируемые сущности, читаемые пользователем
0	1	1	1	Импортируемые сущности с полным доступом
1	0	0	0	Экспортируемые сущности без доступа пользователя
1	0	0	1	Экспортируемые сущности, изменяемые пользователем
1	0	1	0	Экспортируемые сущности, только читаемые пользователем
1	0	1	1	Экспортируемые сущности, созданные пользователем
1	1	0	0	Экспортируемые и импортируемые сущности, недоступные пользователю
1	1	0	1	Экспортируемые и импортируемые сущности, записываемые пользователем без возможности чтения
1	1	1	0	Экспортируемые и импортируемые сущности, только читаемые пользователем
1	1	1	1	Экспортируемые и импортируемые сущности, с полным доступом пользователя

Указанные варианты исчерпывают все возможное множество доступов и могут быть основой для построения политики безопасности ЗХП.

Например, делегируемый маркер владения, полученный от другого ЗХП, должен иметь атрибуты (1,1,1,0), что означает возможность получения (импорта, экспортирования (переноса в другой ЗХП), а также доступ пользователю на чтение (для информирования о наличии прав). Собственный маркер владения, созданный пользователем, будет иметь атрибуты (1,0,1,1), что означает возможности передачи в другой ЗХП и модификации самим пользователем, но невозможность изменения внешней программой.

Экспортируемый объект также может иметь шаблон доступа, входящий в состав подписанного ЭЦП данного объекта. Он описывает свойства данного объекта при его записи в другой ЗХП. Например, сформированный неким пользователем собственный маркер владения при передаче конечным устройствам ЗХП будет иметь шаблон (0,0,1,0), что позволит пользователю только прочитать права, имеющиеся в маркере, но не допустить их переноса в другие ЗХП (тиражирования).

Протокол защищенного обмена между ЗХП предполагает наличие сертификата Удостоверяющего центра ДОПП у обоих пользователей, при этом сертификат ДОПП имеет атрибут (0,1,1,0), т. е. это только импортируемый объект, доступный для чтения пользователю.

ЗХП1		ЗХП2
→	Обмен сертификатами	←
	Проверка сертификатов	
	Выработка общего ключа	
→	Обмен частями сеансового ключа	←
	Получение сеансового ключа	
	Тест сеансового ключа	
	Защищенная сессия	
→	Передача объекта	
		Прием объекта в буфер обмена
		Проверка ЭЦП и атрибутов
		Запись во временный массив
	Передача квитанции о записи, либо ошибки «запись невозможна»	←
Фиксация факта «объект получен», либо ошибка		
→	Передача квитанции о записи/уничтожении	Квитанция об операции в источнике
		Перенос в постоянное хранилище
	Квитанция о записи в приемник	←
Запись в журнал транзакций		
	Завершение защищенной сессии	

Части сеансового ключа получаются с датчика случайных числе (ДСЧ) каждого ЗХП, передаются на общем ключе и суммированием образуют сеансовый ключ. Тестирование сеансового ключа предусматривает зашифрование на нем случайного числа, выполнение предопределенной арифметической операции в приемнике, возврат в зашифрованном виде и проверку. Защищенная сессия протекает на сеансовом ключе. При этом пользовательский объект, полученный в рамках сессии по зашифрованному каналу, проверяется на логическую структуру с проверкой его ЭЦП и атрибутов. Аналогичным образом возможно организовать загрузку программного обеспечения в ЗХП.

4.10. Типовые требования к защищенным хранилищам

Интегрирующий модуль (ИМ) представляет собой отдельное устройство (вычислитель или коммуникатор), в который установлено ЗХП. Под доступом пользователя к ИМ и ЗХП понимается возможность пользователя непосредственно или удаленно инициировать работу аппаратных или программных средств, входящих в состав ИМ (ЗХП), и опериро-

вать с данными, хранящимися в ЗХП.

Под программной средой (ПС) ИМ понимается совокупность программного обеспечения (ПО) и других объектов ИМ.

Под *неудачной попыткой доступа* (НПД) пользователя к ИМ (ЗХП) понимается попытка доступа пользователя к ИМ с отрицательным результатом аутентификации пользователя.

Пользователь – это лицо или процесс, действующий от имени этого лица, действия которого регламентируются правилами разграничения доступа к информации (объектам) ЗХП.

Идентификатор пользователя – уникальный признак, присвоенный пользователю при его регистрации.

Идентификация пользователя – присвоение пользователю идентификатора и (или) сравнение предъявляемого им идентификатора с перечнем присвоенных пользователям идентификаторов.

Аутентификация пользователя – проверка принадлежности пользователю предъявляемого им идентификатора пользователя; подтверждение подлинности.

Аудит состояний ЗХП – комплекс организационно-технических мероприятий, позволяющий контролировать журнал состояний ЗХП, а также убеждаться в правильности функционирования ЗХП

Журнал состояний ЗХП – совокупность записей, содержащих информацию о событиях, произошедших во время работы ЗХП.

Сеанс работы пользователя на ИМ – промежуток времени между двумя последовательными событиями, приводящими к перезагрузке операционной системы ИМ, или между последовательно следующими друг за другом событием, приводящими к загрузке (перезагрузке) операционной системы, и выключением ИМ.

Свойства	**АКХ3**	**АКХ2**	**АКХ1**
Идентификация и аутентификация владельца	+	+	+
Идентификация внешних субъектов	+	+	=
Разграничение доступа к объектам	+	+	+
Контроль целостности	+	+	=
Регистрация событий	+	=	=
Защита от проникновения	+	+	=
Гарантии стирания	+	=	+
Гарантии хранения	+	=	=
Надежность функционирования ЗХП	+	+	+
Тестирование	+	+	=
Используемое СКЗИ	КС1	КС2	КС3
Защищенное хранение данных	-	+	=
Защищенный протокол взаимодействия	+	+	=
Операции над хранимыми объектами	-	+	=
Гарантии проектирования	+	+	+
Гарантии тиражирования и сопровождения	-	+	=
Документация	-	+	+
Специальные требования	-	+	=

Требования к классу АКХ3

Идентификация и аутентификация владельца. В ЗХП должна быть предусмотрена возможность идентификации и аутентификации владельца (пользователя). Вероятность ложной аутентификации на любую попытку доступа за время эксплуатации ЗХП не должна превосходить 10^{-7}.

Идентификация внешних субъектов. ЗХП должен идентифицировать обращения со стороны внешних субъектов, в том числе и со стороны других ЗХП, и отличать их от обращений пользователя и внутренних программ.

Разграничение доступа к объектам. ЗХП должен обеспечивать разграничение доступа к своим объектам на основе информации, полученной от процедур идентификации и аутентификации. ЗХП должен иметь перечень объектов, для которых в Специальном техническом задании (СТЗ) устанавливаются правила доступа. Правила доступа могут быть заданы также для групп объектов. Различение объектов происходит по их типам и идентификаторам (именам).

Контроль целостности. ЗХП должен обеспечивать контроль целостности внутренних данных, перечень которых определен в СТЗ. Алгоритм проверки целостности должен обеспечивать величину вероятности ошибки, не превосходящую 10^{-7} для каждого объекта контроля. При нарушении целостности должны запускаться механизмы стирания объектов, перечень которых также указывается в СТЗ.

Регистрация событий. Во внутренней памяти ЗХП должна быть предусмотрена область для журнала состояний – структуры для сохранения параметров операций над данными ЗХП. Доступ к журналу состояний должен быть разрешен только для внутренних программ ЗХП.

Каждая запись журнала состояний должна содержать: время и дату события; код события. Количество событий, превышающее емкость журнала состояний, не должно приводить к его уничтожению. Экспорт журнала состояний возможен только в виде, подписанном ЭЦП на секретном ключе владельца ЗХП.

Защита от проникновения. ЗХП должен обеспечивать гарантированное стирание заданных в СТЗ объектов при попытках проникновения внутрь корпуса. В обязательном порядке производится стирание секретного ключа владельца.

Гарантии стирания. При стирании объектов ЗХП (при попытках проникновения либо обусловленном правилами работы с объектами ЗХП) должна гарантироваться невозможность восстановления стертых данных. Гарантии стирания обеспечиваются многократной перезаписью заданных областей внутренней памяти ЗХП. Кратность стирания определяется в СТЗ.

Гарантии хранения. ЗХП должен обеспечивать хранение внутренних данных сроком до 10 лет.

Надежность функционирования ЗХП. Вероятность отклонения от заданных алгоритмов работы ЗХП, вызванного неисправностями и сбоями в работе его аппаратных и программных средств, не должна превосходить значения 10^{-4} в течение периода времени между двумя последовательными тестированиями.

Тестирование. В ЗХП должны быть реализованы механизмы проверки правильности работы его программного и аппаратного компонентов. Тестирование ЗХП должно выполняться в начале каждого сеанса работы, связанного с использованием ИМ, подачей питания на ЗХП, а также с заданной периодичностью. В случае невыполнения или ошибки хотя бы в одном тесте должно быть произведено стирание информации в ЗХП с выполнением указанных выше требований (гарантии стирания). В обязательном порядке тестируются СКЗИ, процедуры идентификации и аутентификации, контроля целостности и операций с объектами. Полный перечень тестируемых функций определяется в СТЗ.

Используемое СКЗИ. В составе программной и/или аппаратной части ЗХП должно быть интегрировано СКЗИ, соответствующее классу не ниже АК1.

Защищенный протокол взаимодействия. ЗХП должен обеспечивать унифицированный протокол обмена данными с внешними субъектами, включая другие ЗХП, по протоколу, все элементы которого (допускается исключения для элементов начальной инициализации и завершения протокола) зашифрованы и целостность которых фиксирована. Для организации протокола должна быть использована встроенная СКЗИ.

Гарантии проектирования. Архитектура построения аппаратной и программной компонент ЗХП должна обеспечивать выполнение приведенных выше требований.

Требования к классу АКХ2

Идентификация и аутентификация владельца. В ЗХП должна быть предусмотрена возможность идентификации и аутентификации владельца (пользователя). Вероятность ложной аутентификации на любую попытку доступа за время эксплуатации ЗХП не должна превосходить 10^{-8}.

Идентификация внешних субъектов. ЗХП должен идентифицировать обращения со стороны внешних субъектов, в том числе и со стороны других ЗХП, и отличать их от обращений пользователя и внутренних программ. Должна быть реализована двухсторонняя криптографическая аутентификация внешних субъектов и программ ЗХП. Алгоритм двухсторонней аутентификации формулируется в СТЗ. Для реализации алгоритма должен использоваться встроенный СКЗИ.

Разграничение доступа к объектам. Требования включают соответствующие требования класса АКХ3. Дополнительно в ЗХП должна быть реализована непротиворечивая модель управления доступом к объектам на основе формальной модели прав на «чтение – запись» соответствующих объектов.

Контроль целостности. ЗХП должен обеспечивать контроль целостности внутренних данных, перечень которых определен в СТЗ. Алгоритм проверки целостности должен обеспечивать величину вероятности ошибки, не превосходящую 10^{-9} для каждого объекта контроля. При нарушении целостности должны запускаться механизмы стирания объектов, перечень которых также указывается в СТЗ.

Регистрация событий. Требования полностью соответствуют классу АКХ3.

Защита от проникновения. Требования включают соответствующие требования класса АКХ3. Дополнительно должна быть обеспечена

защита от бесконтактного съема информации с ЗХП, а также невозможность влияния на внутренние данных ЗХП за счет манипуляций управляющими сигналами и напряжением источника питания.

Гарантии стирания. Требования полностью соответствуют классу АКХ3.

Гарантии хранения. Требования полностью соответствуют классу АКХ3.

Надежность функционирования ЗХП. Вероятность отклонения от заданных алгоритмов работы ЗХП, вызванного неисправностями и сбоями в работе его аппаратных и программных средств, не должна превосходить значения 10^{-5} в течение периода времени между двумя последовательными тестированиями.

Тестирование. Требования включают соответствующие требования класса АКХ3. Дополнительно ЗХП должен содержать процедуры тестирования защищенного протокола взаимодействия и процедуры тестирования системы разграничения доступа к данным.

Используемое СКЗИ. В составе программной и/или аппаратной части ЗХП должно быть интегрировано СКЗИ, соответствующее классу не ниже АК2.

Защищенное хранение данных. ЗХП должен обеспечивать работу с хранимыми данными, зашифрованными с использованием персональной информации пользователя, а также подписанными ЭЦП владельца либо других лиц, чьи сертификаты зарегистрированы в системах ведомственных или государственных УЦ установленным порядком.

Защищенный протокол взаимодействия. Требования включают соответствующие требования класса АКХ3. Дополнительно ЗХП должен поддерживать второй уровень защиты дополнительно к реализованному защищенному протоколу взаимодействия и состоящий в простановке (проверке) ЭЦП под каждым объектом, передаваемым (получаемым) в рамках указанного протокола.

Операции над хранимыми объектами. Операции над объектами ЗХП должны выполняться только с учетом свойств объекта (экспортируемый/неэкспортируемый и т. д.). Расширение функциональности операций с объектами в рамках дополнительных программных модулей, загружаемых в ЗХП, допускается только после проверки ЭЦП над указанными программными модулями.

Гарантии проектирования. Архитектура построения аппаратной и программной компонент ЗХП должна обеспечивать выполнение и обоснование выполнения приведенных выше требований.

Гарантии тиражирования и сопровождения. Должна быть реализована процедура аудита состояний ЗХП. Все дополнительные программные модули, разработанные для расширения функциональности ЗХП, должны быть подписаны ЭЦП разработчика.

Требования к классу АКХ1

Идентификация и аутентификация владельца. В ЗХП должна быть предусмотрена возможность идентификации и аутентификации владельца (пользователя). Вероятность ложной аутентификации на любую попытку доступа за время эксплуатации ЗХП не должна превосходить 10^{-9}.

Идентификация внешних субъектов. Требования полностью соответствуют классу АКХ2.

Разграничение доступа к объектам. Требования включают соответствующие требования класса АКХ2. Должны быть представлены формальные обоснования модели разграничения доступа и работы с объектами ЗХП.

Контроль целостности. Требования полностью соответствуют классу АКХ2.

Регистрация событий. Требования полностью соответствуют классу АКХ3.

Защита от проникновений. Требования полностью соответствуют классу АКХ2.

Гарантии стирания. Требования включают соответствующие требования класса АКХ3. Дополнительно должны быть предоставлены обоснования гарантированного стирания. Стирание должно быть поддержано на аппаратном уровне технологией, описанной в СТЗ.

Гарантии хранения. Требования полностью соответствуют классу АКХ3.

Надежность функционирования ЗХП. Вероятность отклонения от заданных алгоритмов работы ЗХП, вызванного неисправностями и сбоями в работе его аппаратных и программных средств, не должна превосходить значения 10^{-6} в течение периода времени между двумя последовательными тестированиями.

Тестирование. Требования полностью соответствуют классу АКХ2.

Используемое СКЗИ. В составе программной и/или аппаратной части ЗХП должно быть интегрировано СКЗИ, соответствующее классу не ниже АК3.

Защищенное хранение данных. Требования полностью соответствуют классу АКХ2.

Защищенный протокол взаимодействия. Требования полностью соответствуют классу АКХ2.

Операции над хранимыми объектами. Требования полностью соответствуют классу АКХ2.

Гарантии проектирования. Архитектура построения аппаратной и программной компонент ЗХП должна обеспечивать выполнение и обоснование выполнения приведенных выше требований. Должна быть построена формальная модель функционирования ЗХП, включая описание защищенного протокола взаимодействия и выполнение операций над объектами ЗХП.

Гарантии тиражирования и сопровождения. Требования полностью соответствуют классу АКХ2.

4.11. Инфраструктура доверия и защита аутентифицирующей информации

Для предотвращения описанной выше атаки посредника при распределении ключевой и аутентифицирующей информации в защищенной КС используется метод цифровых сертификатов. **Сертификат** представляет собой электронный документ, подтверждающий взаимосвязь между открытым ключом и идентификационными данными его владель-

ца посредством ЭЦП с использованием секретного ключа доверенной третьей стороны.

Сертификат представляет собой некоторое удостоверение, позволяющее произвести аутентификацию его владельца аналогично паспорту или водительскому удостоверению. Добавление дополнительной информации позволяет конкретизировать область применения сертификата.

В последнее время преимущественное распространение получили такие средства и системы защиты информации, в которых ведущую роль играют методы криптографической защиты информации, основанные на асимметричных криптосистемах, а симметричные играют по отношению к ним подчиненную роль и используются как необходимое средство в тех случаях, когда асимметричные методы не удовлетворяют требованиям высокой производительности криптосистемы[2]. В связи с этим ведущую роль приобретает именно организация управления ключами асимметричных криптосистем. Она, в свою очередь, включает две подзадачи: управление частными секретными ключами участников и управление их открытыми ключами.

Задача управления секретными ключами здесь проще, чем в симметричных криптосистемах, так как секретные ключи никогда не выходят за пределы собственности их владельцев: нет необходимости передавать их по каким-либо каналам связи, распространять среди других участников. Оставшиеся задачи генерации, надежного хранения и распространения секретных ключей участниками асимметричных криптосистем вполне решаемы традиционными средствами, хорошо отработанными в процессе развития симметричных криптосистем.

Как отмечалось выше, наиболее сложным для обеспечения безопасности открытых ключей является этап распространения их в КС. Известен целый ряд методов решения этой задачи.

- передача открытого ключа через доверенный канал связи, обеспечивающий секретность, целостность и аутентичность;
- прямой доступ субъектов в доверенную базу данных (файл, директорию);
- использование доверенного сервера в режиме реального времени;
- использование сервера в режиме отложенного доступа и метода сертификации открытых ключей;
- использование криптосистем, неявно гарантирующих аутентичность открытых ключей, в том числе, системы, основанные на идентификаторах; системы с неявно сертифицированными открытыми ключами.

Среди всех перечисленных методов преобладающим на практике является метод сертификации открытых ключей. Остальные в силу различных многочисленных причин находят лишь ограниченное применение.

Задача управления открытыми ключами является новой по сравнению с задачами, возникшими в симметричных криптосистемах, и требует своих особых подходов. Из-за необходимости ее решения возникло отдельное научно-практическое направление прикладной криптографии, а сама идея решения этой задачи выразилась в создании специальной инфраструктуры в рамках криптосистемы, получившей наименование

2. Запечников С. В. Криптографические протоколы и их применение в финансовой и коммерческой деятельности: Учебное пособие для вузов. – М.: Горячая линия – Телеком, 2007. – 320 с.

инфраструктуры открытых ключей (что является дословным переводом с английского термина *Public Key Infrastructure – PKI*).

Инфраструктура открытых ключей (ИОК) – это универсальная концепция организованной поддержки криптографических средств защиты информации в крупномасштабных информационных системах в соответствии с принятыми в них политиками безопасности, которая реализует управление криптографическими ключами на всех этапах их жизненного цикла, обеспечивая взаимодействие всех средств защиты распределенной системы.

Логически ИОК объединяет механизмы, субъекты, правила и взаимосвязи, которые необходимы для доступа к криптографическим ключам и для ассоциирования открытых ключей со своими владельцами.

Физически ИОК состоит из программ, форматов данных, коммуникационных протоколов, политик и процедур, требуемых для использования в организации криптосистем с открытым ключом.

ИОК может быть интегрирована со всеми основными ОС, сетевым программным обеспечением и основными прикладными программами. Чтобы использовать сервисы, предоставляемые ИОК, в прикладных программах, они должны быть адаптированы, или должны быть разработаны новые прикладные программы.

Важно иметь в виду, что создание инфраструктуры открытых ключей имеет целью комплексную поддержку всего жизненного цикла открытых криптографических ключей в целом, а не только каких-либо отдельных его фаз, например, распространения ключей (хотя последняя задача и является технически наиболее сложной).

Международные организации, занимающиеся стандартизацией информационных технологий, разработали ряд *моделей ИОК*, из которых наиболее известными и распространенными являются модели SPKI, PKIX и APKI.

Модель SPKI (Simple Public Key Infrastructure) – наиболее простая модель ИОК, разработанная Международной инженерной организацией по сети Internet IETF. Проект SPKI направлен на создание простой, минимально необходимой криптографической инфраструктуры и представлен двумя документами серии RFC (Request for Comments): RFC 2692 – SPKI Requirements; RFC 2693 – SPKI Certificate Theory.

Модель PKIX (Public Key Infrastructure for X.509) обеспечивает существенно более широкую функциональность и основана на стандарте Международного телекоммуникационного Союза ITU X.509. Ее составляют несколько уже утвержденных стандартов RFC и порядка двадцати проектов (draft) стандартов RFC. К утвержденным стандартам относятся:

• RFC 2459 – Internet X.509 Public Key Infrastructure Certificate and CRL;

• RFC 2510 – Internet X.509 Public Key Infrastructure Certificate Management Protocols;

• RFC 2511 – Internet X.509 Certificate Request Message Format;

• RFC 2527 – Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework;

• RFC 2528 – Internet X.509 Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates;

• RFC 2559 – Internet X.509 Public Key Infrastructure Operational Protocols – LDAP v2;

• RFC 2560 – Internet X.509 Public Key Infrastructure Online Certificate Status Protocol – OCSP;

• RFC 2585 – Internet X.509 Public Key Infrastructure Operational Protocols – FTP and HTTP;

• RFC 2559 – Internet X.509 Public Key Infrastructure LDAP v2 Schema.

Рассмотрим основное содержание *метода сертификации открытых ключей*.

Пусть имеется криптосистема, включающая большое число участников (***рис. 2***). Среди участников криптосистемы выделяется один, специальный, которому доверяют все остальные. Он получает название *удостоверяющий центр,* или *центр сертификации ключей*, или *агентство сертификации* (*Certification Authority – CA*). Его функции может выполнять, например, администратор системы, оснащенный соответствующим аппаратным и программным обеспечением (сервер регистрации и сертификации ключей). Все остальные участники являются обычными, «рядовыми» абонентами криптосистемы.

При введении в систему каждого из этих участников – возьмем для примера участника *A* – он должен пройти процедуру регистрации в криптосистеме (что соответствует первой фазе жизненного цикла его криптографического ключа). Для этого он контактирует с удостоверяющим центром, чтобы зарегистрировать свой открытый ключ и получить от него так называемый *сертификат* своего открытого ключа. Удостоверяющий центр должен проверить представленные ему учетные данные, а также (что очень важно!) знание секретного ключа, соответствующего представленному для регистрации открытому ключу. Решить эту задачу можно различными способами: в самом простом случае удостоверяю-

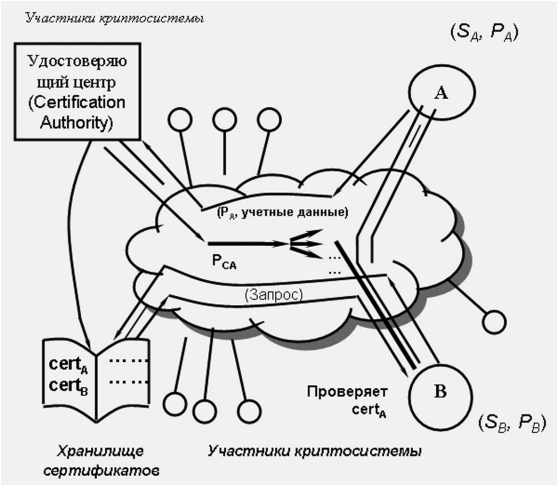


Рис. 2. Процессы получения и использования сертификатов участниками криптосистемы

щий центр может попросить *А* зашифровать на своем секретном ключе текст заданного формата и проверить правильность его расшифрования при помощи представленного открытого ключа, а можно воспользоваться протоколом доказательства знания с нулевым разглашением знания.

Сертификат открытого ключа – специальная структура данных, состоящая из полей данных и поля подписи. Поле данных содержит, как минимум, какие-либо признаки абонента (идентификатор, атрибуты) и его открытый ключ. Поле подписи – это цифровая подпись удостоверяющего центра под полем данных, логически связывающая признаки абонента с его открытым ключом.

Все абоненты, заинтересованные в связи с абонентом *А*, получают впоследствии его сертификат либо путем обмена с абонентом *А*, либо извлекая его из открытого общедоступного справочника, который заводится в криптосистеме. Сертификат, таким образом, является средством для хранения, распространения и передачи через небезопасные каналы связи открытых ключей без опасения их необнаружимого изменения.

Различают две формы сертификатов открытых ключей: идентификационные и атрибутные.

В *идентификационном сертификате* обязательно присутствует идентификатор субъекта – владельца ключа, по которому можно однозначно установить его личность. Основным стандартом по идентификационным сертификатам является стандарт Международного телекоммуникационного Союза ITU X.509. В соответствии со стандартом X.509 сертификат имеет следующий формат (***рис. 3***).

Версия сертификата
Серийный номер сертификата
Идентификатор алгоритма цифровой подписи, используемого удостоверяющим центром
Имя удостоверяющего центра (директориальное имя по стандарту X.500)
Период действия сертификата
Имя владельца открытого ключа (директориальное имя по стандарту X.500)
Информация об открытом ключе владельца: **• идентификатор алгоритма;** **• значение открытого ключа.**
Уникальный идентификатор удостоверяющего центра, выпустившего сертификат (v2)
Уникальный идентификатор владельца открытого ключа (v2)
Поле расширения (v3): содержание не определено.
Цифровая подпись удостоверяющего центра под всеми предыдущими полями

Рис. 3. Формат сертификата открытого ключа по стандарту ITU X.509

Сертификат состоит из двух полей: поля данных и поля подписи. Поле данных имеет формат, описанный в стандарте. Поле подписи содержит цифровую подпись удостоверяющего центра под полем данных. Существуют две версии этого стандарта, которые принято обозначать X.509v2 и X.509v3. Различие между ними заключается в том, что в версии 2 определены поля «Уникальный идентификатор удостоверяющего центра, выпустившего сертификат» и «Уникальный идентификатор владельца открытого ключа», а в версии 3 стандарта эти поля исключены, но предусмотрено наличие в поле данных дополнительного поля расширения, содержание которого не определено: оно может специфицироваться другими стандартами, зависеть от области применения информационной системы и т. п.

Самыми разработанными и широко применяемыми на практике логическими моделями ИОК на базе идентификационных сертификатов являются следующие:

• X9.55 – стандарт США для финансовой индустрии;

• PKIX – проект стандарта IETF на базе стандарта X.509v3, адаптирующий положения этого стандарта для использования в Интернет;

• APKI – архитектура для ИОК, описанная в документах The Open Group.

Использование идентификационных сертификатов не всегда желательно для пользователей, так как при этом может происходить доступ к информации, не имеющей отношения к тому случаю, по которому необходим данный конкретный факт доступа к сертификату. Возможность однозначно установить личность владельца по сертификату может привести к установлению «тотального контроля» над участниками криптосистемы. В связи с этим было предложено использовать другую форму сертификатов.

Атрибутные сертификаты связывают открытый ключ с одним или более «атрибутов», которые в соответствии со стандартом Международного телекоммуникационного Союза X.501 (эквивалентный стандарт – ISO/IEC 9594-2) определяются как «информация любого типа». Таким образом, один и тот же участник в зависимости от ситуации и используемой прикладной программы может предстать в разных «ипостасях», между которыми невозможно установить однозначную связь. К примеру, атрибутом может быть роль пользователя в информационной системе, например, путем указания его должности. Тогда можно реализовать модель управления доступа «по ролям», т. е. участники системы, занимающие одну и ту же должность, имеют абсолютно одинаковые права в системе, и невозможно установить, кто именно из них совершил конкретное действие с применением данного конкретного сертификата.

Наиболее разработанными и широко применяемыми на практике логическими моделями инфраструктуры открытых ключей на основе атрибутных сертификатов являются:

• X9.57 – стандарт США для финансовой индустрии;

• SPKI – проект стандарта IETF для использования в сети Интернет.

Удостоверяющий центр (Центр сертификации открытых ключей) – это специально выделенный участник криптосистемы, которому доверяют все остальные участники («центр доверия»), чья подпись служит гарантией подлинности ключей и который выполняет следующие функции:

• сбор сведений об участниках системы, необходимых для сертификации: имя, почтовый адрес, права доступа, должность, номер кредитной карты и т. п. (зависит от конкретного приложения);

• генерация и рассылка (либо помещение в общедоступное хранилище) сертификатов открытых ключей;

• уничтожение сертификатов с истекшим сроком годности;

• обновление сертификатов;

• аннулирование сертификатов.

Аннулирование сертификата может потребоваться в случаях, когда срок санкционированного использования открытого ключа участника системы прерывается досрочно, ранее, чем это предусмотрено принятым в системе регламентом, например, при компрометации секретного ключа участника криптосистемы, соответствующего данному открытому ключу, при удалении (выбытии) владельца ключа из системы, при смене роли пользователя в системе (перемещении пользователя). Аннулирование сертификата – это чрезвычайное обстоятельство, о котором необходимо оповестить всех участников криптосистемы. Существуют два способа решения этой задачи:

• *проверка статуса сертификата в режиме реального времени;* для этого требуется выполнение специального протокола с удостоверяющим центром, который отвечает на вопрос, не был ли запрошенный сертификат аннулирован;

• *периодическое создание и рассылка списков аннулированных сертификатов* (CRL – Certificate Revocation List); формат CRL определен в стандарте ITU X.509 v2 (***рис. 4***).

Рис. 4. Формат списка аннулированных сертификатов по стандарту ITU X.509 v2

Второй способ на практике используется чаще. При этом необходимо отметить, что у него есть один существенный недостаток: между двумя последовательными рассылками списка аннулированных сертификатов всегда существует какой-то временный «зазор», т. е. об аннулировании сертификата какого-либо участника все остальные участники узнают не мгновенно, а только по прошествии некоторого времени. Наличие такого разрыва создает угрозу несанкционированного использования аннулированного ключа.

Удостоверяющий центр рассылает всем участникам системы свой открытый ключ, который нужен им для проверки подписи на сертифи-

катах. Считается, что подменить его невозможно в силу трех причин: массовости рассылки, периодического повтора и общедоступности.

В криптосистемах с большим числом участников или с большой интенсивностью потока требований к удостоверяющему центру функции регистрации участников нередко возлагают на специально выделяемый центр регистрации (Registration Authority – RA).

Сертификаты всех участников криптосистемы могут либо храниться в специальном общедоступном хранилище, либо рассылаться по сети. На практике преимущественно используется первый способ, причем физически хранилище сертификатов чаще всего реализуется либо с использованием директориального сервиса (как директория, к которой открыт доступ на чтение всем участникам), либо веб-сервиса (как вебстраница, с которой все участники криптосистемы могут забирать сертификаты).

Преимущество сертификата в том, что два участника системы, доверяющие одному и тому же удостоверяющему центру, могут не знать и не хранить открытые ключи никаких других абонентов, а при необходимости обратиться в хранилище сертификатов и получить необходимые ключи. Для этого ему достаточно знать только открытый ключ удостоверяющего центра. Это позволяет применять метод сертификации открытых ключей в криптосистемах со сколь угодно большим и даже неопределенным числом участников, где все участники не покрыты сетью непосредственных контактов между собой.

Теперь для того, чтобы узнать открытый ключ любого интересующего его абонента, участнику криптосистемы (обозначим его *B*) необходимо однократно приобрести аутентичный открытый ключ удостоверяющего центра (что технически реализовать не сложно). Далее для установления связи с абонентом *A* ему необходимо:

1) приобрести сертификат открытого ключа *A* одним из следующих способов: обратившись в хранилище сертификатов, непосредственно получив его от удостоверяющего центра или от абонента *A* (зависит от порядка, установленного в системе);

2) выполнить процедуру проверки сертификата, состоящую из следующих действий:

• проверки текущей даты и времени и сравнения с периодом действия сертификата;

• проверки действительности в данный момент времени открытого ключа самого удостоверяющего центра;

• проверки подписи удостоверяющего центра на сертификате открытого ключа абонента *A*, используя открытый ключ удостоверяющего центра;

• проверки, не был ли сертификат аннулирован к текущему моменту времени;

3) в случае, если все проверки окончились с положительным результатом, принять открытый ключ, извлеченный из сертификата *A* как аутентичный ключ.

Далее *B* может использовать открытый ключ абонента *A* для выполнения любых необходимых ему криптографических алгоритмов или протоколов. Например, участники *A* и *B*, приобретя таким образом открытые ключи друг друга, могут выработать общий секретный ключ для симметричной криптосистемы.

Хотя сертификат является вспомогательным средством для транспортировки и обеспечения подлинности открытого ключа, он, как и сам ключ, имеет свой жизненный цикл. В его жизненном цикле можно выделить те же стадии и ряд состояний, в которых пребывает сертификат: создание, рассылка, штатное использование, обновление или аннулирование, уничтожение сертификата.

Недостаток метода сертификации открытых ключей заключается в том, что участникам криптосистемы после проверки сертификата часто необходим доступ в реальном масштабе времени в базу данных с информацией о других участниках криптосистемы (например, для проверки аннулирования сертификата либо для того, чтобы получить какие-то дополнительные данные об этом участнике, необходимые для работы прикладных программ). Это не всегда удобно, поэтому сейчас наблюдается «откат» к модели ИОК с удостоверяющим центром, работающим не в отложенном режиме, а в реальном масштабе времени.

Таким образом, личный цифровой сертификат включает личный открытый ключ пользователя, его идентификационные данные и подписан ЭЦП некоторого **уполномоченного органа**.

Уполномоченными органами для заверения личных открытых ключей пользователей и его идентификационные данные выступают аккредитованные государством Удостоверяющие центры.

Использование электронной цифровой подписи на территории РФ регламентируется Федеральным Законом «Об электронной цифровой подписи» от 10 января 2002 года № 1-ФЗ[3].

Содержащиеся в Федеральном законе «Об электронной цифровой подписи» подходы согласуются с аналогичными зарубежными законодательными актами и учитывают мировой опыт использования ЭЦП. Большинство зарубежных законодательных актов признает аналогом собственноручной подписи исключительно электронную цифровую подпись[4], реализованную на основе применения асимметричного криптографического преобразования. Так, законодательные акты американских штатов Вашингтон и Юта также опираются на признание свойств асимметричных криптографических алгоритмов. Вместе с тем Федеральный закон США об электронных подписях в национальной и глобальной коммерции в отличие от упомянутых законодательных актов отдельных штатов этого не делает, за что и подвергается критике. По заключению ряда американских экспертов, этот закон, введенный в действие с 1 октября 2000 года, не обеспечивает сколько-нибудь существенной защиты потребителя от фальсификации, так как в нем не зафиксированы надежные (криптографические) методы обеспечения ее подлинности. Интересно отметить венгерский закон об электронной подписи. Закон признает возможность использования в обращении трех различных видов электронной подписи. Но при этом устанавливает, что только документ, подписанный электронной подписью, выполненной на основе асимметричных криптографических алгоритмов, имеет ту же юридическую силу, что и документ на бумажном носителе, собственноручно подписанный человеком.

В Законе РФ определен как порядок использования ЭЦП, так и порядок функционирования Удостоверяющих центров (УЦ). При использо-

3. (http://www.internet-law.ru/intlaw/laws/ecp.htm)

4. Антон Серго «Интернет и Право». М.: «Бестселлер», 2003 – 272 с.

вании электронной цифровой подписи должны быть соблюдены следующие положения:

• при создании ключей электронных цифровых подписей для использования в компьютерной системе общего пользования должны применяться только **сертифицированные** (одобренные уполномоченно государственной организацией, которой в настоящее время является 8 Центр ФСБ РФ) средства электронной цифровой подписи;

• сертификат ключа подписи должен содержать следующие сведения:

✓ уникальный регистрационный номер сертификата ключа подписи, даты начала и окончания срока действия сертификата ключа подписи, находящегося в реестре удостоверяющего центра;

✓ фамилию, имя и отчество владельца сертификата ключа подписи или псевдоним владельца

✓ открытый ключ электронной цифровой подписи;

✓ наименование средств электронной цифровой подписи, с которыми используется данный открытый ключ электронной цифровой подписи;

✓ наименование и место нахождения УЦ, выдавшего сертификат ключа подписи;

✓ сведения об отношениях, при осуществлении которых электронный документ с электронной цифровой подписью будет иметь юридическое значение;

✓ сертификат ключа подписи должен быть внесен УЦ в реестр сертификатов ключей подписей не позднее даты начала действия сертификата ключа подписи.

При использовании УЦ необходимо выполнять следующие требования:

• по хранению ключевой информации:

✓ срок хранения сертификата ключа подписи в форме электронного документа в УЦ определяется договором между этим центром и владельцем сертификата ключа подписи; при этом обеспечивается доступ участников информационной системы в УЦ для получения сертификата ключа подписи;

✓ срок хранения сертификата ключа подписи в форме электронного документа в УЦ после аннулирования сертификата ключа подписи должен быть не менее установленного Федеральным Законом об электронной цифровой подписи срока исковой давности для отношений, указанных в сертификате ключа подписи; по истечении указанного срока хранения сертификат ключа подписи исключается из реестра сертификатов ключей подписей и переводится в режим архивного хранения; срок архивного хранения составляет не менее чем пять лет; порядок выдачи копий сертификатов ключей подписей в этот период устанавливается в соответствии с законодательством РФ;

✓ сертификат ключа подписи в форме документа на бумажном носителе хранится в порядке, установленном законодательством РФ об архивах и архивном деле;

• по статусу Удостоверяющего центра:

✓ УЦ, выдающим сертификаты ключей подписей для использования в информационных системах общего пользования, должно быть юридическое лицо, выполняющее функции, предусмотренные Федеральным

Законом об электронной цифровой подписи, при этом УЦ должен обладать необходимыми материальными и финансовыми возможностями, позволяющими нести гражданскую ответственность перед пользователями сертификатов ключей подписей за убытки, которые могут быть понесены ими вследствие недостоверности сведений, содержащихся в сертификатах ключей подписей; требования, предъявляемые к материальным и финансовым возможностям УЦ, определяются Правительством РФ[5] по представлению уполномоченного федерального органа исполнительной власти;

✓ статус УЦ, обеспечивающего функционирование корпоративной информационной системы, определяется ее владельцем или соглашением участников этой системы;

✓ деятельность УЦ подлежит лицензированию в соответствии с законодательством РФ о лицензировании отдельных видов деятельности;.

• по деятельности Удостоверяющего центра:

Удостоверяющий центр:

✓ изготавливает сертификаты ключей подписей;

✓ создает ключи электронных цифровых подписей по обращению участников информационной системы с гарантией сохранения в тайне закрытого ключа электронной цифровой подписи;

✓ приостанавливает и возобновляет действие сертификатов ключей подписей, а также аннулирует их;

✓ ведет реестр сертификатов ключей подписей, обеспечивает его актуальность и возможность свободного доступа к нему участников информационных систем;

✓ проверяет уникальность открытых ключей электронных цифровых подписей в реестре сертификатов ключей подписей и архиве Удостоверяющего центра;

✓ выдает сертификаты ключей подписей в форме документов на бумажных носителях и (или) в форме электронных документов с информацией об их действии;

✓ осуществляет по обращениям пользователей сертификатов ключей подписей подтверждение подлинности электронной цифровой подписи в электронном документе в отношении выданных им сертификатов ключей подписей;

✓ может предоставлять участникам информационных систем иные связанные с использованием электронных цифровых подписей услуги.

• изготовление сертификатов ключей подписей осуществляется на основании заявления участника информационной системы, которое содержит сведения, указанные в ст. 6 Федерального Закона об электронной цифровой подписи и необходимые для идентификации владельца сертификата ключа подписи и передачи ему сообщений; заявление подписывается собственноручно владельцем сертификата ключа подписи; содержащиеся в заявлении сведения подтверждаются предъявлением соответствующих документов;

• при изготовлении сертификатов ключей подписей УЦ оформляются в форме документов на бумажных носителях два экземпляра сертификата ключа подписи, которые заверяются собственноручными подписями владельца сертификата ключа подписи и уполномоченного

5. В других странах полномочия УЦ также определяются правительствами или иными государственными регулирующими органами

лица удостоверяющего центра, а также печатью УЦ; один экземпляр сертификата ключа подписи выдается владельцу сертификата ключа подписи, второй остается в УЦ;

• услуги по выдаче участникам информационных систем сертификатов ключей подписей, зарегистрированных УЦ, одновременно с информацией об их действии в форме электронных документов оказываются безвозмездно.

Кроме того, действующее законодательство регламентирует отношения между УЦ и уполномоченным федеральным органом исполнительной власти:

• УЦ до начала использования электронной цифровой подписи уполномоченного лица Удостоверяющего центра для заверения от имени УЦ сертификатов ключей подписей обязан представить в уполномоченный федеральный орган исполнительной власти сертификат ключа подписи уполномоченного лица УЦ в форме электронного документа, а также этот сертификат в форме документа на бумажном носителе с собственноручной подписью указанного уполномоченного лица, заверенный подписью руководителя и печатью УЦ;

• уполномоченный федеральный орган исполнительной власти ведет единый государственный реестр сертификатов ключей подписей, которыми УЦ, работающие с участниками информационных систем общего пользования, заверяют выдаваемые ими сертификаты ключей подписей, обеспечивает возможность свободного доступа к этому реестру и выдает сертификаты ключей подписей соответствующих уполномоченных лиц УЦ;

• электронные цифровые подписи уполномоченных лиц УЦ могут использоваться только после включения их в единый государственный реестр сертификатов ключей подписей; использование этих электронных цифровых подписей для целей, не связанных с заверением сертификатов ключей подписей и сведений об их действии, не допускается.

Уполномоченный федеральный орган исполнительной власти:

• осуществляет по обращениям физических лиц, организаций, федеральных органов государственной власти, органов государственной власти субъектов РФ и органов местного самоуправления подтверждение подлинности электронных цифровых подписей уполномоченных лиц УЦ в выданных ими сертификатах ключей подписей;

• осуществляет в соответствии с положением об уполномоченном федеральном органе исполнительной власти иные полномочия по обеспечению действия Федерального Закона об электронной цифровой подписи.

УЦ при изготовлении сертификата ключа подписи принимает на себя следующие обязательства по отношению к владельцу сертификата ключа подписи:

• вносить сертификат ключа подписи в реестр сертификатов ключей подписей;

• обеспечивать выдачу сертификата ключа подписи обратившимся к нему участникам информационных систем;

• приостанавливать действие сертификата ключа подписи по обращению его владельца;

• уведомлять владельца сертификата ключа подписи о фактах, которые стали известны УЦ и которые существенным образом могут сказаться на возможности дальнейшего использования сертификата ключа подписи;

• иные установленные нормативными правовыми актами или соглашением сторон обязательства.

Владелец сертификата ключа подписи обязан:

• не использовать для электронной цифровой подписи открытые и закрытые ключи электронной цифровой подписи, если ему известно, что эти ключи используются или использовались ранее;

• хранить в тайне закрытый ключ электронной цифровой подписи;

• немедленно требовать приостановления действия сертификата ключа подписи при наличии оснований полагать, что тайна закрытого ключа электронной цифровой подписи нарушена.

Для проверки ЭЦП под сертификатом используется сертификат уполномоченного органа, который должны иметь все участники.

Процесс синтеза и анализа криптографических алгоритмов отличается высокой сложностью и трудоемкостью. В связи с этим практически во всех странах, обладающих развитыми криптографическими технологиями, разработка и внедрение криптографических средств относится к сфере государственного регулирования. Государственное регулирование включает, как правило, *лицензирование* деятельности, связанной с разработкой и эксплуатацией криптографических средств, *сертификацию* криптографических средств и *стандартизацию* алгоритмов криптографических преобразований.

В России в настоящее время организационно-правовые и научно-технические проблемы синтеза и анализа средств криптографической защиты информации находятся в компетенции ФСБ РФ.

Правовая сторона разработки и использования криптографических средств регламентируется в основном указом Президента Российской Федерации от 03.04.95 № 334 с учетом принятых ранее законодательных и нормативных актов РФ.

Дополнительно учитываемой законодательной базой являются законы: «О федеральных органах правительственной связи и информации», «О государственной тайне», «Об информации, информатизации и защите информации», «О сертификации продукции и услуг».

Порядок *сертификации* криптографических средств установлен «Системой сертификации средств криптографической защиты информации РОСС.RU.0001.030001 Госстандарта России.

Стандартизация алгоритмов криптографических преобразований включает всесторонние исследования и публикацию в виде стандартов элементов криптографических процедур с целью использования разработчиками апробированных криптографически стойких преобразований, обеспечения возможности совместной работы различных криптографических средств, а также возможности тестирования и проверки соответствия реализации криптографических средств заданному стандартом алгоритму.

В России приняты и действуют следующие стандарты – алгоритм криптографического преобразования 28147-89, алгоритмы хеширования, выработки и проверки цифровой подписи Р34.10 и Р34.11.

Из зарубежных стандартов широко известны и применяются алгоритмы шифрования *DES*, *RC2*, *RC4*, алгоритмы хеширования *MD2*, *MD4* и *MD5*, алгоритмы простановки и проверки цифровой подписи *DSS* и *RSA*.

5. УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ В КОМПЬЮТЕРНОЙ СИСТЕМЕ

5.1. Введение

Проблема управления безопасностью (далее – управления) является весьма важной для выполнения ПБ в течение всего времени существования защищенной КС. В рамках введенных выше понятий управление подразумевает изменение объекта, хранящего информацию о множестве L (объекта управления, ОУ) в соответствии с текущим состоянием объектов, субъектов и пользователей. Например, возможно изменение перечня доступных какому-либо субъекту объектов, либо изменение во множестве объектов-источников, доступных для порождения субъектов (для ИПС). Иначе говоря, управление защитой в некоторый момент времени описывает политику безопасности применительно к текущему состоянию КС.

Целесообразно процесс управления рассматривать в рамках существования субъекта реализации ПБ (МБО) и субъекта гарантирования ПБ (МБС). Очевидно, что управление должно быть организовано таким образом, чтобы ПБ при изменениях в ОУ не нарушалась (т.е. в ОУ не включались бы потоки из множества N).

Итак, в данной части изучаются вопросы управления защитой, а именно – формирование и изменение объектов управления (ОУ) для субъектов реализации политики безопасности (МБО) и субъектов гарантий политики безопасности (МБС). Управление описывает методы формирования ОУ, доставки ОУ на локальные сегменты КС, а также методы оперативного изменения ОУ при изменении прав пользователей.

Ранее проблеме управления средствами защиты уделялось достаточно небольшое внимание, однако практика проектирования, реализации и эксплуатации защищенных КС показала, что в данной области имеется ряд проблем.

В первую очередь это проблема, связанная с формированием списков объектов-источников, необходимых для порождения субъектов КС.

Данная проблема связана с тем, что последовательность порождения субъектов в сложных программных комплексах динамически изменяется с изменением их конфигурации и администратору защиты заранее неизвестна. Вместе с тем очевидно, что исключение какого-либо субъекта-сервиса из числа разрешенных сказывается на доступности ресурса, что для систем критичного применения недопустимо (на практике это означает невозможность запуска сервиса или вызова функции и, как следствие, отказ в обслуживании). В общем случае, отсутствие какого-либо объекта-источника в ОУ МБС не позволит использовать порождаемый им субъект, что, несомненно, сказывается на пользовательских свойствах КС.

Классический подход к управлению безопасностью связан с обеспечением физического доступа администратора (пользователя) к локаль-

ному сегменту КС и выполнению действий по управлению при помощи субъектов управления, имеющихся на ЛС КС, либо хранящихся на внешнем носителе. Зачастую управление аппаратной компонентой защиты подразумевает работы по извлечению элементов аппаратной защиты из ПЭВМ с целью изменения списков пользователей и т.д. Уже в развитой топологически сложной ЛВС (с несколькими серверами, десятками рабочих мест) метод администрирования с физическим доступом испытывает серьезные трудности. Затраты времени на администрирование одной станции так велики, что оперативное управление защитой всей сети практически невозможно.

Необходимо заметить также, что в ряде случаев необходима коррекция политики безопасности, связанная с тем, что может существовать пользователь-нарушитель, стремящийся получить доступ к ресурсу коллективного пользования, к которому он имеет ограниченный доступ. Проблема изменения объекта управления должна быть рассмотрена и в этом контексте. А именно, метод составления, транспортировки и использования объекта управления должен обеспечивать невозможность его модификации локальным пользователем с целью произвольного присвоения себе прав.

В данной части рассмотрены следующие важные теоретические вопросы компьютерной безопасности:

1. введение определения корректного управления и формализации процесса управления,

2. формулирование методов формирования и транспортировки объектов управления для различных конфигураций КС,

3. формулирование условий для реализации управления в различных конфигурациях без нарушения политики безопасности (т.е. обеспечение корректности управления в рамках введенных определений).

5.2. Модель управления безопасностью. Термины

Выше неоднократно рассматривалось задаваемое ПБ разделение всего множества потоков в КС на множества легального доступа L и нелегального N.

Реализацию данного множества в виде объекта КС будем называть объектом управления МБО (Ouo – объект управления объектами). В случае фиксации в объекте Ouo потоков множества L принято называть данный объект **белым списком разграничения доступа**, для потоков множества N – **черным списком**.

Аналогично, объект, содержащий аргументы (объекты-источники) для операции порождения процессов ***Create***, будем называть объектом управления МБС (Ous – объект управления субъектами).

Примем за аксиому наличие в составе субъектов КС субъекта управления (администрирования), который инициирует потоки между своими ассоциированными объектами и ОУ.

Как было указано выше, можно выделить потоки типа «запись» $Stream(S_i,O_k)->O_m$ (если O_k и O_m нетождественны) и потоки типа «чте-

ние» Stream(S_i,O_m)->O_k, $O_к$ – ассоциированный объект S_i. S_i в данном случае субъект управления.

Введем понятие управляемой КС.

Определение 1. КС называется управляемой, если в ней существует субъект (обозначим его Sa – субъект администрирования), для ассоциированных объектов которого существует поток к объекту управления.

Сам субъект, как было сказано выше, называют управляющим или администрирующим (управляющий субъект, УС). Пользователя, управляющего администрирующим субъектом логично назвать администратором безопасности КС, или, наоборот, только администратор безопасности должен иметь возможность порождения субъекта управления.

В случае разделения объекта управления на два независимых подобъекта можно говорить о системе, управляемой по объектам и по субъектам. Далее будем в необходимых случаях конкретизировать, о каком именно ОУ идет речь.

Если управляющий субъект принадлежит ЛС КС (в смысле принадлежности множеству субъектов ЛС КС), то управление будем называть локальным, если внешнему сегменту КС – удаленным.

Определение 2. Компьютерная система называется корректно управляемой, если поток к объекту управления существует только для субъекта управления.

При этом необходимо заметить, что доступ на чтение к объектам управления (или единому объекту) обязателен для МБО и МБС, поскольку в противном случае они не могут выполнять своих функций за отсутствием эталона, согласно которому производится разделение потоков или поиск разрешенных значений аргументов операции ***Create***.

Резюмируя, можно говорить о следующих свойствах ОУ и субъектов управления КС.

1. Субъект управления должен быть доступен для порождения только избранному пользователю (или нескольким пользователям), данного пользователя будем называть администратором (администраторами).

2. Только субъект управления должен иметь доступ на запись к объекту управления.

3. МБО и МБС должны иметь доступ на чтение к ОУ.

Сформулируем достаточное условие корректного управления.

Утверждение 1 (о корректном управлении в ИПС).

Если в КС поддерживается ИПС с контролем неизменности объектов-источников и существует МБО, который разрешает доступ на запись к ОУ только управляющему субъекту, то с момента активизации МБО управление в КС корректно.

Доказательство

По условию утверждения в КС существуют только субъекты, попарно корректные относительно друг друга, МБО и субъекта управления, следовательно, гарантированы только потоки, разрешенные МБО. Также по условию разрешен доступ к ОУ на запись только управляющему субъекту. Существование других субъектов, кроме входящих в ИПС, невозможно. Определение корректного управления выполнено в таком случае с момента активизации МБО. Утверждение доказано.

Смысл данного утверждения состоит в том, что условия, описывающие гарантии произвольной политики безопасности, и условия корректного управления как достаточные условия совпадают. Дополнительные требования для корректного управления корректируют политику безопасности КС (в части необходимости доступа на запись к ОУ только субъекту управления). Если ОУ является однотипным с другими объектами, рассматриваемыми МБО, то коррекция ПБ производится на этапе параметрических установок (например, запись в файл объекта управления разрешена только управляющему субъекту).

Важно описать также и условия нарушения корректности управления, которые связаны в ИПС с порождением некоторого произвольного субъекта, который имеет доступ к ОУ (поток типа «запись»).

Сформулируем другое важное утверждение.

Утверждение 2 (условия нарушения корректности управления).

При существовании ИПС с контролем неизменности объектов-источников и наличии корректного управления, нарушение ИПС (как возможность инициирования произвольного субъекта) возможно только при включении в ОУ МБС объекта-источника, порождающего указанный субъект.

Доказательство

От противного. Пусть при управлении не происходит включения объекта-источника для порождения некорректного субъекта, тогда порождение некорректного субъекта по условию и утверждениям части 1 невозможно. Противоречие.

Это достаточно очевидное утверждение, тем не менее, говорит о том, что «размыкание» гарантирующей выполнение ПБ программной среды в случае корректного управления может произойти только при включении администратором некорректного субъекта в состав программной среды (включение некорректного субъекта во множество субъектов эквивалентно включению в ОУ объекта-источника для данного субъекта). Следовательно, гарантии управления безопасностью (как гарантии поддержания ПБ в процессе управления) могут быть выполнены лишь частично (в смысле того, что в рамках рассматриваемой модели не описаны критерии включения субъектов в ИПС – данный вопрос решается администратором), если существует возможность дополнения множества субъектов.

С другой стороны, если администратор не имеет возможности дополнять ИПС новыми субъектами, то гарантии управления выполнены.

Если ИПС создает эксплуатационные гарантии политики безопасности, то при рассмотрении вопросов управления целесообразно говорить о гарантиях управления, которые описываются как достаточные условия приведенным выше утверждением.

Определение 3. КС с генерацией ИПС и контролем неизменности объектов-источников называется корректно управляемой в строгом смысле, если невозможно размыкание ИПС (появление любого субъекта, не входящего в состав ИПС).

Следствие. В ИПС, содержащей СУ без возможности изменения ОУ МБС, выполнены условия корректного управления в строгом смысле.

Вполне очевидно, что в реальных КС практически невозможно поддержать условия корректного управления в строгом смысле, поскольку постоянно требуется дополнять субъекты к ИПС. С другой стороны, дополнение некорректного субъекта разрушает ИПС и противоречит задаче поддержания гарантий заданной ПБ (поскольку требуется дополнять только субъект, корректный относительно существующих в ИПС). В связи с этим в подтверждение тезиса о том, что достаточные условия ИПС и корректного управления в строгом смысле совпадают, сформулируем еще одно следствие.

Следствие. При корректном управлении дополнение корректного субъекта не нарушает ИПС.

Итак, сформулированы основные условия поддержания ИПС в процессе управления. Обратимся теперь к необходимым дополнениям политики безопасности для различных конфигураций управляемой КС. При этом будем рассматривать схему с разделением на локальный и внешний сегменты КС.

Введем понятие технологии управления.

Определение 4. Технология управления защитой в КС – порядок формирования ОУ и применения управляющего субъекта в КС, находящийся в рамках понятий корректного управления и зависящий от локализации управляющего субъекта и объектов управления в КС и методики формирования исходной информации для его работы.

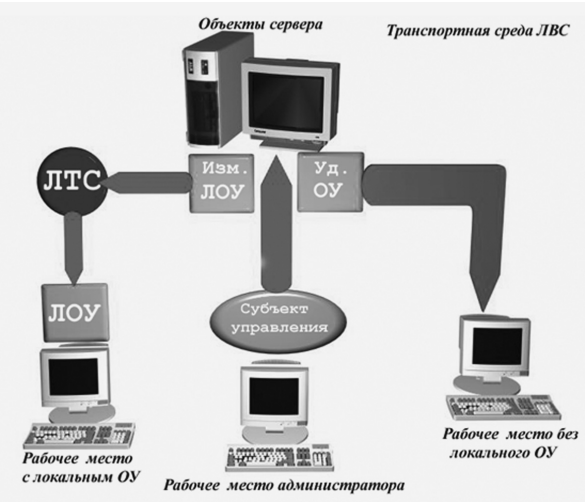


Рис. 1. Локализация субъекта и объектов управления в распределенной КС

Очевидно, что субъект управления должен в общем случае иметь доступ к объектам КС на чтение (это достаточно и для вычисления функций КЦ в части управления ОУ для МБС). В случае преобразования со-

держимого объектов доступ необходим также и на запись. Рассмотрим разделение КС на локальный сегмент (ЛС КС) и внешний сегмент (ниже используются термины «удаленные субъекты и объекты», соответствующие внешним субъектам и объектам).

Поскольку ОУ может быть локальным (принадлежащим ЛС КС) или удаленным и субъект управления также может быть локализован либо в ЛС КС, либо во внешнем сегменте КС (***рис. 1***), то различные конфигурации локализации объекта управления и управляющего субъекта можно классифицировать следующей таблицей (легко заметить, что иные локализации ОУ и УС невозможны – таблица исчерпывает всевозможные технологии управления, зависящие от локализации ОУ и УС).

Таблица 1. (локализация управляющего субъекта и объекта управления)

	Объект управления		**Управляющий субъект**		**Примечание**
	Локальный	Удаленный	Локальный	Удаленный	
0	-	-	-	-	КС не управляема
1	-	-	-	+	КС до момента установки защиты
2	-	-	+	-	КС до момента установки защиты
3	-	-	+	+	КС до момента установки защиты
4	-	+	-	-	КС неуправляема, либо режим без изменения ОУ
5	-	+	-	+	Локальное место не имеет объекта управления, удаленное управление
6	-	+	+	-	Локальное управление, ОУ формируется как удаленный
7	-	+	+	+	Удаленный объект управления, смешанное управление
8	+	-	-	-	Локальный ОУ, система в стационарной фазе (ОУ не меняется)

9	+	-	-	+	Локальный ОУ, удаленное управление
10	+	-	+	-	Локальный ОУ и локальное управление (классический случай изолированного рабочего места (ПЭВМ))
11	+	-	+	+	Локальный ОУ с возможностью как удаленного, так и локального управления
12	+	+	-	-	Локальный и удаленный ОУ, система в стационарной фазе
13	+	+	-	+	Локальный и удаленный ОУ, управление чисто удаленное
14	+	+	+	-	Локальный и удаленный ОУ, управление чисто локальное
15	+	+	+	+	Локальный и удаленный ОУ, управление смешанное

Рассмотрим подробнее описанные в таблице конфигурации.

При рассмотрении будем руководствоваться положением о том, что физический доступ администратора к ЛС КС нерационален, управление защитой должно быть централизовано и, при необходимости, проводиться из внешнего сегмента КС.

Конфигурация 0 не рассматривается, поскольку не существует ни локального, ни удаленного субъекта управления, следовательно, КС не управляема (по определению управляемости).

Конфигурации 4, 8, 12 описывают КС в некоторой стационарной относительно управления фазе (когда ОУ в соответствующих конфигурациях сформированы, но субъект управления в КС отсутствует).

Конфигурации 1 и 2 описывают начальный момент активности субъекта управления (удаленного и локального соответственно), при котором объекты управления еще не сформированы. Конфигурация 3 является функциональным объединением конфигураций 1 и 2.

Конфигурация 5 описывает технологию управления, при которой как объект управления, так и управляющий субъект находятся во внешнем сегменте КС. Данная конфигурация соответствует распределенной КС, в которой управление сосредоточено на месте (местах) администратора, а объект управления находится, как правило, на общем ресурсе хранения (например, файл-сервере).

Конфигурация 6 описывает технологию управления, при которой локальный управляющий субъект создает удаленный объект управления. Практически это соответствует работе администратора на своем рабочем месте, при которой создаются объекты управления во внешнем сегменте. Будем рассматривать данную конфигурацию как разновидность конфигурации 5 (поскольку ЛС КС выделен произвольно).

Конфигурация 7 соответствует возможности изменения удаленного ОУ (т.е. находящегося во внешнем сегменте КС) как со стороны удаленного субъекта, так и субъектом ЛС КС. Далее данная конфигурация рассматриваться не будет, поскольку не имеет содержательного смысла (не имеет смысла изменять объект управления с локального рабочего места, если он доступен с удаленного места администратора).

Конфигурация 9 описывает изменение локального ОУ при помощи субъекта внешней КС. Данная конфигурация весьма интересна, поскольку подразумевает такую организацию взаимодействия локального и внешнего сегмента КС, при котором некоторым субъектам доступны ресурсы управляемых рабочих мест (такова, например, конфигурация одноранговой сети с разделяемыми (SHARED) объектами коллективного доступа).

Конфигурация 10 не представляет интереса, поскольку описывает классический случай изменения локального ОУ локальным субъектом управления.

Конфигурация 11 является функциональным объединением конфигураций 9 и 10.

Конфигурации 13-15 рассматривают либо распределенный объект управления, либо разделенный ОУ (в этом случае необходимо рассматривать объединение рассмотренных конфигураций).

Ниже последовательно детализируем различные варианты технологий управления и рассмотрим их особенности.

Будем учитывать также возможность существования в КС таких ЛС КС, для которых не установлены никакие средства защиты и потенциально доступны удаленные объекты управления.

5.3. Система удаленного управления безопасностью в отсутствии локального объекта управления

Данная конфигурация соответствует конфигурации 5 таблицы 1. В данном случае важно, что ЛС КС не имеет объекта управления, объект управления принадлежит внешнему сегменту КС и создается субъектом внешнего сегмента.

Технически удобно реализовать работу субъекта управления в рамках другого ЛС КС (рабочего места администратора); субъект управле-

ния будет создавать ОУ где-либо во внешнем сегменте, доступном субъектам локальной КС.

Утверждение 3 (необходимое условие 1 для создания системы корректного управления)

Необходимым условием для выполнения МБО заданной политики безопасности в рамках ЛС КС при наличии локальных объектов и выделении во множестве потоков непустого подмножества легальных потоков к локальным объектам является наличие в удаленном ОУ элементов, описывающих потоки между субъектами и локальными объектами.

Доказательство

Первоначально определим область применения утверждения. Если все множество потоков к локальным объектам является либо разрешенным, либо запрещенным (либо такие объекты отсутствуют), то ОУ может не включать потоков к локальным объектам, поскольку для любого локального объекта делается однозначный вывод о доступе к нему.

Если же есть хотя бы один локальный объект, к которому разрешен или запрещен доступ, то данный поток должен быть описан в ОУ.

От противного. Пусть МБО выполняет заданную ПБ. Для определенности предположим, что некоторый поток $Stream(S_i, O_k)->O_m$ описан как нелегальный, тогда если в ОУ данный поток не указан, то МБО не сможет запретить данный поток в случае его возникновения, а, следовательно, и реализовать ПБ. Противоречие.

Несмотря на то, что данное утверждение является достаточно очевидным, оно, тем не менее, описывает важную техническую проблему, возникающую при управлении МБО и МБС. Проблема заключается в необходимости доступа к объектам ЛС КС при формировании удаленного ОУ со стороны удаленного управляющего субъекта.

Как видно из условий сформулированного выше утверждения, можно выделить две основные разновидности управления при существовании удаленного ОУ – управление при пустом множестве либо L, либо N и управление при непустом множестве L и N для ЛС КС.

В первом случае администратор при помощи субъекта управления изменяет удаленный ОУ, который включает только потоки между удаленными субъектами и удаленными объектами. Как правило, данная конфигурация реализуется в сетях с одним или несколькими серверами (т.е. в случае наличия ресурса общего пользования, где и находится ОУ).

Во втором случае управляющий субъект должен иметь доступ к объектам ЛС КС. Рассмотрим два способа организации доступа к объектам ЛС КС для внешнего субъекта.

1. Постоянный доступ к объектам ЛС КС при помощи локального субъекта (субъект обеспечения постоянного доступа).

2. Транспортировка информации об объектах ЛС КС к какому-либо удаленному объекту во время стартового периода работы пользователя при помощи локального субъекта.

Две конфигурации отличаются только интервалом времени активности локального субъекта, при помощи которого осуществляется доступ УС.

Способ постоянного доступа к объектам ЛС КС требует активности некоторого локального субъекта, управляемого удаленным субъектом администрирования. Основной проблемой в данном случае является возможность управления локальным субъектом со стороны злоумышленника. Вполне очевидно, что для включения в ОУ потоков между любым объектом и субъектом локальный субъект (управляемый удаленным субъектом администрирования) должен иметь доступ ко всем объектам ЛС КС. В этом случае (см. главу 7) субъект внешнего злоумышленника будет иметь такой же доступ также ко всем объектам ЛС КС. Канал управления в данном случае является и каналом НСД.

Уменьшить или блокировать возможности удаленного злоумышленника (в частности, полностью исключить возможности изменения объектов ЛС КС (доступ на запись)) возможно, если субъект постоянного доступа имеет доступ только на чтение к объектам ЛС КС.

Для обеспечения корректного управления необходимо отсутствие злоумышленных субъектов во всей КС. Достаточным условием для этого является существование ИПС на каждом ЛС КС и наличие МБО, запрещающее доступ к объектам ЛС КС любого субъекта, кроме управляющего. Докажем это.

Утверждение 4 (необходимое условие 2 для создания системы корректного управления)

Пусть в КС выделяется конечное множество ЛС КС и объединение всех субъектов и объектов ЛС КС составляет все множество субъектов и объектов КС. В случае существования локального субъекта постоянного доступа в любой ЛС КС и наличии управляющего удаленного субъекта достаточным условием корректного управления является существование ИПС с контролем целостности объектов источников в рамках каждого ЛС КС и существование МБО в рамках каждой ЛС КС, запрещающего доступ любого локального субъекта к удаленному ОУ (кроме субъекта управления).

Доказательство

Выберем произвольный ЛС КС. Возможны два случая – в составе субъектов ЛС КС есть субъект управления, либо субъект управления отсутствует.

Первый случай.

В каждом ЛС КС (также и в рассматриваемом) генерируется ИПС с контролем неизменности объектов источников, следовательно, реализуются только потоки, разрешенные МБО, по условию утверждения доступ к ОУ невозможен.

Второй случай. В ИПС возможны только потоки, разрешенные МБО, аналогично, по условию утверждения доступ разрешен только субъекту управления.

Утверждение доказано.

Данное утверждение структурно конкретизирует предыдущее утверждение, описывая условия, при которых корректно управление в КС, разделяемой на несколько ЛС КС.

На практике выполнение условий данного утверждения возможно только в топологически замкнутой КС (например, ЛВС, которая построе-

на как топологически замкнутая сеть – т.е. подключение дополнительных рабочих станций без защитных субъектов (МБО, МБС) невозможно).

Как отмечалось выше, доступ на чтение к объектам ЛС КС при наличии субъекта полного доступа в ряде случаев нежелателен. Исходя из этого, требуется уточнить ПБ в части запрета доступа к локальным объектам одного ЛС КС со стороны субъектов другого ЛС КС.

Легко видеть, что предыдущее утверждение в этом случае дополняется реализацией в локальном МБО правила запрета потоков между локальным субъектом и удаленным объектом, принадлежащим другому ЛС КС локальному объекту (к части объектов потоки могут быть разрешены). Информация о разрешенных потоках может храниться в удаленном ОУ.

Необходимо остановиться на работе пользователя-злоумышленника. Особенность в данном случае состоит в том, что пользователь пытается реализовать потоки, не описанные в ОУ (для множества L). В случае существования удаленного объекта управления он может осуществить данную цель следующим образом.

Поскольку удаленный ОУ описывается некоторым положением в общедоступном удаленном ресурсе хранения (путем), а путь к ОУ для локального МБО является ассоциированным объектом, то возможны действия по корректировке данного пути таким образом, чтобы он указывал на локальный объект управления. Необходимо отметить также, что методы логического преобразования ОУ (с целью его предназначения данному пользователю – например, дополнение электронной цифровой подписи к ОУ на личном ключе администратора и проверка на его открытом ключе) решают проблему лишь частично, поскольку субъект МБО, принадлежащий ЛС КС, может быть изменен с редуцированием данных проверок. Следовательно, ИПС в рамках каждого ЛС КС в данном случае является не только достаточным, но и необходимым условием корректного управления.

5.4. Система управления безопасностью при локальном объекте управления и при удаленном управляющем субъекте

Рассматриваемая конфигурация является некой «смысловой инверсией» ранее рассмотренной. Для нее можно сформулировать аналогичные утверждения, связанные со свойствами ОУ.

Утверждение 5

Необходимым условием для выполнения МБО заданной политики безопасности в рамках ЛС КС при наличии локальных объектов и выделении во множестве потоков непустого подмножества легальных потоков к удаленным объектам является наличие в локальном ОУ элементов, описывающих потоки между субъектами и удаленными объектами.

Доказательство данного утверждения аналогично доказательству предыдущего утверждения.

В данной конфигурации управления также возникает проблема доступа к объектам ЛС КС при формирования локального ОУ со стороны

удаленного управляющего субъекта, причем в данном случае кроме доступа к локальным объектам на чтение необходим и доступ на запись для коррекции локального ОУ. Следовательно, данная конфигурация является более сложной с точки зрения обеспечения корректного управления.

Как видно из условий сформулированного утверждения, можно, как и ранее, выделить две основные разновидности управления при существовании локального ОУ – управление при пустом множестве либо L, либо N и управление при непустом множестве L и N для удаленного сегмента КС.

В первом случае администратор при помощи субъекта управления изменяет (создает) локальный ОУ, который включает только потоки между локальными субъектами и локальными объектами.

Во втором случае управляющий субъект должен иметь доступ к объектам ЛС КС. Аналогично предыдущему случаю рассмотрим два способа организации доступа к объектам ЛС КС для внешнего субъекта.

1. Постоянный доступ к объектам ЛС КС при помощи локального субъекта.

Особенностью данной конфигурации в случае локального ОУ является необходимость доступа на чтение ко всем объектам ЛС КС и на запись – к ОУ.

2. Транспортировка информации об объектах ЛС КС к какому-либо удаленному объекту во время стартового периода работы пользователя при помощи локального субъекта.

Способ постоянного доступа к объектам ЛС КС требует активности некоторого локального субъекта, управляемого удаленным субъектом администрирования. Основной проблемой в данном случае, как уже отмечалось, является возможность управления локальным субъектом со стороны злоумышленника.

Уменьшить возможности удаленного злоумышленника в части НСД к объектам управления (в частности, полностью исключить возможности изменения объектов ЛС КС (доступ на запись)) возможно, если субъект постоянного доступа имеет доступ только на чтение к объектам ЛС КС, а режим записи в ОУ включается только для некоторых интервалов времени после опознания управляющего субъекта.

Для обеспечения корректного управления необходимо отсутствие злоумышленных субъектов во всей КС. Достаточным условием для этого является существование ИПС на каждом ЛС КС и наличие МБО, запрещающее доступ к объектам ЛС КС любого субъекта, кроме управляющего. Этот факт был сформулирован и доказан выше.

Данное утверждение структурно конкретизирует условия, при которых корректно управление в КС, разделяемой на несколько ЛС КС.

На практике выполнение условий данного утверждения также возможно лишь в топологически замкнутой КС.

Как отмечалось выше, доступ на чтение к объектам ЛС КС при наличии субъекта полного доступа в ряде случаев нежелателен. Требуется уточнить ПБ в части запрета доступа к локальным объектам одного ЛС КС со стороны субъектов другого ЛС КС.

Легко видеть, что предыдущее утверждение в этом случае дополняется реализацией в локальном МБО правила запрета потоков между локальным субъектом и удаленным объектом, принадлежащим другому ЛС КС локальному объекту (к части объектов потоки могут быть разрешены). Информация о разрешенных потоках хранится в локальном ОУ.

5.5. Метод «мягкого администрирования». Автоматизированное формирование списков разрешенных задач и правил разграничения доступа

В развитых программных комплексах весьма сложным является вопрос формирования ОУ для МБС. Сложность в данном случае связана с тем, что последовательность активизации субъектов какого-либо программного комплекса не является информацией, доступной пользователю. Например, текстовый процессор типа Microsoft Word 6.0 инициирует несколько десятков субъектов с объектами-источниками типа .DLL (динамические библиотеки). При замыкании программного комплекса в ИПС отсутствие в списке разрешенных для порождения субъектов объектов-источников какой-либо библиотеки делает невозможным доступ к реализуемой в библиотеке функции. Это в свою очередь означает снижение пользовательских свойств КС и в конечном итоге ухудшение такой характеристики КС, как доступность некоторого ресурса.

Таким образом, налицо достаточно сложная проблема управления МБС. Ранее для разрешения возникающих проблем в сложных программных пакетах в список разрешенных объектов-источников вносились все файлы типа «исполняемый» (включая библиотеки, драйверы и т.д.). Такой подход порождал некоторую функциональную избыточность относительно защиты.

Часто в сложных программных средах существуют возможности типа инструментальных или отладочных, которые не предназначены для пользователя, но либо используются при конфигурировании и наладке среды, либо недокументированны. Избыточность при составлении ОУ для МБС автоматически включает все указанные инструментальные возможности внутрь сформированной ИПС. Выше указывалось, что инструментальные возможности потенциально могут нарушать условия функционирования ИПС (нарушать корректность межсубъектного взаимодействия, либо инициировать потоки к объектам с иным (более низким) уровнем представления).

Предлагается метод формирования ОУ для МБС, который условно можно назвать методом мягкого администрирования. Суть метода заключается в следующем.

До установки защитных модулей в КС (имеется в виду МБО и МБС) в программную среду устанавливается субъект, который обладает следующими свойствами: отслеживает все факты порождения субъектов с фиксацией (как минимум) объектов-источников и протоколирует (записывает) их в некоторый объект (исходный список мягкого администрирования). Через некоторый промежуток времени содержимое объекта изучается администратором, который проделывает ряд действий.

1. Проводит сортировку имен объектов-источников (с целью удаления повторяющихся имен объектов-источников).

2. Проверяет по указанным именам объектов фактическое наличие их в КС.

3. Редуцирует имена несуществующих объектов.

4. Исключает из списка объекты-источники, порождающие субъекты со свойством заведомой некорректности (инструментальные, отладочные средства и т.д.).

После выполнения пп. 1-4 администратор получает список объектов-источников, который он может использовать для формирования ОУ МБС. Однако необходимо заметить, что в случае наличия среди пользователей КС пользователя со злоумышленными намерениями, осведомленного о применении метода мягкого администрирования, в процедуре управления возможны некорректные действия. Данные действия связаны с маскировкой объекта-источника для порождения некорректного субъекта под объект с иным именем. В данном случае регистрация такого субъекта в ОУ МБС приведет к невыполнению условий изолированности среды. Кроме того, пользователь-злоумышленник может подменить некий файл своим в промежуток времени от снятия списка мягкого администрирования до установки защиты (формирования ОУ МБС).

Указанного выше недостатка лишен модифицированный метод мягкого администрирования. Условием применения данного метода будет наличие эталонного перечня объектов-источников для некоторого программного пакета. Итак, пусть имеется программный пакет, состоящий из m исполняемых файлов (объектов-источников) $P=\{F_1, ..., F_m\}$. Администратор вычисляет хеш-функции $h_i=H(F_i)$ и хранит объект эталонов $Oe=\{h1, ..., hm\}$. Пусть в списке-источнике мягкого администрирования присутствует некоторое подмножество R объектов множества P. Тогда при регистрации объекта-источника F_i (F_i принадлежит R) в ОУ МБС производятся следующие действия:

1. Проверяется принадлежность F_i к P. В случае положительного исхода проверки – переход к п. 2, иначе объект игнорируется.

2. Вычисляется $gi=H(F_i)$ и проверяется совпадение h_i и gi. При несовпадении объект игнорируется, иначе к п. 3.

3. Формируется список объектов LIST для пополнения ОУ МБС.

Покажем, что при таком методе мягкого администрирования в рамках множества субъектов, порождаемых из объектов источников P, генерируется ИПС. Предположим, что до момента времени t_0 в КС существует ИПС, причем все субъекты порождены с контролем неизменности из объектов некоторого множества T (содержательный смысл данного множества – описание объектов-источников субъектов, которые были порождены до момента порождения субъектов пакета P). В момент t_0 ОУ МБС пополняется объектами списка LIST.

Утверждение 6 (лемма для обоснования метода мягкого администрирования)

Пусть в момент времени t_0 в КС действует ИПС с контролем неизменности объектов-источников и все множество разрешенных для по-

рождения субъектов объектов-источников составляет объединение P и T. Тогда в любой момент $t>t_0$ в КС также сохраняется ИПС при условии попарной корректности субъектов, могущих быть порожденными из P и T.

Доказательство

Предположим для простоты, что вывод о неизменности объекта-источника безошибочен (поскольку используется неоднозначное отображение, то это не так, в таком случае вероятность разрушения ИПС совпадает с вероятностью принять нетождественный объект-источник за тождественный). В этом случае по построению процедуры составления списка LIST в нем зафиксированы только неизменные относительно эталонов объекты-источники. По условию все файлы из пакета P попарно корректны между собой и субъектами, могущими быть порожденными из T. Следовательно, подмножество R также содержит только попарно корректные субъекты, поскольку в КС действует контроль порождения субъектов с контролем неизменности объектов-источников, исходя из утверждений части 1 (о сохранении ИПС при дополнении множества субъектов попарно корректными), получим, что при $t>t_0$ ИПС сохраняется.

Утверждение доказано.

Смысл данного утверждения состоит в описании процедуры мягкого администрирования, которая сохраняет ИПС при дополнении множества объектов-источников. С другой стороны, описанные в списке LIST объекты обеспечивают исполнение пользовательских свойства программного пакета P. Далее, процедура исключения объектов-источников из ОУ МБС, как было доказано выше, не приводит к нарушению изолированности программной среды.

Итак, метод мягкого администрирования состоит в фиксации на этапе опытной эксплуатации всех или избранных аргументов операций ***Create*** (исходный материал для составления множества E) и ***Stream*** (для составления ПРД). Ценность метода состоит в том, что администратор безопасности выполняет только операцию редуцирования (сокращения) списков и сравнение функций целостности объектов с эталонными. Возможность применения мягкого администрирования целесообразно предусмотреть еще на этапе проектирования. Применение метода мягкого администрирования позволяет говорить о процессе автоматизации составления ПРД, что является новым результатом.

5.6. Системы управления безопасностью при распределенном объекте управления

Рассмотрим различные представления объекта управления (для простоты будем рассматривать только ОУ МБС). Рассмотрим ОУ как единый объект или как распределенный объект в КС. Рассмотрим пример. Пусть с каждым объектом-источником O_j ассоциирован некоторый объект Ouj (с точки зрения МБС, т.е. при выполнении Create(S_i,O_j) МБС обращается к Ouj). Данный объект содержит информацию о множестве

разрешенных для порождения субъектов объектов-источников E_i для пользователя i. Предположим, что Ouj в свою очередь состоит из последовательности подобъектов (неразделяемых) Oumj. Каждый из этих подобъектов описывает информацию о пользователе i, который может инициировать порождение субъекта из данного объекта (IDi) и функцию контроля целостности h_j (вычисленную на основе индивидуальной информации пользователя Ri, $h_j=H(K_i,O_j)$). Данная информация является минимально достаточной для порождения ИПС.

В самом деле, для работы МБС необходимо определить для пользователя i включенность некоторого субъекта во множество E_i и проверить неизменность объекта-источника. Первое определяется по IDi, второе – по h_j.

Рассмотрим параметр Pri – вероятность запуска случайно равновероятно выбранного объекта O_j*, не совпадающего с O_j в ИПС с КЦ для пользователя i. Причем, зададимся еще двумя априорно известными вероятностями: Pi – вероятность принять пользователя m, не равного i, за пользователя i (по IDi) и P_h – вероятность совпадения хеш-значений для двух случайно равновероятно заданных объектов-источников. Поскольку события «ложная аутентификация пользователя i» (вероятность Pi) и «совпадение хеш-значений объектов» (вероятность P_h) независимы, то вероятность Pri вычисляется как произведение вероятностей соответствующих событий: $Pri = Pi*P_h$.

Очевидно, что задана верхняя оценка вероятности, поскольку случайный равновероятный файл-источник может вообще не породить субъекта, либо порожденный субъект сохраняет свойства корректности.

Объект управления, спроектированный по указанной выше схеме, обеспечивает при каждой попытке активизации субъекта дополнительную операцию авторизации (подтверждения прав) пользователя. За счет этого интегральная оценка надежности защиты повышается, а вероятность Pir существенно снижается.

Рассмотрим преимущества распределенного ОУ с точки зрения технологии управления.

При использовании распределенного ОУ минимизируются затраты памяти для реализации МБС. Наконец, при различном отображении имен и путей объектов (mapping) положение ОУ не меняется (поскольку он находится по тому же пути). Распределенный ОУ соответствует конфигурациям 14 и 15, поскольку для удаленных объектов и подобъекты будут удаленными. Тем самым достигается свойство топологической однородности.

Обратимся теперь к понятию распределенного управления.

Рассмотрим процедуру управления в случае, когда ОУ является объектом произвольной локализации (локальным, удаленным или распределенным) и выделяются два субъекта управления – локальный субъект управления (ЛСУ) и удаленный субъект управления (УСУ). В качестве задачи будем рассматривать задачу разработки такого метода управления (для простоты будем рассматривать систему управления МБС), при котором:

• администратор управляет защитой со своего рабочего места, которое относительно пользователя является внешним сегментом КС,

• администратор не имеет физического доступа к ЛС КС.

Исходные положения следующие:

• в рамках ЛС КС действует ИПС с контролем целостности объектов-источников,

• ЛСУ входит для каждого пользователя в состав разрешенных субъектов.

Необходимо заметить, что во время сеанса работы пользователя в рамках ЛС КС ЛСУ обладает всей информацией пользователя, которая необходима для изменения ОУ МБС. Для администратора целесообразно организовать канал управления ЛСУ и доступность удаленного ОУ МБС (в случае его использования). Тогда возможна коррекция прав пользователя (сводящаяся к коррекции ОУ МБС) во время начала работы пользователя в системе.

Выше упоминалась задача обеспечения доступа администратора к ресурсам ЛС КС. В данном случае доступ осуществляется через ЛСУ и технически легко реализуем.

Сформулируем алгоритм управления.

1. Перенос информации о локальных объектах-источниках в область, доступную администратору защиты. Данная операция выполняется ЛСУ и может быть сопряжена с мягким администрированием. Предположим, что после работы субъектов, производящих протоколирование используемых объектов источников создался объект $LIST_i$, который содержит полные имена локальных (и удаленных) объектов-источников и, при необходимости их хеш-значения (см. описанный выше алгоритм мягкого администрирования) для i-го пользователя. Тогда инициируется ***Stream(ЛСУ, LIST$_i$)->LIST$_i$*.*** Объекты $LIST_i$ и $LIST_i$* тождественны, но имеют различную локализацию (первый принадлежит ЛС КС, второй – внешнему сегменту, доступному администратору). Следовательно, после выполнения п.1 администратору доступна информация о локальных объектах.

2. Проверка целостности объектов-источников ЛС КС путем вычисления такой же хеш-функции. Допускается зависимость хеш-функции от K_i, но в таком случае администратор должен располагать K_i для всех пользователей. При несовпадении хеш-значения, вычисленного администратором, с тем, которое хранится в $LIST_i$, объект-источник редуцируется из списка.

3. После выполнения процедуры п.2 образуется объект $SOURCE_i$, содержащий имена объектов-источников для i-го пользователя. Данный объект должен обладать следующим свойством – быть доступным на запись только для УСУ.

4. Начало сеанса работы пользователя. Активизация ЛСУ, чтение объекта $SOURCE_i$, коррекция локальных и удаленных ОУ МБС.

В случае пользователя-злоумышленника угроза корректному управлению исходит от ЛСУ, который имеет доступ на запись к ОУ МБС. Естественным требованием является отсутствие возможностей внешнего

управления. С другой стороны, либо в ассоциированном объекте ЛСУ, либо во внешнем объекте должна содержаться информация локализации объекта $SOURCE_i$ (путь). В случае локализации внутри объекта-источника ЛСУ путь к объекту $SOURCE_i$ защищен от изменений процедурой контроля неизменности объекта-источника ЛСУ. В случае локализации пути во внешнем объекте необходима коррекция ОУ МБО так, чтобы исключить возможность изменения данного объекта.

Докажем корректность управления в строгом смысле при условии ИПС в рамках ЛС КС и неизменности пути к объекту $SOURCE_i$.

Утверждение 7 (условия корректности управления при мягком администрировании).

Пусть в рамках ЛС КС действует ИПС с контролем неизменности объектов-источников, в которую включен также ЛСУ, последовательность активизации компонент Z_L неизменна, путь к объекту $SOURCE_i$ и его содержание также неизменно, тогда для i-го пользователя управление корректно в строгом смысле.

Доказательство

По условию утверждения ОУ изменяется только ЛСУ, который не имеет возможности внешнего управления со стороны пользователя i. С другой стороны, ЛСУ достоверно воспримет объект $SOURCE_i$, поскольку при порождении ЛСУ из объекта-источника произведен контроль неизменности его, а объект $SOURCE_i$ недоступен для изменений и невозможно имитировать другой объект за счет изменения пути (по условию утверждения). Следовательно, ОУ будет изменен в соответствии с содержимым $SOURCE_i$, который сформирован администратором. Далее, пользователь не может изменить последовательность активизации Z_L и инициирование ЛСУ будет предопределенным, после активизации МБС ОУ недоступны пользователю для изменения. Утверждение доказано.

Управление безопасностью относится к ключевым моментам гарантирования политики безопасности в течение всего времени работы защищенной КС. Кроме вопросов гарантирования управления без нарушения ПБ важными являются эксплуатационные свойства управляющих комплексов.

6. МОДЕЛИ СЕТЕВЫХ СРЕД. СОЗДАНИЕ МЕХАНИЗМОВ БЕЗОПАСНОСТИ В РАСПРЕДЕЛЕННОЙ КОМПЬЮТЕРНОЙ СИСТЕМЕ

6.1. Введение

Ниже будет рассмотрен «внешний» аспект защищенности информации в терминах описания модели КС, сформулированных выше, относительно локального сегмента КС при злоумышленном воздействии субъектов внешней среды (например, сети Internet).

Будем полагать, что пользователь локального сегмента КС использует ресурсы внешнего сегмента. Использование внешних ресурсов подразумевает существование потоков информации от объектов внешнего сегмента КС к ассоциированным объектам (в общем случае как функционально ассоциированным, так и к ассоциированным объектам-данным) субъектов локального сегмента и наоборот (причем данные потоки инициируются субъектами локального сегмента при управлении этими субъектами со стороны локальных пользователей), а также потоков от ассоциированных объектов субъектов внешнего сегмента к объектам локального сегмента (к которым могут относиться и потоки инициируемые субъектами, управляемыми злоумышленником).

В данном случае можно говорить о внешнем злоумышленном воздействии. При этом необходимо учесть, что в подавляющем большинстве случаев локальный сегмент КС технически реализован на одном или нескольких связанных сегментах ЛВС и используют общее стандартизированное программное обеспечение в части коммуникации. Стандартной является и операционная среда (среды) всей ЛВС.

Исходя из указанных положений уточним положения модели воздействия на КС извне следующим образом. В едином пространстве объектов рассматриваемого локального сегмента КС (который соответствует, как правило, корпоративной сети организации) действует один или несколько субъектов (телекоммуникационных программных модулей) с возможностью внешнего управления. Внешнее управление реализуется возможностями телекоммуникационной программы по двунаправленной передаче данных по командам, исходящим от субъекта, принадлежащего внешнему сегменту КС. Под командами в данном случае понимается поток от ассоциированных объектов внешнего субъекта к ассоциированным объектам локального телекоммуникационного субъекта, причем указанные ассоциированные объекты, измененные под воздействием потока от внешнего субъекта, существенно влияют на текущее состояние локального субъекта. При этом модули телекоммуникационного ПО, расположенные в локальном сегменте, имеют доступ к объектам этого сегмента как прочие локальные субъекты.

В чем состоит отличие от рассмотренных ранее положений реализации и гарантирования ПБ в КС? Отличие заключается в том, что факти-

чески частью локальных субъектов управляет легальный пользователь, а частью – удаленный анонимный пользователь, который потенциально преследует злоумышленные цели.

Выделим пассивное воздействие, исходящее от субъектов внешней сети (чтение и транспортирование объектов локального сегмента во внешнюю сеть), и активное воздействие (модификация локальных объектов). Пассивное воздействие связано с нарушением конфиденциальности корпоративной информации (компрометация), активное – с нарушением целостности.

Во множестве объектов можно выделить более ценные для злоумышленника – объекты, имеющие отношение к обеспечению локальной безопасности (например, для UNIX-систем файл для аутентификации пользователей etc\passwd).

Активное воздействие может проявляться опосредованно, т.е. в принятых из внешней сети программных модулях могут находиться РПВ, либо специально интегрированные возможности злоумышленных действий.

Отдельно выделим распространение разрушающих программных воздействий в интерпретируемых данных (активное воздействие). Сущность злоумышленного воздействия состоит в интеграции внутрь интерпретируемого объекта (обрабатываемого редакторами типа Word) операций, направленных на реплицирование (распространение) РПВ, либо на разрушение локальных данных. С другой стороны, можно рассмотреть и порождение субъекта в рамках локального сегмента КС, но инициированного внешним субъектом и из переданного извне объекта-источника. Программно-технически это механизмы удаленного запуска (REXEC) или удаленные вызовы процедур (Remote Procedure Call, RPC).

При проектировании защитных механизмов для локального сегмента КС представляется возможным двигаться по двум практически независимым направлениям – использовать локальные механизмы защиты (возможно каким-либо образом скорректированные по отношению к описанной специфике угрозы) и синтезировать специализированные механизмы для защиты преимущественно от внешних угроз. Именно такого подхода предлагается придерживаться ниже, выделяя методы проектирования локальных механизмов защиты, устойчивых к воздействию извне, и подход сетевой защиты, ориентированный на работу с интегральным потоком информации от внешней сети (внешнего сегмента КС).

В зарубежной литературе преобладает подход общей защиты локального сегмента (сформулированный вторым). Он характеризуется перенесением ответственности за защиту в локальном сегменте КС на уровень сетевого взаимодействия и реализуется технологией межсетевого экранирования (брандмауэр). Межсетевое экранирование – основной подход, предлагаемый в зарубежных системах, сопряженных с Internet. При этом анализируется поток информации сетевого уровня (Network), который обладает уникальной информацией, характеризую-

щей отправителя и получателя пакета (адресом). Работа межсетевого экрана сводится к анализу последовательности пакетов (фильтрации) по некоторым априорно заданным критериям (при этом необходимо обратить внимание на необходимости двунаправленной фильтрации – для входящих и для исходящих пакетов).

В литературе описаны два основных подхода к процедуре фильтрации: пакетная фильтрация (или трансляция адресов) (анализируется и/или преобразуется адресная часть пакетов) и фильтры прикладного уровня (ФПУ) (в зарубежной литературе proxy service или application proxy). ФПУ анализирует содержательную часть пакетов, исходя из особенностей работы прикладных телекоммуникационных программ (приложений), которые порождают поток пакетов (обычно ФПУ ориентирован на распространенные приложения типа FTP или Telnet). Для конкретной программы принципиально возможно по последовательности пакетов установить, к какому объекту обращается то или иное приложение (субъект). Однако в общем случае проблема установления факта доступа к объекту высокого уровня по информации низкого уровня иерархии для произвольного порождающего поток субъекта является практически неразрешимой. При этом ФПУ, будучи с точки зрения надежности фильтрации более надежным методом, тем не менее проигрывают в совместимости с приложениями (в смысле необеспечения корректной работы всех используемых в КС типов телекоммуникационных субъектов).

Описанные технические решения межсетевых экранов практически не поддаются осмыслению с точки зрения надежностных характеристик защиты, т.е. не удается сделать ни детерминированных, ни вероятностных выводов о качестве фильтра. В связи с этим ниже вводится ряд уточняющих определений, описывающих процесс фильтрации с учетом логического уровня представления объектов и доказывается ряд утверждений относительно гарантий работы экрана в рамках введенных понятий.

Предлагается рассматривать ситуацию, при которой на рабочих местах корпоративной ЛВС уже установлены средства защиты. Общепринятое исполнение программно-технических средств защиты предполагает реализацию политики безопасности с полным проецированием прав пользователя на права любого субъекта локального сегмента (т.е. если пользователю разрешен запуск какой-либо программы (т.е. произошла активизация субъекта), то порожденный процесс обращается к объектам с правами инициировавшего его пользователя). Ниже будет показано, что такой подход некорректен для защиты при воздействии извне, поскольку доступ злоумышленника к локальным объектам через управляемый им телекоммуникационный модуль на локальном месте пользователя будет происходить с правами локального пользователя. Следовательно, политика безопасности при использовании локальных механизмов защиты нуждается в некоторой конструктивной коррекции.

6.2. Модели воздействия внешнего злоумышленника на локальный сегмент компьютерной системы

Определим использованные выше понятия локального и внешнего сегмента КС.

Определение 1. Локальный сегмент КС (ЛС КС) – подмножество субъектов и объектов КС, выделяемое по одному из следующих критериев:

• критерию группирования в одно множество всех субъектов с возможностью непосредственного управления субъектами (если такая возможность присутствует в субъекте),

• критерию локализации некоторого подмножества объектов и субъектов в рамках некоторой технической компоненты КС,

• критерию присвоения объектам и субъектам ЛС КС некоторой информации, однозначно характеризующей субъект или объект (которая, как правило, называется адресом или сетевым адресом ЛС КС).

Определение 2. Внешний сегмент КС – дополнение множества субъектов и объектов локального сегмента до всего множества объектов КС.

Очевидно, что во внешнем сегменте могут быть выделены несколько локальных. Ниже будем полагать, что рассматривается один произвольно выделенный ЛС КС.

Непосредственное управление (рассматриваемое как один из критериев выделения ЛС КС) подразумевает возможность изменения состояния субъекта непосредственно через органы управления конкретной ЭВМ.

На практике, как правило, локальный сегмент включает в себя одну ЭВМ либо сегмент ЛВС.

Удаленным субъектом будем называть субъект, принадлежащий множеству субъектов внешнего сегмента КС. Очевидно, что множества субъектов локального и внешнего сегмента КС не пересекаются.

Доступ удаленного субъекта к локальному объекту подразумевает организацию сложного потока от удаленного субъекта к ассоциированным объектам локального субъекта, т.е. фактически управление локальным субъектом со стороны удаленного субъекта. Целью удаленного злоумышленника (пользователя, управляющего удаленным субъектом) является организация потоков от локальных объектов, не принадлежащих множеству L.

В случае разделения КС на локальный и внешний сегменты множество всех потоков «семантически» можно разделить на четыре (поток между субъектом и объектом означает поток между ассоциированными объектами субъекта и объектом):

1. Потоки между локальными субъектами и локальными объектами.
2. Потоки между локальными субъектами и удаленными объектами.
3. Потоки между удаленными субъектами и локальными объектами.
4. Потоки между удаленными субъектами и удаленными объектами.

Четвертая конструкция описывает взаимодействие субъектов какого-либо локального сегмента, отличного от выделенного, либо с локальными объектами (конструкция 1), либо удаленными относительно этого сегмента (конструкция 2). Третья конструкция описывает фактически взаимодействие между ассоциированными объектами субъектов локального сегмента и удаленными субъектами. Вторая конструкция также описывает потоки, которые могут реализоваться лишь через ассоциированные объекты удаленных субъектов. Первая конструкция подробно изучалась выше. Вторая и третья конструкция по смыслу тождественны, поскольку выбор локального сегмента КС произволен. Итак, будем рассматривать потоки между внешними субъектами и локальными объектами с учетом замечания об обязательном участии в потоке локального субъекта.

В терминах потоков рассмотрим межсубъектное взаимодействие между удаленным субъектом X и локальным субъектом S_i. Целью данного взаимодействия является реализация потока между локальным объектом O_j и ассоциированным объектом O_x субъекта X, причем данный поток проходит через ассоциированные объекты локального субъекта S_i.

Говоря о потоках, нельзя опускать потенциально возможное свойство порождения субъектом S_i нового субъекта S_i^*: ***Create***(S_i,Ov)->S_i^*

Порождение данного субъекта может произойти из объекта-источника:

1. локального сегмента КС,
2. внешнего сегмента КС.

К объектам локального сегмента можно отнести и ассоциированные объекты самого активизирующего субъекта S_i.

На ***рис. 1*** схематически представим рассматриваемое взаимодействие субъектов S_i и X.

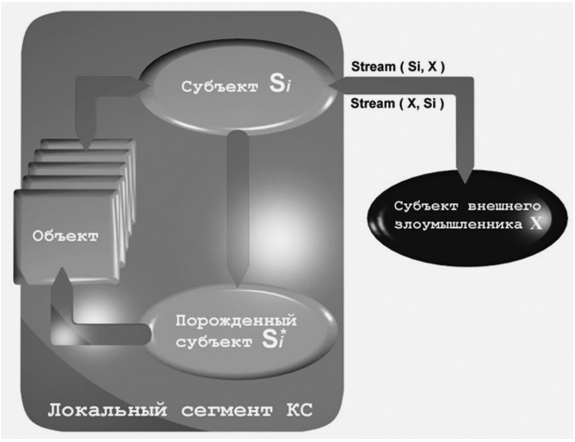


Рис. 1. К моделям воздействия внешнего злоумышленника на локальный сегмент КС

Введем следующие обозначения: X – субъект внешнего сегмента КС, который инициирует поток через S_i, S_i – телекоммуникационный субъект, принадлежащий подмножеству субъектов локального сегмента КС, O_j – объект локального сегмента КС, S_i* – субъект локального сегмента, порожденный субъектом S_i. O_k – ассоциированный объект субъекта S_i.

Рассмотрена следующая упрощенная модель работы территориально распределенной КС, которая состоит из двух ЭВМ. Имеются две ЭВМ, соединенные каналом связи. На рассматриваемых ЭВМ установлено телекоммуникационное ПО (ТПО), обеспечивающее совместную работу прикладных программ и аппаратуры передачи данных (модемов) для обмена информацией по каналу связи. Отметим, что передаваемая и принимаемая информация представляется в различных частях КС на различных уровнях (файлы, части файлов, пакеты). Очевидно также, что в телекоммуникационном ПО обеих ЭВМ имеется возможность чтения/записи на внешние носители прямого доступа, управляемая посылками из канала связи (в противном случае невозможно хранение принятой информации). Запись происходит с участием собственно телекоммуникационного ПО, либо прикладной программы принимающей станции. Кроме того, всегда существует возможность записи в оперативную память принимающей ЭВМ (буферизация). Буферизации могут подвергаться команды управления, поступающие от передающей ЭВМ, либо передаваемые данные.

Злоумышленника полагаем в данном случае лицом, которое имеет доступ к каналу связи и располагает идентичным по отношению к передающей ЭВМ комплексом программных и аппаратных средств. Таким образом, работу злоумышленника можно представить как работу либо передающей ЭВМ, либо принимающей, производящую посылку (или прием) на принимающую ЭВМ управляющей и содержательной информации. Обычно говорят о том, что на атакуемой злоумышленником ЭВМ работает некий программный субъект, который традиционно называется телекоммуникационным субъектом. Как правило, современный телекоммуникационный субъект обладает развитым сервисом, причем работает с информацией на уровне файлов ОС (например, продукты FTP Ssoftware). Злоумышленные действия в рассматриваемом случае возможны двух основных видов:

• пассивное воздействие, связанное с чтением данных с атакуемой ЭВМ и транспортировкой их на ЭВМ злоумышленника (воздействие инициировано с активной ЭВМ).

• активное воздействие, связанное с навязыванием данных (новых файлов) и модификацией уже существующих.

Обобщим данную модель и сформулируем ее на языке потоков.

Обозначим потоки от ассоциированного объекта O_x субъекта X к ассоциированному объекту O_k субъекта S_i и наоборот ***Stream***(X,O_x)->O_k и ***Stream***(X, O_k)->O_x. Предположим также, что свойства субъекта S_i таковы, что возможно существование потоков вида ***Stream***(S_i, O_j)->O_k и ***Stream***(S_i, O_k)->O_j. По свойству транзитивности потоков имеет место доступ субъекта X к объекту O_j через субъект S_i.

В локальном сегменте КС возможны также две основные ситуации, связанные с упомянутой выше возможностью порождения нового субъекта:

1. Доступ к объекту O_j со стороны субъекта S_i при управляющем воздействии субъекта X.

2. Порождение субъектом S_i из локального объекта нового субъекта S_i*, для которого существует поток ***Stream***(X, S_i*).

Вообще говоря, существенных различий с точки зрения доступа субъекта X к локальному объекту между ситуациями активности субъекта S_i или S*i не существует – в обоих случаях существуют сложный поток, включающий в себя ассоциированные объекты локального субъекта (S_i или S_i* соответственно).

Различие описанных ситуаций находится в области корректного межсубъектного взаимодействия в рамках ЛС КС, т.е. поскольку процесс активизации может быть инициирован субъектом X, то вновь порожденный субъект помимо реализации потоков к внешнему субъекту может реализовать и потоки к ассоциированным объектам субъектов ЛС КС, например, МБС и МБО. В данном случае нарушается основное условие попарной корректности субъектов.

6.3. Механизмы реализации политики безопасности в локальном сегменте компьютерной системы

Рассмотрение алгоритмов локальной защиты не затрагивает в указанных работах реализованной политики безопасности и носит в основном технический характер (реализация различных механизмов контроля доступа и т.д.).

Будем полагать, что в ЛС КС могут существовать только попарно корректные субъекты, замкнутые в ИПС (т.е. в составе ЛС КС существует МБС) с контролем целостности порождаемых субъектов. Кроме того, во множестве субъектов ЛС КС действует МБО, реализующих некоторую политику безопасности.

Рассмотрим общепринятую на сегодняшний день политику безопасности в контексте сформулированного взаимодействия локального и внешнего сегмента КС. Предварительно заметим, что в сформулированных условиях (генерация ИПС) для выделенного ЛС КС в случае отсутствия или неактивности ТПО) гарантированно реализуется любая политика безопасности, заданная в МБО (см. утверждение 3 части 1).

В случае существования субъекта с внешним управлением даже в случае его корректности относительно других субъектов условия утверждения 3 части 1 в общем случае не выполнены. Покажем это.

Предположим для произвольно выделенного нами ЛС КС наличие m пользователей P1, ..., P_n, ... , Pm.

Введем также понятие прав доступа субъектов к объектам как возможностей реализации потоков ***Stream***(S_m,O_m)->Ov (O_m ассоциированный объект S_m) – право доступа типа W (Write – запись) и потоков ***Stream***(S_m, Ov)->O_m – право доступа R (Read – чтение). Следовательно,

если субъект имеет право доступа R к объекту Ov, то это эквивалентно возможности существования потока ***Stream***(S_m,Ov)->O_m.

Вообще говоря, задаваясь некоторыми свойствами потока ***Stream***(S_m,O_m)->O_j или ***Stream***(S_m,O_j)->O_m, можно говорить о некотором конечном множестве прав G={g1, ..., gl}. В рассматриваемом случае мощность множества G равна 2.

Определение 3. Множеством доступных пользователю P_n субъектов S_n называется множество субъектов, которые данный пользователь может активизировать в ИПС из произвольного множества объектов-источников.

Определение 4. Множеством доступных пользователю P_n объектов $L_n(T)$ относительно права доступа T называется подмножество всех объектов, относительно которых реализуемы потоки соответствующего права доступа при активизации всех субъектов, входящих в S_n и имеющих право доступа T.

Рассмотрим теперь традиционную политику безопасности, связанную с понятием доступа пользователя (не субъекта!) к объекту. Данная политика задает $L_n(T)$ для любого подмножества множества субъектов S_n (если субъект потенциально способен реализовать поток, соответствующий праву доступа T) и в период работы пользователя P_n обеспечивает для выполнения потоков права T любому субъекту из S_n доступ к любому объекту, принадлежащему $L_n(T)$.

Для введенного множества прав это означает, что любой ***Stream***(S_i,O_k)->O_j разрешен, если S_i принадлежит S_n, а O_j принадлежит $L_n(W)$. Практически это означает, что в спроектированной с учетом такой политики безопасности системе защиты права пользователей определяются программами управления относительно пользователей, а не принадлежащих им программ (субъектов).

Определим политику безопасности с полным проецированием прав.

Определение 5. Политикой безопасности с полным проецированием прав пользователя или методом доступа с полным проецированием прав пользователя P_n на объекты КС называется такой порядок составления ПРД, при котором любой из субъектов, принадлежащий S_n, обладает одним и тем же правом доступа T к любому объекту множества $L_n(T)$.

Теперь сформулируем утверждение, описывающее потоки в ЛС КС в присутствии телекоммуникационного субъекта S_i.

Утверждение 1 (о распределенной КС с полным проецированием прав пользователя на субъекты).

В условиях действия политики безопасности с полным проецированием прав пользователя P_n на локальные объекты КС субъект X имеет доступ T к любому объекту множества $L_n(T)$ при условии существования потоков ***Stream*** (X,O_k)->O_x и ***Stream***(X,O_x)->O_k и доступности субъекта S_i для пользователя P_n.

Доказательство

Пусть у пользователя есть право R доступа к объекту O_j. В условиях полного проецирования прав на любой субъект это означает, что

для любого субъекта S_m из S_n (доступному пользователю) возможен поток ***Stream*** (S_m,O_j)->O_m, где O_m – ассоциированный объект S_m. По условию доказываемого утверждения S_i входит в S_n, следовательно, существует ***Stream***(S_i,O_j)->O_k. По условию утверждения существует и поток ***Stream***(X,O_k)->O_x. По свойству транзитивности потоков существует ***Stream***(X,O_j)->O_x. Это означает, что субъект X также имеет право доступа R к объекту O_j. Поскольку O_j произвольно выбран из множества $L_n(R)$, которое описано потоком ***Stream***(S_m,O_j)->O_m, то утверждение верно для всех объектов $L_n(T)$.

Аналогично доказывается утверждение в случае наличия у пользователя права доступа W к объекту O_j.

Утверждение доказано.

Из данного утверждения следует весьма важный факт, заключающийся в том, что система защиты от НСД любого ЛС КС, в котором гарантированно выполнена политика безопасности с полным проецированием прав доступа пользователей (к системам с такой политикой безопасности относится подавляющее большинство программно-аппаратных систем защиты локальных ресурсов, а также практически все штатные средства защиты в ОС) является потенциально ненадежной (т.е. допускающей возможность злоумышленных действий) при подключении к внешним сетям (т.е. при дополнении множества субъектов телекоммуникационным субъектом для взаимодействия с внешним сегментом КС). Необходима коррекция методов составления ПРД в системах, где возможно воздействие внешнего злоумышленника.

Сформулируем конструктивную политику безопасности, исключающую описанные выше ситуации.

Определение 6. Методом расщепления прав пользователя по отношению к множеству доступных ему субъектов называется такой порядок составления ПРД, при котором права доступа пользователя P_n задаются отдельно для каждого доступного пользователю субъекта (или подмножества субъектов), принадлежащего множеству S_n.

Легко видеть, что метод расщепления прав включает метод проецирования прав доступа (когда для любого субъекта задаются равные права доступа к объектам).

Сформулируем утверждение, описывающее условия защиты локальных объектов от внешнего злоумышленника.

Утверждение 2 (о доступе в системе с проецированием прав)

В условиях расщепления прав субъект X получит тот же доступ к объекту O_j, что и субъект S_i при условии существования потоков ***Stream***(X,O_x)->O_k и ***Stream***(X,O_k)->O_x и отсутствии в ЛС КС других субъектов, для которых существуют потоки между их ассоциированными объектами и O_x.

Доказательство

Поскольку других субъектов, связанных с внешним субъектом X в ЛС КС нет, то возможны потоки к объекту O_j только через ассоциированный объект O_k субъекта S_i. Поскольку между O_k и O_j возможен только поток, соответствующий праву доступа S_i, то и между O_x и O_j возможен только такой же поток.

Утверждение доказано.

Следствие. В условиях расщепления прав субъект X не получит доступ к объекту O_j в том случае, если субъект S_i не имеет доступ к O_j и не существует другого субъекта Sr в локальном сегменте КС, для которого существуют потоки между ассоциированными объектами данного субъекта и O_x.

Сформулированные и доказанные утверждения позволяют сформулировать методику проектирования защиты ЛС КС при условии попарной корректности всех субъектов (включая телекоммуникационный) и гарантированным выполнением политики безопасности.

Методика проектирования защиты описывается последовательностью шагов.

1. Формулируется политика безопасности с расщеплением прав пользователей (допустимо выделить два множества субъектов – чисто локальные и телекоммуникационные и установить раздельные права для этих групп).

2. Относительно каждого субъекта или групп субъектов формируется множество прав доступа к конкретным объектам (или группам объектов).

3. Реализуется МБО, выполняющий указанную политику безопасности.

4. Субъекты ЛС КС замыкаются в ИПС с контролем целостности объектов-источников.

Сформулируем одну из возможных политик безопасности, связанную с группированием объектов. Предположим, что объекты разделяются на три подмножества O_1 – доступные только пользователям ЛС КС (относительно O_1 все пользователи имеют доступ R и W), O_2 – множество доступных для внешних пользователей объектов с правом доступа R (например, объекты типа электронных объявлений) и O_3 – множество доступных для внешних пользователей объектов с правом W (например, почтовые ящики для входящих писем). Субъекты также разделены на две группы: S_i – локальные субъекты и S_2 – телекоммуникационные субъекты.

Тогда правила разграничения доступа формулируются следующим образом:

Таблица 1. Групповые правила разграничения доступа в ЛС КС

	O_1	O_2	O_3
S_1	RW	RW	RW
S_2	--	R	W

Вообще говоря, произвольная политика безопасности, разделяющая локальные и телекоммуникационные субъекты при их доступе к объектам будет гарантировать разделение также внутренних и внешних пользователей.

Рассмотрим возможность преднамеренной компрометации информации самим пользователем. Такая возможность реализуется иниции-

рованием потока ***Stream***(S_i,O_j)->O_x со стороны управляющего телекоммуникационным субъектом S_i пользователем, имеющим злоумышленные цели. Обозначим множество критичных к отправке во внешнюю сеть объектов как O_kr. При разделении прав между локальными и телекоммуникационными субъектами возможно задать такие ПРД, при реализации которых в МБО можно обеспечить полноценную работу локальных субъектов с объектами множества O_kr, но невозможность транспортировки объектов O_kr во внешнюю сеть. Обозначим множество некритичных к транспортировке и модификации локальных объектов как On.

Тогда ПРД относительно введенных групп субъектов S_1 и S_2 задаются таблицей.

Таблица 2. Правила разграничения доступа при запрете транспортировки вовне избранных объектов

	O_kr	On
S_1	RW	RW
S_2	--	RW

Теперь обратимся к ситуации, когда возможно порождение субъекта S_i*, нарушающего корректность межсубъектного взаимодействия. Возможны два случая:

• объект-источник Ov для порождения S_i* является внешним (не ассоциированным) для активизирующего субъекта S_i;

• объект-источник Ov является ассоциированным для субъекта S_i (в данном случае ассоциированный объект источник может быть неизменным или зависеть от информации, передаваемой субъектом X).

В первой ситуации при действии МБС с контролем целостности объекта источника порождение нового субъекта возможно с вероятностью, не превышающей вероятность принять нетождественный объект-источник за тождественный эталонному (данный параметр полностью определен свойствами функции контроля целостности).

Рассмотрим подробнее вторую ситуацию, обращая внимание на практические вопросы влияния на ассоциированные объекты других субъектов или данного.

Очевидно, что активное воздействие предоставляет широкие возможности для реализации как непосредственного, так и опосредованного НСД.

Пусть происходит процесс записи в оперативную память принимающей ПЭВМ (ПРМ ПЭВМ), принадлежащей ЛС КС. Практически это означает существование потока к ассоциированным объектам телекоммуникационного субъекта ЛС КС. При этом активизация нового субъекта в ПРМ ПЭВМ может произойти только тогда, когда процессор начнет выполнять инструкции в области памяти ПРМ ПЭВМ, куда был помещен принятый код. Это может произойти по трем причинам:

1. Область памяти для записи приходится на исполняемую программу, либо на область, в которую постоянно передается управление (таблица прерываний, драйверы операционной системы и т.д.).

2. Запись происходит в область, находящуюся вне критичных для работы программной среды зон, но происходит некорректно (некорректность возникает в основном при записи длинных отрезков принимаемых данных в буфера меньшего размера); за счет этого производится «переполнение» (нарушение попарной корректности), и дальнейшая работа происходит в условии п.1.

3. Запись приходится в область размещения самого ТПО и в ней образуется код, который в дальнейшем используется.

Все описанные ситуации означают нарушение корректности межсубъектного взаимодействия. Очевидно, что все они могут быть исключены при выполнении условий проектирования ИПС для телекоммуникационного ПО.

Посылка РПВ в виде исполняемого файла используется чаще всего для удаленного внедрения закладок и при достаточной тривиальности почти всегда приводит к успеху.

Интересно, что зачастую пользователь провоцируется на запуск некоторого исполняемого файла. Так, в одном случае пользователям передавался набор файлов, среди которых был архивированный (ZIP) файл. Для его разворачивания в текущей директории (куда был принят набор файлов) запускалась программа PKUNZIP, на которую почти всегда установлен путь. Однако мало кто из пользователей замечал, что в наборе переданных файлов уже есть программа с таким именем. Естественно, запускалась именно она и помимо разворачивания архива производилась установка закладки.

Легко видеть, что описанные атаки связаны с формированием объекта-источника, содержащего злоумышленные действия, в составе объектов ЛС КС. Очевидно, что в условиях действия ИПС с контролем целостности объектов-источников активизация субъекта из объекта-источника, не входящего в список объектов-источников для множества S_n ЛС КС, невозможна в принципе, а при замене одного из «легальных» объектов-источников возможна с вероятностью принять измененный объект за эталонный.

Перейдем теперь к анализу защитных механизмов, ориентированных на отражение внешнего злоумышленного воздействия на весь ЛС КС на сетевом уровне.

6.4. Метод межсетевого экранирования. Свойства экранирующего субъекта

Недостатки классических политик безопасности, связанных с полным проецированием прав пользователя на все множество субъектов привели к появлению методов защиты, связанных с «экранированием» ЛС КС от внешнего сегмента КС. Суть экранирования состоит в прохождении потоков между O_k и Ox через дополнительный объект (возможно более низкого уровня представления), ассоциированный с субъектом-анализатором потока.

Рассмотрим модель взаимодействия локального и внешнего сегментов КС, когда поток от субъекта X к объекту O_j проходит кроме ассоции-

рованных объектов O_m субъекта S_i через некоторый объект O_f, ассоциированный с субъектом S_f (т.е. поток отображается на ассоциированные объекты данного субъекта).

Обозначим ***Stream***(X,O_x)->O_f, ***Stream*** (S_f,O_f)->O_m и ***Stream***(S_i,O_m)->O_j потоки от ассоциированных объектов субъекта X к объекту O_j через субъект локального сегмента КС S_i и прохождением потока через S_f.

Обозначим также ***Stream***(S_i,O_j)->O_m, ***Stream***(S_i,O_j)->O_f, ***Stream***(S_f,O_f)->O_x поток к ассоциированным объектам субъекта X.

Аналогичная ситуация с точки зрения участия субъекта в потоке была рассмотрена выше, где было изучено взаимодействие субъектов с точки зрения корректной передачи информации и возврата результата преобразования. В данном случае априорно заданная цель субъекта S_f иная. Данный субъект рассматривается как некоторый фильтр, который определяет факт доступа к объекту O_j со стороны субъекта X, либо фиксирует потоки между O_k и O_x. Допускаем также, что поток может рассматриваться на различном уровне относительно объекта O_j.

Предположим, что объект O_j имеет уровень представления R – максимальный для локального сегмента КС. Тогда очевидно, что субъект S_f (субъект-фильтр) участвует в потоке к объектам уровня не выше R.

Рассмотрим субъектно-зависимую функцию декомпозиции объекта O_j уровня R на последовательность O*r объектов уровня r<R:

Decomp(O_j,R,S_i)->O*r, где O*r = (O_j1r, ..., $O_j$kr).

Смысл применения функции Decomp состоит в описании взаимодействия пары субъектов S_i и S_f: в момент времени t субъект S_i инициирует поток ***Stream***(S_i,$O_j$mr)[t]->O_f[t]. t=1, ... , k. Причем $O_j$mr[t] и O_f[t] тождественны, за интервал времени k через O_f проходит весь O_j.

Каждый из объектов $O_j$mr множества O*r можно представить объектом вида Ajm||Bjm, где || операция конкатенации объектов. Объект C, полученный как результат операции конкатенации объектов как слов A=(a1, ..., an) и B=(b1, ..., bm) описывается как C=(a1, ...,an,b1, ...,bm). Смысл расщепления объекта $O_j$mr состоит в выделении части, являющейся элементом объекта O_j.

Выделяются фрагменты объекта O_j, составляющие при конкатенации весь объект O_j (Bjm) и дополнительные объекты Ajm, необходимые для преобразования подобъектов Bjm на уровень r.

Назовем Bjm информационной частью подобъекта, а Ajm управляющей или адресной.

Поясним введенную конструкцию на примере декомпозиции файла на пакеты стека TCP/IP. При такой декомпозиции файл (объект O_j) разделяется на части (одинаковой или различной длины), которые составляют часть тела пакета IP. При этом дополнение (Ajm) каждого m-го пакета составляет адресную часть пакета, контрольную сумму и т.д.

Утверждение 3 (о существовании декомпозиции на подобъекты).

Если существует поток ***Stream***(O_j,X)->O_x, где O_x – ассоциированный объект субъекта X и объекты O_j и O_x тождественны, то для любого субъекта X существует декомпозиция каждого объекта $O_j$mr=Ajm||Bjm, при которой ||Bjm = O_j, i=1, ..., k (т.е. конкатенация всех информационных подобъектов составляет целый объект O_j).

Доказательство

Исключим из рассмотрения ситуацию, когда конкатенация Bjm составляет объект D, который включает объект O_j или O_x (которые тождественны по условию утверждения). В таком случае некоторые объекты Bjm редуцируются так, чтобы объекты D и O_j стали тождественными. Осталось рассмотреть ситуацию, когда конкатенация Bjm составляет подобъект O_j.

От противного. Предположим, что существует субъект, для которого при любом разбиении на Ajm и Bjm конкатенация Bjm составляет объект Dj нетождественный O_j, это в свою очередь означает, что нетождественны O_j и O_x. Противоречие с условием утверждения.

Из доказанного утверждения следует, что на произвольном уровне r поток подобъектов, проходящий через субъект-фильтр, содержит полную информацию о всем объекте O_j. Однако поток подобъектов проходит через фильтр в течение интервала времени T=k и структура объектов $O_j$mr (как и число k) зависит от конкретного субъекта.

Смысл данного утверждения достаточно очевиден. На языке субъектов сборки-разборки пакетов оно означает, что из последовательности пакетов всегда полностью восстанавливается передаваемый объект.

При рассмотрении любого подмножества подобъектов, составляющих O_j, получение полной информации о том, к какому именно объекту ЛС КС происходит доступ не представляется возможность установить. В связи с этим как минимально необходимую задачу реализации ПБ в субъекте-фильтре необходимо рассмотреть сборку полного объекта из подобъектов. Заметим, что ПБ реализуется на уровне целого объекта, а не составляющих его подобъектов.

Сформулируем задачу корректного экранирования на уровне r, вводя понятие корректного экранирования в следующем определении.

Определение 7. Субъект S_f называется корректно экранирующим (или корректно фильтрующим) на выход относительно субъекта S_i, если для любого объекта O_j при ***Stream***(S_i,O_j)->O_f по последовательности O_f[1], ..., O_f[k] можно однозначно восстановить O_j.

Определение 8. Субъект S_f называется корректно экранирующим (или корректно фильтрующим) на вход относительно субъекта S_i, если для любого объекта O_j при ***Stream***(S_i,O_f)->O_j по последовательности O_f[1], ..., O_f[k] можно однозначно восстановить O_j.

Определение 9. Субъект S_f называется корректным фильтром, если он является корректно фильтрующим на вход и на выход.

Утверждение 4 (Основная теорема о корректном экранировании).

Экранирующий субъект S_f, участвующий в потоке подобъектов уровня r будет корректным на вход и на выход, тогда и только тогда, когда для любого S_i и для любого Oj по последовательности O*r однозначно определяется объект O_j.

Доказательство

Достаточность утверждения следует из определения корректности на вход и на выход.

Докажем необходимость. От противного. Если экранирующий субъект является корректным фильтром относительно любого объекта, но по

последовательности подобъектов неоднозначно определяется объект O_j, то возникает противоречие с условием. Утверждение доказано.

Основная теорема о корректном экранировании, хотя и является критерием, но, тем не менее, недостаточно конструктивна. Кроме того, субъект-фильтр не производит разделение потоков на N и L. Необходимо отметить два принципиально важных момента:

1. Субъект-фильтр должен иметь информацию о самих объектах O_j для осуществления сравнений.

2. Субъект-фильтр должен иметь информацию о разрешенных или запрещенных потоках между объектами O_k и O_x.

Введем определение фильтра, учитывающего разделение потоков на множества N и L.

Определение 10. Гарантированно-изолирующим фильтром называется корректный фильтр, который разрешает прохождение потока ***Stream***(X, O_x)->O_j и ***Stream***(X,O_j)->O_x только для потоков, принадлежащих множеству L.

На практике субъект-фильтр не имеет доступ к множеству объектов ЛС КС. В этом случае задача восстановления объекта O_j по последовательности подобъектов O_jm не может быть решена в явном виде.

Утверждение 5 (необходимое условие гарантированной изоляции для субъекта-фильтра)

Для того чтобы фильтр был гарантировано изолирующим необходимо обеспечить существование потока ***Stream***(S_f,O_j)->Oe, где Oe – ассоциированный объект S_f, служащий для сравнения с восстановленным из последовательности подобъектов объектом и выполнить условие тождественности Oe и O_j.

Доказательство

От противного. Пусть указанный в условии поток отсутствует, либо отображение, задаваемое потоком нетождественно, тогда невозможно восстановить по последовательности подобъектов весь объект за отсутствием эталона сравнения.

Существующие методики проектирования и реализации экранирующих субъектов и управления ими рассматривают процесс фильтрации применительно к особенностям функции Decomp. Рассматривается полученная после декомпозиции последовательность с точки зрения информационных подобъектов (Ajm), которые интегрально описывают подмножество объектов, относящихся к выделенному адресу, либо рассматривают указанную последовательность относительно некоторого субъекта, который производит декомпозицию на подобъекты.

С точки зрения особенностей работы субъекта, производящего декомпозицию объекта, зарубежные работы вводят понятие сервиса, описывая его как субъект, в котором локализованы конкретные алгоритмы декомпозиции (т.е. порождающие некие свойственные только данному субъекту последовательности подобъектов).

Для описания доступа из внешней сети выделяется множество доступных сервисов, которые описывают множество субъектов, для которых разрешается поток к произвольному объекту ЛС КС.

При этом действительны сформулированные выше замечания относительно политики безопасности, реализованной в ЛС КС. Любая политика с полным проецированием прав будет некорректна относительно сервиса, допускающего доступ субъекта X к объекту ЛС КС. Конечно, свойства телекоммуникационного субъекта ЛС КС могут быть таковы, что опасные для защищенности ЛС КС потоки между O_x и O_k могут быть исключены, следовательно, ограничение доступных сервисов имеет смысл для построения защиты.

С другой стороны, фильтрация сервисов является аналогом ограничения множества локальных субъектов, могущих иметь доступ к объектам ЛС КС.

Утверждение 6 (о тождестве фильтра сервисов и изолированной программной среды в рамках локального сегмента КС)

Возможности внешнего злоумышленника по отношению к объектам ЛС КС одинаковы как в случае существования фильтрующего субъекта S_f – фильтра сервисов, допускающего существование только сервисов S_1, ..., S_m из S_n, так и генерации ИПС с включением субъектов S_1, ..., S_m.

Доказательство

Выделим субъект S_i, принадлежащий ЛС КС, допускаемый фильтром сервисов. Зафиксируем все потоки между ассоциированным объектом O_k субъекта S_i и O_x. Очевидно, что при включении в ИПС субъекта S_i возможны точно такие же потоки. Распространяя на остальные субъекты S_1, ..., S_m, доказываем утверждение.

Из приводимого утверждения следует, что методы защиты, связанные с разрешенными сервисами, в принципе эквивалентны методу генерации ИПС для ЛС КС, в которую включены локальные субъекты, обеспечивающие телекоммуникационное взаимодействие. В сущности, уменьшение множества субъектов как методом генерации ИПС, так и методом фильтрации сервисов является только гарантией выполнения политики безопасности, реализованной в субъектах ЛС КС, либо субъекте-фильтре.

С другой стороны, свойства произвольного субъекта X внешнего сегмента относительно S_f могут быть произвольны.

Аксиома. При произвольном составе субъектов внешнего сегмента КС возможно формирование подобъектов (на уровне ассоциированных объектов O_f *субъекта-фильтра* S_f *для потока* ***Stream(X,***O_x***)->***O_f*) с произвольной адресной и информационной частью.*

Исходя из данного положения, можно утверждать, что фильтрация подобъектов изолированно от содержания объектов ЛС КС в общем случае потенциально ненадежна относительно любых критериев фильтрации при возможности управления телекоммуникационным субъектом ЛС КС со стороны злоумышленника.

Требование гарантированной фильтрации в части доступа S_f к любому объекту ЛС КС технически достаточно сложно реализовать в силу возможной гетерогенности операционных сред ЛС КС, скоростных параметров и т.д. Однако можно предложить альтернативный метод проектирования гарантированно-изолирующего субъекта-фильтра.

Предположим, что в субъекте-фильтре однозначно выделяются информационные подобъекты и реализация ***Stream***(S_i,O_jm)->O_f является тождественным отображением (технически это означает безошибочную передачу в тракте фильтр – ЭВМ).

Для всех объектов ЛС КС вычислим хеш-функции H(O_j,Kg)=hjg и гарантируем их доступность для субъекта-фильтра, хеш-функция возможно зависит от индивидуальной информации пользователя K_i.

Процедура фильтрации на выход (относительно существующих объектов) формулируется следующим образом:.

1. По последовательности подобъектов O_j1r, ..., $O_j$kr восстанавливается объект Dj.

2. Вычисляется H(Dj,K_i)=hji*.

3. Вычисленное значение hji* сравнивается с hji.

4. В случае совпадения проверяются права доступа к объекту O_j.

5. В случае доступности объекта для передачи во внешний сегмент КС разрешается передача подобъектов, соответствующих декомпозиции объекта, во внешнюю сеть.

6. В случае несовпадения передача запрещается.

Указанный метод может быть дополнен фильтрацией сервисов для обеспечения достоверного восстановления объекта по последовательности подобъектов.

6.5. Модель политики безопасности в распределенной системе

Данная модель приводится по «A multilevel security policy for networks»[1] и описывает требования безопасности для построения защищенной сети. Предполагается, что имеется множество защищенных и незащищенных станций, связанных в сеть.

Каждый компонент сети имеет классификацию защищенности; каждый пользователь сети имеет уровень благонадежности. Предполагается, что на множестве классов защищенности установлен частичный порядок ≥. Отношение частичного порядка – рефлексивно, антисимметрично и транзитивно. Для двух классов безопасности sc_1 и sc_2, если $sc_1 \geq sc_2$, говорят, что sc_1 доминирует над sc_2. Для двух элементов sc_1 и sc_2:

• множество классов безопасности, доминирующее над sc_1 и sc_2 – непустое и содержит наибольшую внешнюю границу, доминирующую над всеми классами.

• множество классов безопасности, над которыми доминируют sc_1 и sc_2 – непустое и содержит наименьшую внешнюю границу (inf), над которой доминируют все классы.

Далее предполагается, что сущности X и Y(субъекты или объекты) могут иметь более одной классификации Scls(X)={scx_1, scx_2,... scx_n} и Scls(Y)={scy_1, scy_2,... $scy_м$}. Допустим, что sc – класс безопасности. Тогда:

- $Scls(X) \geq sc \Leftrightarrow scx_i \geq sc$, $\forall i$, $1 \leq i \leq n$;

1. Vijay Varadharajan «A multilevel security policy for networks», 1990.

- $Scls(X) \le sc \Leftrightarrow scx_i \le sc$, $\forall i$, $1 \le i \le n$;
- $Scls(X) \ge Scls(Y) \Leftrightarrow \forall scx_i$ $(1 \le i \le n)$, $scx_i \ge lub(scy_1, scy_2, ... scy_м)$.

Формальная модель.

Определим модель безопасности сети MODEL:

MODEL = <S,O,A,s0>

S – множество состояний:

O – множество операций системы;

A – функция системы;

s_0 – начальное состояние системы.

Множество S моделирует состояние переменных, относящихся к защищенности сети. Множество O описывает сетевые операции, а множество A – переходы модели из одного состояния в другое после применения последовательности операций из множества O. Состояние s0 описывает начальное состояние системы.

Определим основные множества, используемые для описания модели.

-*Sub*: множество всех субъектов сети. Включает множество всех пользователей(Users) и всех процессов(Procs) сети

Sub=Procs ∪ Users

- *Obj*: множество всех объектов сети. Включает множество сетевых компонентов(NC) и информационных блоков(UI) сети.

Obj=NC ∪ UI

Обычно множество сетевых компонентов включает хосты (H), устройства ввода-вывода(IOD) и устройства вывода(OD), а множество информационных блоков включает файлы и сообщения.

NC=H ∪IOD ∪OD

- Scls: множество классов безопасности. Предполагается, что на этом множестве определен частичный порядок.

- Rset: множество ролей пользователя. Включает роль администратора безопасности.

- Strings: множество символьных строк.

Состояние системы.

Каждое состояние s∈ S описывается:

s=<Subs, Objs, authlist, conlist, accset, subcls, objcls, curcls, subrefobj, role, currole, term, contents>, где:

- Subs определяет множество субъектов в состоянии s;

- Objs определяет множество объектов в состоянии s;

- authlist – множество, состоящее из элементов в форме (sub, nc), где sub∈ Subs nc∈ Objs; Существование элемента (sub_1, nc_1) в множестве, отмечает то, что субъект sub_1 имеет право на связь с компонентом сети nc_1.

- connlist – множество, состоящее из элементов в форме (sub, nc). Это множество отражает текущее множество допустимых соединений в данном состоянии.

- subcls – Sub → SCls.

subcls – функция, приписывающая каждому субъекту его степень доверия.

- objcls – Obj → PS(SCls), где PS обозначает мощность множества.

objcls – функция, приписывающая каждому объекту один или более уровней классификации. Предполагается, что выходные устройства и информационные блоки имеют единственный уровень классификации, тогда как хосты могут иметь несколько классификаций.

- curcls – Sub → Scls.

curcls – функция, задающая текущую степень доверия субъекта.

subrefobj – Sub → PS(Obj).

subrefobj – описывает множество объектов, к которым субъект может обратиться в данном состоянии.

role: Users → PS(Rset)

role описывает множество ролей, для которых авторизован пользователь.

currole: Users → Rset

currole описывает текущую роль пользователя.

term: Users → IOD

term определяет терминал, с которого пользователь вошел в систему.

contents – IU → Strings

contents – функция, которая отображает множество информационных блоков в множество строк. Она выделяет содержание информационных объектов.

Для nc $\in$ IOD$\cup$OD, view(nc) – множество упорядоченных пар{(x_1, y_1), (x_2, y_2),... (x_n, y_n)}, где каждое y_i отражается на компоненте nc. Каждое x_i – информационный блок и y_i – результат применения функции contents к x_i.

Предположения безопасности.

Модель сети содержит следующие предположения безопасности.

1. На хостах сети существует надежная схема пользовательской аутентификации. Каждый пользователь и процесс в сети имеет уникальный идентификатор.

2. Только пользователь с ролью Администратор Безопасности Сети может присваивать классы безопасности субъектам и компонентам сети, и роли пользователям.

3. Все сущности сети имеют сравнимые классы безопасности.

4. Имеет место надежная передача данных по сети.

Безопасное состояние.

Определим условия, при которых состояние сети безопасно. С целью определения данных условий, рассмотрим различные фазы, через которые проходит система во время выполнения операций.

Фаза доступа к системе

Предполагается, что существует надежный механизм идентификации-аутентификации. Эти аспекты не включены в данную модель. Но существует требование, по которому степень доверия пользователя должна быть больше или равна классификации терминала, с которого он получил доступ к системе. Более того, текущая степень доверия пользователя не должна быть больше его максимальной степени до-

верия, а его роль должна принадлежать к списку его авторизованных ролей. Это дает следующие ограничения.

Ограничения при доступе к системе

Предложение 1: Состояние s удовлетворяет ограничениям при доступе к системе, если $\forall x \in$ Userss

- subcls(x) ≥ objcls(term(x))
- subcls(x) ≥ curcls(x)
- currole(x) ∈ role(x)

Фаза установления связи

После получения доступа к системе, пользователь может захотеть установить связь с другими компонентами сети. Для определения доступных соединений, должны поддерживаться дискретная и мандатная политика сети.

Ограничения связи

Предложение 2. Состояние s удовлетворяет ограничениям связи, если $\forall$(sub, nc)$\in$ connlist

- $\forall$(sub, nc) ∈ authlist
- если nc$\notin$ OD $\Rightarrow$ curcls(sub) ≥ sup(oc_1, oc_2,... oc_j), где objcls(nc)={ oc_1, oc_2,... oc_j }
- если nc ∈ OD $\Rightarrow$ objcls(nc) ≥ subcls(sub)

Первое условие дает контроль дискретного доступа, т.е. объект, к которому осуществлен запрос на связь, должен находиться в списке объектов, к которым имеет доступ субъект.

Второе ограничение говорит о том, что если запрашиваемый сетевой компонент – не устройство вывода, то для установления соединения, текущая степень доверия к пользователю должна быть, по крайней мере, не меньше самой нижней классификации объекта.

Третье условие относится к устройствам вывода. Степень доверия к субъекту должна быть не больше класса безопасности компонента сети, для предотвращения утечки информации через выходное устройство.

Другие условия

1. Классификация информации, которая может быть просмотрена через устройство ввода-вывода, должна быть не больше классификации данного устройства.

2. Роль пользователя в данном состоянии должна принадлежать к списку ролей, к которым авторизован пользователь.

Теперь определим безопасное состояние.

Определение 11. Состояние s ***безопасно***, если:

1. s удовлетворяет *Ограничениям при доступе к системе.*
2. s удовлетворяет *Ограничениям связи.*
3. $\forall z \in$ (IODs$\cup$ODs), $\forall x \in$ (x, contents(x)) $\in$ view(x) $\Rightarrow$ objcls(z)≥objcls(x)
4. $\forall z \in$ Users, currole(u) ∈ role(u).

Начальное состояние.

Предполагается, что начальное состояние системы s0 определено таким образом, что оно удовлетворяет условиям безопасного состояния, описанным выше.

Операции.

Операция связи

Операция connect(sub, nc) позволяет субъекту sub связаться с удаленным объектом сети nc.

Операции манипуляции информацией

После установления связи с удаленным компонентом, субъект может выполнять операции, которые требуют манипуляции с информационными объектами. Манипуляция информацией состоит из двух этапов: этап получения доступа к информации, на котором субъект связывается с информационным объектом, над которым он хочет произвести манипуляции, и этап манипуляции, на котором действуют ограничения, принятые в модели Белла-ЛаПадула для операций чтения, записи, добавления и выполнения. В данной модели рассматривается операция, выполняющая передачу информации от одного сетевого компонента к другому, как самая важная при рассмотрении сети. (Фактически данная операция является частью всех других операций).

Операция получения доступа

Операция bind(iobj, nc) позволяет субъекту sub получить доступ к информационному объекту iuobj на сетевом компоненте nc.

Операция передачи информации

Операция transfer(iuobj1, nc1, iuobj2, nc2) позволяет субъекту sub добавить содержание информационного блока объекта iuobj1 на сетевом компоненте nc1 к содержанию информационного блока объекта iuobj2 на сетевом компоненте nc2.

Операция освобождения

Операция unbind(sub, iuobj) позволяет субъекту sub освободить связь с объектом iuobj. До выполнения данной операции iuobj2 $\in$ subrefobj(sub), а после выполнения – iuobj $\notin$ subrefobj(sub).

Другие операции

Рассмотрим другие операции, модифицирующие атрибуты безопасности субъектов и объектов. Обычно они включают операции, изменяющие степень доверия к субъектам, классификацию информационных блоков, а также множества доступа к информационным блокам. В случае модели безопасности сети, необходимо добавить операции, такие как присвоение классификации сетевым компонентам и изменение ролей пользователей.

Присвоение классификации сетевому компоненту

Операция assign-sclass-ncobj(nc,scls) позволяет субъекту sub установить классификацию сетевому компоненту nc. То есть objcls'(nc)={scls}. Данная операция применима только в том случае, когда компонент не используется. Более того, только Администратор Безопасности Сети авторизован для проведения данной операции.

Присвоение степени доверия пользователю

Операция assign-sclass-user(usr, scls) позволяет субъекту sub установить степень доверия пользователю usr.

Присвоение пользователю текущей степени доверия

Операция assign-curclass-user(usr,scls) позволяет субъекту sub установить текущую степень доверия к пользователю usr.

Присвоение пользователю роли

Операция assign-role-user(usr, rlset) позволяет субъекту sub установить пользователю usr множество ролей.

Присвоение пользователю текущей роли

Операция assign-currole-user(usr,rl) позволяет субъекту sub изменить текущую роль пользователя usr.

Установка списка доступа

Операция setauthlist(al) позволяет субъекту установить список доступа.

Функции системы.

Функции системы описывают переход системы из одного состояния в другое, поле применения одиночной операции или их последовательности, так как это было описано выше. То есть:

$A: Sub \times O \times S \rightarrow S$

$s'=A(sub, op, s)$ – результирующее состояние после применения операции $o \in O$, примененной субъектом $sub \in Sub$ в состоянии $s \in S$.

О функции А говорят, что переход безопасен, если он удовлетворяет условиям, описанным в предыдущих пунктах.

6.6. Оценка качества и сертификация средств сетевой безопасности

В России вопросы разработки и сертификации средств межсетевой защиты регулируются Руководящим документом «Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации», разработанный рядом ведущих разработчиков и экспертов в области сетевой безопасности и принятый Гостехкомиссией РФ в 1997 г..

Этот руководящий документ устанавливает классификацию межсетевых экранов (МЭ) по уровню защищенности от несанкционированного доступа (НСД) к информации на базе перечня показателей защищенности и совокупности описывающих их требований.

Под сетями ЭВМ, распределенными автоматизированными системами (АС) в данном документе понимаются соединенные каналами связи системы обработки данных, ориентированные на конкретного пользователя.

В терминологии Гостехкомиссии РФ МЭ представляет собой локальное (однокомпонентное) или функционально-распределенное средство (комплекс), реализующее контроль за информацией, поступающей в АС и/или выходящей из АС, и обеспечивает защиту АС посредством фильтрации информации, т.е. ее анализа по совокупности критериев и принятия решения о ее распространении в (из) АС.

Руководящий документ разработан в дополнение к Руководящим документам Гостехкомиссии России «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации»,

М., Военное издательство, 1992 и «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации», М., Военное издательство, 1997. Более подробно упомянутые документы рассмотрены в главе 9.

Деление МЭ на соответствующие классы по уровням контроля межсетевых информационных потоков с точки зрения защиты информации необходимо в целях разработки и применения обоснованных и экономически оправданных мер по достижению требуемого уровня защиты информации при взаимодействии сетей ЭВМ, АС.

Документ устанавливается пять классов защищенности МЭ.

Каждый класс характеризуется определенной минимальной совокупностью требований по защите информации.

Самый низкий класс защищенности – пятый, применяемый для безопасного взаимодействия АС класса 1Д с внешней средой, четвертый – для 1Г, третий – 1В, второй – 1Б, самый высокий – первый, применяемый для безопасного взаимодействия АС класса 1А с внешней средой.

Требования, предъявляемые к МЭ, не исключают требований, предъявляемых к средствам вычислительной техники (СВТ) и АС в соответствии с руководящими документами Гостехкомиссии России «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации» и «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации».

При включении МЭ в АС определенного класса защищенности, класс защищенности совокупной АС, полученной из исходной путем добавления в нее МЭ, не должен понижаться.

Для АС класса 3Б, 2Б должны применяться МЭ не ниже 5 класса.

Для АС класса 3А, 2А в зависимости от важности обрабатываемой информации должны применяться МЭ следующих классов:

– при обработке информации с грифом «секретно» – не ниже 3 класса;

– при обработке информации с грифом «совершенно секретно» – не ниже 2 класса;

– при обработке информации с грифом «особой важности» – не ниже 1 класса[2].

2. Руководящий документ. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. // Информационный бюллетень JetInfo. 17-18, 1997 – с. 3-9.

7. КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ И НАДЕЖНОСТЬ. ЗАЩИТА В ОПЕРАЦИОННЫХ СИСТЕМАХ

7.1. Особенности современных операционных систем и группы требований, предъявляемых к ним

К операционной платформе компьютерных систем, обеспечивающих работу в режиме высокой надежности и защищенности предъявляются достаточно высокие требования. Требования, касающиеся универсальных операционных систем, достаточно хорошо известны[1], а вот те, которым должны отвечать ОС, применяемые, в частности, в платежных системах и в защищенных КС, обрабатывающих критичную информацию, находятся пока в процессе формирования.

Общие требования по безопасности можно обобщить в следующих положениях:

• Должна осуществляться идентификация и аутентификация пользователей при входе в систему.

• Должна осуществляться регистрация входа/выхода пользователей в систему/из системы. Должна осуществляться регистрация введения новых пользователей системы и изменения их полномочий. Должна осуществляться регистрация изменений статуса объектов доступа с регистрацией внесённых изменений, времени и даты изменений, а также кем внесены изменения. Введение новых пользователей системы и изменение их полномочий имеют право производить только специально выделенные администраторы системы.

• Средства разграничения доступа должны контролировать доступ именованных субъектов и управляющих ими пользователей к именованным объектам.

• Должны быть реализованы механизмы (процедуры) контроля несанкционированного случайного и/или преднамеренного искажения (изменения, модификации) и/или разрушения компонентов, содержащих исполняемый код. Должно проводиться периодическое тестирование функций средств защиты информации.

• Должна осуществляться защита всей конфиденциальной информации, передаваемой в рамках внешнего сегмента КС. Информация, передаваемая во внешнем сегменте КС, должна быть защищена с использованием криптографических средств (зашифрована), или для ее передачи должны использоваться защищенные каналы связи. Должна осуществляться защита информации, записываемой на отчуждаемые носители (внешние относительно КС).

• Используемые средства криптографической защиты информации должны быть сертифицированы.

1. Philip Melanson, Siamak Tafazoli. A Selection Methodology for the RTOS Market, Canadian Space Agency, 2003.

Сформулируем также основные требования по надежности:

• ОС должны удовлетворять требованиям «жесткого» реального времени (детерминизму поведения при различных нагрузках на систему). Например, таким требованиям, как детерминированное время реакции задачи на прерывание или детерминированное время переключения контекста между процессами.

• ОС должны обеспечивать высокую степень «живучести» системы, чтобы при отказе какой-либо части программного обеспечения другая продолжала нормально функционировать, ОС должна также гарантировать невозможность полного отказа системы.

• ОС должна удовлетворять жестким требованиям по качеству ПО, что подразумевает соответствие различным отраслевым, национальным и международным стандартам. Особенностью требований к ОС является то, что ПО должно иметь доказанное качество.

• Вероятность сбоя в ПО должна быть очень мала.

Несколько слов о терминологии. В английской литературе используются два термина – safety и security, которые могут быть похожим образом переведены на русский язык. Однако, как правило, они обозначают несколько различающиеся понятия. Чтобы не путаться в дальнейшем, будем использовать safety как синоним надежности (вторая часть требования), а security – как синоним секретности (первая часть требований).

В настоящее время определены следующие базовые стандарты надежности для ОС.

• Стандарт DO-178 «Software Consideration in Airborne Systems and Equipment Certification» разработан и поддерживается ассоциацией RTCA (Radio Technical Commission for Aeronautics, www.rtca. org). Первая его версия была принята в 1982 г., вторая (DO-178A) – в 1985-м, текущая DO-178B – в 1992 г., а принятие новой версии, DO-178C, произошло в конце 2005-го. Стандартом предусмотрено пять уровней серьезности отказа, и для каждого из них определен набор требований к программному обеспечению, которые должны гарантировать работоспособность всей системы в целом при возникновении отказов данного уровня серьезности:

• уровень A – ПО должно обеспечивать защиту от сбоев, приводящих к катастрофическим (catastrophic) последствиям, и удовлетворять 66 требованиям;

• уровень B – ПО должно обеспечивать защиту от сбоев, приводящих к опасным (hazardous) последствиям, и удовлетворять 65 требованиям;

• уровень C – ПО должно обеспечивать защиту от сбоев, приводящих к серьезным (major) последствиям, и удовлетворять 57 требованиям;

• уровень D – ПО должно обеспечивать защиту от сбоев, приводящих к незначительным (minor) последствиям, и удовлетворять 28 требованиям;

• уровень E – ПО должно обеспечивать защиту от сбоев, не приводящих ни к каким последствиям.

• Стандарт ED-12B – европейский аналог DO-178B – определяется EUROCAE (The European organisation for civil aviation equipment, www.eurocae.org).

• ARINC 653 – «Avionics Application Software Standard Interface» – разработан компанией ARINC (Aeronautical Radio, Inc.) в 1997 г. Этот стандарт определяет универсальный программный интерфейс APEX (Application/Executive) между ОС и прикладным ПО (www.arinc.com/cf/store/ documentlist.cfm). Требования к интерфейсу между прикладным ПО и сервисами операционной системы определяются таким образом, чтобы разрешить прикладному ПО контролировать диспетчеризацию, связь и состояние внутренних обрабатываемых элементов. В 2003 г. принята новая редакция этого стандарта. ARINC 653 в качестве одного из основных требований для ОС вводит архитектуру изолированных (partitioning) виртуальных машин.

• Общие критерии для оценки секретности информационных технологий (Common Criteria for Information Technology Security Evaluation, CCITSE) – это набор требований и условий секретности, одобренный Агентством национальной безопасности и Национальным институтом стандартов и технологий США (United States National Security Agency/ National Institute of Standards and Technologies, http://csrc.nist.gov/cc/), а также соответствующими органами других стран (в данный момент еще 13 стран помимо США). Первая версия требований была опубликована в январе 1996 г., вторая – в апреле 1998-го. В 1999 г. CCITSE получил статус международного стандарта ISO 15408. Дополнительную информацию по сертификации CCITSE можно найти на сайте: www.commoncriteria.org.

• CCITSE определяет уровни гарантии секретности – EAL (Evaluation Assurance Level). Common Criteria оценивает не только безопасность и надежность продуктов, но и процессы их разработки и поддержки, что гарантирует достаточно быстрое решение проблем безопасности.. Выделены семь EAL-уровней гарантии секретности:

• EAL1 («Functionally tested»). Этот уровень применим там, где необходима минимальная конфиденциальность, но обеспечение секретности не рассматривается как важное требование.

• EAL2 («Structurally tested»). Применим там, где от системы требуется средний уровень гарантированной секретности в отсутствие полной информации обо всех процедурах разработки.

• EAL3 («Methodically tested and checked»). Применим там, где разработчики или пользователи требуют среднего уровня гарантированной секретности и исчерпывающего исследования операционной системы и этапов ее разработки, не прибегая к существенной переработке ОС.

• EAL4 («Methodically designed, tested and reviewed»). Применим там, где разработчики или пользователи требуют высокого уровня гарантированной секретности операционной системы и специальной доработки уже существующей ОС для обеспечения этих требований.

• EAL5 («Semi formally designed and tested»). Применим там, где разработчики или пользователи требуют высокого уровня гарантированной секретности операционной системы и строгого подхода к проектированию, так чтобы эти свойства были заложены уже на этапе проектирования, используя специальные средства обеспечения секретности.

• EAL6 («Semi formally verified, designed and tested»). Применим там, где существует высокий уровень опасных ситуаций и где оправданны высокие затраты на защиту от несанкционированного доступа.

• EAL7 («Formally verified, designed and tested»). Этот уровень должен применяться в приложениях с очень высокой ценой несанкционированного доступа.

• MILS (Multiple Independent Levels of Security/Safety) делает возможной математическую верификацию программного ядра системы путем уменьшения функциональности за счет предъявления к системам четырех обязательных групп требований (Information Flow, Data Isolation, Period Processing, Damage Limitation). Развивается проект усилиями заинтересованных компаний и организаций, таких, как U.S. Air Force Research Laboratory, Lockheed Martin, Агентство национальной безопасности США и др. (http://mils.ois.com). MILS-архитектура представляет собой систему с изолированными разделами, каждый из которых включает ядро, middleware и приложение.

• POSIX (Portable Operating System interface for unIX) определяет переносимый интерфейс операционных систем на уровне исходных текстов (www.pasc.org). Основная спецификация разработана как спецификация IEEE 1003.1 и одобрена в качестве международного стандарта ISO/IEC 9945-1:1990. С точки зрения ОС наибольший интерес представляют три стандарта: 1003.1a (OS Definition), 1003.1b (Realtime Extensions) и 1003.1c (Threads).

7.2. Концепция изолированных разделов

По многим аспектам указанные выше документы являются взаимно перекрывающимися и дополняющими друг друга. В результате многочисленных исследований в качестве основной была принята концепция изолированных разделов. Разделы (partition) должны жестко изолироваться друг от друга с точки зрения используемого процессорного времени и оперативной памяти. Удовлетворение требованиям жесткой изоляции разделов должно быть доказано поставщиками программных решений в соответствии с методологией сертификации, изложенной в DO-178B. Хотя существуют различные подходы к реализации изолированных разделов[2], в настоящее время принята архитектура, соответствующая ARINC 653. Спецификация ARINC 653 определяет поддержку изолирования для ОС и в качестве языка описания использует языки программирования Си и Ada-95.

С точки зрения пользователя, ARINC 653 представляет собой спецификацию интерфейса (APEX – Application Executive) и при этом не определяет то, как должен быть реализован этот интерфейс. Например, некоторые поставщики программных средств реализуют диспетчеризацию (в соответствии с их «адаптацией» ARINC 653) с помощью одно-

2. Commercial Off-The-Self Real-Time Operating System and Architectural Consideration. Final Report, U.S. Federal Aviation Administration, DOT/FAA/AR-03/77, February 2004.

уровневого диспетчера, другие – с помощью двухуровневого. Первый управляет разделами, второй – процессами внутри каждого раздела. Такая ситуация с поддержкой ARINC 653 влечет за собой значительные трудности при сертификации программных продуктов, соответствующих только ARINC 653, так как один и тот же API может отображаться на различные подходы к реализации интерфейса. Как следствие, появилось множество существенно различающихся ОС, которые, однако, соответствуют спецификации ARINC 653.

Все данные и программный код в каждом разделе (partition) компонуются вместе и выполняются в пользовательском режиме. Компоненты MOS (Module Operating System) и BSP (Board Support Package) работают в супервизорном (Supervisor) режиме. Дополнительно может существовать один специальный раздел с некоторыми специальными возможностями, такими, как средства ввода-вывода и переключения режимов.

ARINC 653 как раз и задает интерфейс для обмена информацией между разделами, каждый из которых представляет собой некое приложение плюс POS (Partition Operating System). Этот интерфейс должен гарантировать изоляцию следующих элементов друг от друга.

• Оперативная память. Каждому разделу выделяется непрерывное линейное физическое адресное пространство, и границы этого пространства не могут меняться в процессе работы системы. Для каждого раздела выделение (из этого жестко определенного адресного пространства) и освобождение памяти выполняется и контролируется MOS. ОС должна обеспечивать изолирование информации в адресном пространстве внутри выделенного каждому разделу линейного куска. Это относится к программному коду, константам, статическим данным, стеку, «куче» (heap – область, откуда берется память по запросу allocate).

• Время использования процессора. Фундаментальной концепцией ARINC 653 является то, что процессы в одном разделе либо не должны влиять на поведение процессов в другом разделе совсем, либо могут влиять на них заранее определенным и контролируемым способом (например, путем установки некоторых флагов или условий). ARINC 653 требует, чтобы процессорное время выделялось каждому разделу строго циклически (round-robin), причем время владения процессором при этом должно задаваться заранее в конфигурационной таблице. Внутри того или иного раздела процессы (нити) могут конкурировать между собой за процессорное время на основе приоритетов («вытеснять» друг друга), используя диспетчеризацию с вытеснением по приоритетам.

• Программный код. Для некоторых областей памяти MOS устанавливает (в режиме Supervisor) атрибут «только выполнение». Это означает, что приложения в пользовательском режиме не могут разрушить область кода.

• Прерывания. Прерывания являются асинхронными событиями, которые требуют особой внимательности к вопросам обеспечения изолированности разделов. Важно, чтобы они не «посягали» на время и память в другом разделе. Одни из возможных источников прерывания – таймеры, которые управляют диспетчеризацией событий как внут-

ри POS, так и внутри MOS. Для обеспечения изоляции прерываний приняты следующие соглашения:

а) прерывания таймера возникают только в супервизорном режиме. Прямой доступ к часам в пользовательском режиме невозможен;

б) если POS и MOS находятся на различных уровнях защиты, то должен обеспечиваться механизм распространения информации о временных событиях в прикладные разделы, работающие в пользовательском режиме;

в) в те моменты, когда раздел активен (владеет процессорным временем), ему должны передаваться все относящиеся к нему временные события;

г) когда раздел не активен, то все относящиеся к нему временные события должны сохраняться и затем при активизации передаваться ему.

Другим источником прерываний являются внешние устройства ввода-вывода. В соответствии с требованиями ARINC 653 для устройств с синхронной передачей данных рекомендуется использовать алгоритм опроса устройства. Однако некоторые устройства требуют обработки прерываний. Эти прерывания должны обрабатываться MOS и затем передаваться в POS в момент активизации раздела. Очевидно, что при этом могут возникать временные задержки и даже потеря информации, и, следовательно, разработчик должен это учитывать.

ARINC 653 определяет не только требования по изолированию разделов, но и механизмы взаимодействия между ними. Введены следующие функции взаимодействия между разделами:

• обмен сообщениями с помощью команд send/receive путем установления канала связи, который должен быть заранее описан в конфигурационной таблице системы;

• обмен через буфер. При этом только один раздел может писать в конкретный буфер, а все другие – только читать. Это обеспечивает возможность широковещательного обмена информацией между разделами.

7.3. Стандарт DO-178B

Стандарт DO-178B описывает технику и методы, ориентированные на то, чтобы гарантировать целостность и надежность ПО. Он также охватывает все этапы жизненного цикла программного обеспечения: планирование, разработку требований, проектирование, кодирование, интеграцию и тестирование. DO-178B требует от разработчика ПО предъявления следующих данных (и, следовательно, выполнения соответствующих действий), относящихся к ПО.

Планирование должно включать следующие планы: план для программных аспектов сертификации (PSAC), план программного проекта (SDP), план верификации ПО (SVP), план управления конфигурацией ПО (SCMP) и план обеспечения качества ПО (SQAP). На протяжении всего жизненного цикла ПО должно быть обеспечено соответствие сле-

дующим стандартам к ПО: формулировке требований, проектированию, кодированию, верификации и документированию. В соответствии с приведенными стандартами должны быть разработаны такие компоненты, как требования к ПО (требования высокого уровня) – SRD, проект ПО – SDD (требования и архитектура), программный код в исходном и объектном виде. В соответствии с DO-178B каждый из разработанных компонентов должен быть верифицирован по разнообразным критериям. Верификация требований высокого уровня к ПО включает в себя проверку следующих моментов: соответствуют ли они системным требованиям и доступны ли для анализа вместе с ними; являются ли они точными и согласованными; совместимы ли с аппаратными средствами, что должно быть подтверждено результатами тестирования. Верификация проекта программного обеспечения предусматривает проверку требований к проекту ПО на их соответствие требованиям высокого уровня (SRD), на возможность проведения их анализа, на точность и согласованность, что должно быть подтверждено результатами тестирования. Аналогично должны быть верифицированы архитектура и код ПО, которые, кроме того, должны соответствовать стандартам. DO-178B определяет процесс верификации этапа интеграции программного обеспечения, проверку полноты самого процесса верификации, обеспечение менеджмента конфигураций (в том числе управление версиями), квалификацию автоматизированных средств. В DO-178B определены три уровня структурного тестирования кода:

1. SC (Statement Coverage) – покрытие утверждений. Означает, что каждое утверждение в программе было вызвано хотя бы один раз.

2. DC (Decision Coverage) – покрытие решений. Означает, что каждая точка входа и выхода в программе была выполнена хотя бы один раз и что каждое решение в программе принимало все возможные (булевские) выходные значения хотя бы один раз.

3. MCDC (Modified Condition Decision Coverage) – покрытие решений модифицируемым условием. Это означает, что каждая точка входа и выхода в программе была выполнена хотя бы один раз, что каждое решение в программе принимало все возможные (булевские) выходные значения хотя бы один раз и что каждое условие в решении приводило к независимому изменению выходного значения.

Определенные в DO-178B уровни сертификации различаются количеством требований, которым должно удовлетворять ПО (т. е. глубиной верификации). Например, при верификации кода в зависимости от уровня сертификации должны удовлетворяться следующие критерии покрытия кода:

Теоретические аспекты верификации кода изложены выше.

Сертификация на соответствие DO-178B должна проводиться назначаемыми инженерными представителями (Designated Engineering Representative, DER), которые назначаются FAA для проверки данных, используемых при сертификации. DER – это опытные и независимые специалисты, которых обычно привлекают к процессу сертификации ПО с начальных этапов сертификации.

7.4. Реализация требований к безопасности ОС. Подсистема администрирования ОС Windows XP

К штатным средствам администрирования ОС можно отнести Microsoft Management Console (консоль управления MMC), Windows Script Host (сервер сценариев Windows), Windows Management Instrumentation (инструментарий управления Windows), Resultant Set of Policy (набор результирующих политик), Remote Assistance (удаленный помощник), Group Policy (групповые политики).

Рассмотрим подробнее каждую из этих технологий.

Microsoft Management Console (MMC)

Консоль управления MMC (Microsoft Management Console) – компонент, предоставляющий единую среду для всех административных программ. Она является как бы стержнем, на который «нанизываются» все программы администрирования.

В окнах MMC загружаются управляющие программы, называемые оснастками, каждая из которых выполняет определенную административную роль. Оснастки организованы в виде древовидной структуры, и все вместе образуют полный комплект административных инструментов. Одновременно администратор может загрузить несколько необходимых ему оснасток и управлять системой из одной консоли, что избавляет его от загрузки разных программ и постоянного переключения между ними.

Windows Script Host (WSH)

Windows Script Host – автономный сервер сценариев, позволяющий выполнять специальные сценарии для операционной системы Windows. Сценарии WSH могут создаваться с помощью специализированных языков (например, VBScript или Microsoft JScript) и использовать любые объекты ActiveX, зарегистрированные в системе.

Собственная объектная модель WSH позволяет из сценариев работать с файловой системой, системным реестром, специальными папками и ярлыками Windows, ресурсами локальной сети, а также запускать процессы и контролировать ход их выполнения.

Сценарии WSH позволяют организовать взаимодействие с разработанными Microsoft современными ActiveX технологиями:

• ActiveX Data Object (ADO) – доступ к базам данных разных форматов;

• Active Directory Service Interface (ADSI) – работа со службами каталогов (Active Directory для Windows 2000/XP, Windows Directory Service для Windows NT 4.0);

• Windows Management Instrumentation (WMI) – инструментарий управления операционной системой Windows.

Windows Management Instrumentation (WMI)

Windows Management Instrumentation – это глобальная концепция настройки, управления и слежения за работой различных частей корпоративной компьютерной сети. Используя WMI, можно контролировать и изменять параметры самых разнородных физических и логических

элементов компьютерной системы, в качестве которых могут выступать, например, файл на жестком диске, запущенный экземпляр приложения, системное событие, сетевой пакет или установленный в компьютере процессор.

Для доступа ко всем элементам используется единый интерфейс, который позволяет управлять компьютером в сети дистанционно (если на локальной и удаленной машине установлен WMI и у пользователя имеются соответствующие права). Управление осуществляется при помощи CIMOM – Common Information Model Object Manager – базы данных объектов, представляющих эти элементы. Это позволяет, в частности, быстро получать информацию разнообразного типа об объектах с помощью запросов на языке SQL.

Resultant Set of Policy (RSoP)

Resultant Set of Policy – представляет из себя расширение к встроенной в Windows XP Group Policy с использованием WMI, и позволяет администратору определять и анализировать текущее множество политик, применяемых к выбранному пользователю или компьютеру, а также просматривать установки текущей политики на заданных компьютерах.

Remote Assistance (RA)

Remote Assistance – представляет собой технологию, которая позволяет пользователям, образно говоря, «помочь друг другу», используя соединение по локальной сети. С этим инструментом, один пользователь, называемый «Эксперт», может просмотреть рабочий стол другого пользователя, называемого «Новичок». С разрешения «Новичка», «Эксперт» может в равной доле контролировать его компьютер удаленно. После подключения к удаленному компьютеру «Эксперт» может видеть экран удаленного компьютера и обмениваться с удаленным пользователем текстовыми сообщениями в реальном времени. С позволения пользователя удаленного компьютера «Эксперт» может получить доступ к мыши и клавиатуре удаленного компьютера.

Group Policy

Group Policy (групповые политики) – предоставляет возможность для администрирования на основе политик с использованием Microsoft Active Directory. Администрирование с применением набора политик, предоставляет администратору следующие возможности:

• Стандартный набор политик.

• Установки безопасности. Применяются для локальных компьютеров, доменов и сетей.

• Установка приложений.

Скрипты. Применяются при старте и завершении работы операционной системы, и при входе-выходе пользователя.

Администратор может управлять установками групповых политик непосредственно со своего рабочего места.

Операционная система Windows XP обеспечивает безопасность, используя следующие механизмы:

• аутентификация: процесс проверки подлинности субъекта доступа;

• авторизация: процесс, позволяющий контролировать доступ субъекта к разнообразным сетевым ресурсам, таким как файлы, папки и принтеры.

Как следует из изложенного выше, процесс аутентификации является неотъемлемой частью нормального функционирования операционной системы.

При интерактивной регистрации в системе (в отличие от регистрации пользователя через сеть) происходит взаимодействие с процессами Winlogon, Lsass, одним или несколькими пакетами аутентификации, а также SAM или Active Directory. В данном случае мы понимаем термин «регистрация в системе» как синоним «идентификация и аутентификация».

Пакеты аутентификации (authentication packages) – это DLL-модули, выполняющие проверки, связанные с аутентификацией. Пакетом аутентификации для интерактивной регистрации в домене является Kerberos, аналогичным пакетом для интерактивной регистрации на локальных компьютерах, доменного входа в доверяемые домены под управлением версии Windows до Windows 2000, а также для регистрации в отсутствие контроллера домена – MSV1_0.

Winlogon – системный процесс, отвечающий за управление взаимодействием с пользователем в связи с защитой. Он координирует регистрацию, запускает пользовательскую оболочку для входа в систему, обрабатывает выход из системы и управляет множеством других операций, имеющих отношение к защите, – вводом паролей при регистрации, сменой паролей, блокированием и разблокированием рабочих станций и т. д. Процесс Winlogon должен обеспечить невидимость операций, связанных с защитой, другим активным процессам. Так, Winlogon гарантирует, что в ходе этих операций недоверенный процесс не сможет перехватить управление рабочим столом и таким образом получить доступ к паролю.

Winlogon получает имя и пароль пользователя через Graphical Identification and Authentication (GINA) DLL. Стандартная GINA – \Winnt\System32\Msgina.dll. Msgina выводит диалоговое окно для входа в систему. Позволяя заменять Msgina другими GINA-библиотеками, Windows дает возможность менять механизмы идентификации пользователей. Например, сторонний разработчик может создать GINA для поддержки устройства распознавания отпечатков пальцев и для выборки паролей пользователей из зашифрованной базы данных.

Winlogon – единственный процесс, который перехватывает запросы на регистрацию с клавиатуры. Получив имя и пароль пользователя от GINA, Winlogon вызывает Lsass для аутентификации этого пользователя. Если аутентификация прошла успешно, процесс Winlogon активизирует оболочку. Схема взаимодействия между компонентами, участвующими в процессе регистрации, показана на ***рис.1***.

Winlogon не только поддерживает альтернативные GINA, но и может загружать дополнительные DLL компонентов доступа к сетям, необходимые для вторичной аутентификации. Это позволяет сразу нескольким

компонентам сетевого доступа получать идентификационные и регистрационные данные в процессе обычного входа пользователя в систему. Регистрируясь в системе под управлением Windows XP, пользователь может одновременно зарегистрироваться и на UNIX-сервере. После этого он получит доступ к ресурсам UNIX-сервера с компьютера под управлением Windows XP без дополнительной аутентификации. Эта функциональность является одной из форм унифицированной регистрации (single sign-on).

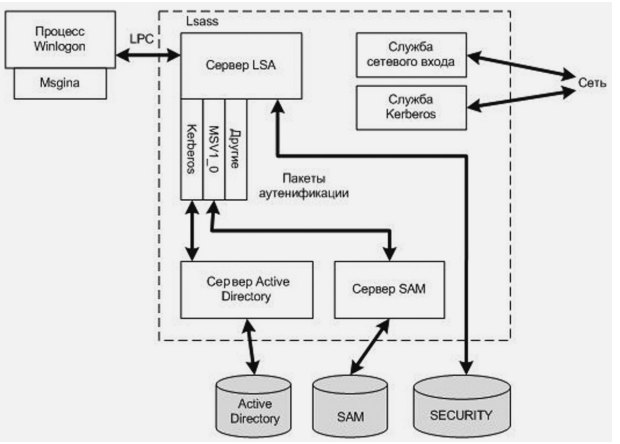


Рис.1. Компоненты Windows XP, участвующие в регистрации

При инициализации системы, когда ни одно пользовательское приложение еще не активно, Winlogon выполняет ряд операций, обеспечивающих ему контроль над рабочей станцией с момента готовности системы к взаимодействию с пользователем.

1. Создает и открывает интерактивный объект WindowStation, \Windows\WinSta0, представляющий клавиатуру, мышь и монитор. Далее создает дескриптор защиты станции с одним ACE, содержащим только SID-идентификатор Winlogon. Этот уникальный дескриптор безопасности гарантирует, что другой процесс получит доступ к рабочей станции, только если Winlogon явно разрешит это.

2. Создает и открывает три объекта «рабочий стол»: для приложений (\Windows\WinSta0\Default), Winlogon (\Windows\WinSta0\Winlogon) и экранной заставки (\Windows\WinSta0\ScreenSaver). Защита объекта «рабочий стол» Winlogon организуется так, чтобы к нему мог обращаться только Winlogon. Другие объекты «рабочий стол» доступны как Winlogon, так и пользователям. Следовательно, пока активен объект «рабочий стол» Winlogon, никакой другой процесс, относящийся к другому рабочему столу, не получает доступа к очереди клавиатурных (мыши) сообщений, буферу обмена (clipboard) и т.п., сопоставленным с этим рабочим столом. Эта функциональность используется Windows

XP для защиты операций, требующих передачи паролей, а также для блокировки и разблокировки рабочего стола.

До регистрации какого-либо пользователя на компьютере видимым рабочим столом является объект «рабочий стол» Winlogon. После входа в систему нажатие клавиш Ctrl+Alt+Del вызывает переключение объектов «рабочий стол» – с Default на Winlogon. (Это объясняет, почему после нажатия Ctrl+Alt+Del с рабочего стола исчезают все окна и почему они возвращаются, как только закрывается диалоговое окно Windows Security.) Таким образом, SAS всегда активизирует защищенный объект «рабочий стол», контролируемый Winlogon.

3. Устанавливает LPC-сосдинение с Lsass через порт LsaAuthenticationPort (вызовом *LsaRegisterLogonProcess).* Это соединение понадобится для обмена информацией при входе и выходе пользователя из системы и при операциях с паролем.

Далее Winlogon настраивает оконную среду.

4. Инициализирует и регистрирует структуру данных оконного класса, которая сопоставляет процедуру Winlogon с создаваемым ею окном.

5. Регистрирует SAS, сопоставляя ее с только что созданным окном. Это гарантирует, что ввод пользователем SAS будет вызывать именно оконную процедуру Winlogon и что программы типа троянских коней не смогут перехватывать управление при вводе SAS.

6. Регистрирует окно, чтобы при выходе пользователя вызывалась процедура, сопоставленная с этим окном. Подсистема Win32 проверяет, что запросивший уведомление процесс является именно Winlogon.

Как только при инициализации системы создается рабочий стол Winlogon, он становится активным рабочим столом. Причем активный рабочий стол Winlogon всегда заблокирован. Winlogon разблокирует свой рабочий стол лишь для переключения на рабочий стол приложений или экранной заставки. (Блокировать или разблокировать рабочий стол может только процесс Winlogon.)

Регистрация начинается, когда пользователь нажимает комбинацию клавиш SAS (по умолчанию – Ctrl+Alt+Del). После этого Winlogon вызывает GINA, чтобы получить имя и пароль пользователя. Winlogon также создает уникальную локальную группу для этого пользователя и назначает ее данному экземпляру объекта «рабочий стол» (который представляет клавиатуру, экран и мышь). Winlogon передает имя этой группы в Lsass при вызове LsaLogonUser. Если регистрация пользователя прошла успешно, эта группа будет включена в маркер процесса регистрации (logon process marker) – такой шаг предпринимается для защиты доступа к объекту «рабочий стол». Например, второй пользователь, зарегистрировавшийся по той же учетной записи, но в другой системе, не получит доступа для записи к объекту «рабочий стол» первого пользователя, так как не входит в его группу.

После ввода имени и пароля пользователя Winlogon поочередно вызывает все пакеты аутентификации, зарегистрированные в системе. Эти пакеты перечисляются в разделе реестра

HKLM\SYSTEM\CurrenlControlSet\Control\Lsa. Winlogon получает описатель пакета вызовом Lsass-функции LsaLookupAuthenticationPackage и передает пакету регистрационные данные через LsalogonUser. После того как пакет аутентифицирует пользователя, Winlogon продолжает процесс регистрации этого пользователя. Если не один из пакетов не сообщает об успешной аутентификации, регистрация прекращается.

Windows XP использует два стандартных пакета аутентификации: Kerberos и MSV1_0. Пакет аутентификации по умолчанию в автономной системе Windows – MSV1_0 (\Winnt\System32\Msvl_0.dll); он реализует протокол LAN Manager 2. Lsass также использует MSV1_0 на компьютерах, входящих в домен, чтобы аутентифицировать домены и компьютеры под управлением версий Windows до Windows 2000, не способные найти контроллер домена для аутентификации. (Отключенные от сети портативные компьютеры относятся к той же категории.) Пакет аутентификации Kerberos (\Winnt\System32\Kerberos.dll) используется на компьютерах, входящих в домены Windows 2000 или Windows Server 2003. Этот пакет во взаимодействии со службами Kerberos, выполняемыми на контроллере домена, поддерживает протокол Kerberos версии 5 (ревизии 6). Этот протокол определен в RFC 1510.

Пакет аутентификации MSV1_0 принимает имя пользователя и хешированную версию пароля и посылает локальному SAM запрос на получение информации из учетной записи, включая пароль, группы, в которые входит пользователь, и список ограничений по данной учетной записи. Сначала MSV1_0 проверяет ограничения, например, разрешенное время или типы доступа. Если ограничения из базы данных SAM запрещают регистрацию пользователя в это время суток, MSV1_0 возвращает LSA статус отказа.

Далее MSV1_0 сравнивает хешированный пароль и имя пользователя с теми, которые хранятся в SAM. В случае кэшированной доменной регистрации MSV1_0 обращается к кэшированной информации через функции Lsass, отвечающие за сохранение и получение «секретов» из базы данных LSA (куст реестра SECURITY). Если эти данные совпадают, MSV1_0 генерирует LUID сеанса регистрации и создает собственно сеанс регистрации вызовом Lsass. При этом MSV1_0 сопоставляет данный уникальный идентификатор с сеансом и передает данные, необходимые для того, чтобы, в конечном счете, создать маркер доступа для пользователя. Отметим, что маркер доступа включает SID пользователя, SID группы и информацию из профиля пользователя, например, начальный каталог.

Если MSV1_0 требуется аутентифицировать пользователя с удаленной системы, например при его регистрации в доверяемом домене под управлением версий Windows до Windows 2000, то MSV1_0 взаимодействует с экземпляром Netlogon в удаленной системе через службу Net Logon (сетевого входа в систему). Netlogon в удаленной системе взаимодействует с пакетом аутентификации MSVl_0 этой системы, передавая результаты аутентификации системе, в которой проходит регистрация.

Базовая последовательность действий при аутентификации Kerberos в основном та же, что и в случае MSV1_0. Однако в большинстве слу-

чаев доменная регистрация проходит на рабочих станциях или серверах, входящих в домен (а не на контроллере домена), поэтому пакет в процессе аутентификации должен взаимодействовать с ними через сеть. Взаимодействие этого пакета со службой Kerberos на контроллере домена осуществляется через TCP/IP-порт Kerberos (88).

Служба Kerberos (\Winnt\System32\Kdcsvc.dll), реализующая протокол аутентификации Kerberos, выполняется в процессе Lsass на контроллерах домена.

После проверки хешированной информации об имени и пароле пользователя с помощью объектов учетных записей пользователей (user account objects) Active Directory (через сервер Active Directory, \Winnt\System32\Ntdsa.dll) Kdcsvc возвращает удостоверения домена Lsass, который при успешной регистрации передает через сеть результат аутентификации и удостоверения доменной регистрации пользователя той системе, где проходит регистрация.

ПРИМЕЧАНИЕ. Приведенное здесь описание аутентификации пакетом Kerberos сильно упрощено, и, тем не менее, оно иллюстрирует роль различных компонентов в этом процессе.

Как только регистрационные данные аутентифицированы, Lsass ищет в базе данных локальной политики разрешенный пользователю тип доступа – интерактивный, сетевой или сервисный. Если тип запрошенного входа в систему не соответствует разрешенному, регистрация прекращается. Lsass удаляет только что созданный сеанс регистрации, освобождая его структуры данных, и сообщает Winlogon о неудачной регистрации. Winlogon в свою очередь сообщает об этом пользователю. Если же запрошенный тип входа в систему разрешается, Lsass добавляет любые дополнительные идентификаторы защиты (SID), например, Everyone, Interactive и т.п. Затем он проверяет в своей базе данных привилегии, назначенные всем идентификаторам данного пользователя, и включает эти привилегии в маркер доступа пользователя.

Собрав всю необходимую информацию, Lsass вызывает исполнительную систему для создания маркера доступа. Исполнительная система создает основной маркер доступа для интерактивного или сервисного сеанса и маркер олицетворения для сетевого сеанса. После успешного создания маркера доступа Lsass дублирует его, создавая описатель, который может быть передан Winlogon, а свой описатель закрывает. Если нужно, проводится аудит регистрации. На этом этапе Lsass сообщает Winlogon об успешной регистрации и возвращает описатель маркера доступа, LUID сеанса регистрации и информацию из профиля, полученную от пакета аутентификации (если она есть).

Далее Winlogon просматривает параметр реестра HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon\Userinit и создает процесс для запуска программ, указанных в строковом значении этого параметра (там могут присутствовать имена нескольких EXE-файлов, разделенные запятыми). Значение этого параметра по умолчанию приводит к запуску Userinit.exe, который загружает профиль пользователя, а затем создает процесс для запуска программ, перечисленных в

HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon\Shell (значение этого параметра по умолчанию – Explorer.exe). После этого Userinit завершается – вот почему команды типа tlist/t никогда не сообщают о родительском процессе Explorer.exe.

В ОС Windows XP появилось средство управления аттестатами пользователей. Это средство позволяет управлять и использовать различную информацию из аттестата пользователя, такую как имя пользователя и его пароль. Существуют два типа аттестатов: доменные аттестаты и общие аттестаты.

Диспетчер аттестатов (Credential Manager) является составляющей частью службы аутентификации пользователей. При установке операционной системы вместе с установкой службы аутентификации пользователей автоматически устанавливается такой компонент, как диспетчер аттестатов.

Если на компьютер в сети приходит запрос на аутентификацию с использованием протокола NTLM или Kerberos, в окне «Сетевая идентификация пользователей» появляются дополнительные флаги «Обновить стандартный аттестат» и «Сохранить пароль». Если пользователь выбирает соответствующий флаг, диспетчер аттестатов сохраняет имя пользователя, его пароль, а также всю информацию, которая каким-либо образом связана с процессом аутентификации этого пользователя. При следующем использовании этой службы диспетчер аттестатов автоматически использует данные последнего вошедшего в систему пользователя. В случае отказа использовать указанные данные, система снова запрашивает все данные необходимые для входа в систему. Если процесс аутентификации при этом прошел успешно, диспетчер аттестатов перезаписывает последний использованный аттестат на новый.

Архитектура системы построена таким образом, что для каждой службы аутентификации существует вполне определенное ограниченное пространство для хранения аттестатов. По этой причине, если разные компьютеры в сети используют одну и ту же службу аутентификации пользователей, использование диспетчера аттестатов нежелательно.

Стандартные аттестаты, которые будут использоваться по умолчанию, можно указать в диалоговом окне «Идентификация», доступ к которому можно получить через панель управления. Операционная система позволяет вводить самые разнообразные комбинации имени домена, имени пользователя и пароля. Если процесс ввода аттестатов был произведен успешно, они будут автоматически использоваться только в том случае, если был выставлен флаг «Сохранять пароль». Если какой-то из аттестатов окажется некорректным или в нем не будет доставать каких-либо значений, необходимых для корректной аутентификации, пользователю будет предложено ввести недостающие поля.

Хотя и для шифрования данных, необходимых для аутентификации, используется функция CryptProtectData, присутствие этих данных на компьютере сетевого клиента даже в зашифрованном виде представляет собой потенциальную угрозу безопасности. Запретить использование диспетчера аттестатов можно путем изменения в реестре системы клю-

ча DisallowSavedNetworkPasswords по адресу HKEY_LOCAL_MACHINE\Comm\Security на значение 1. После этой процедуры флаг «Использовать диспетчер сертификатов» хотя и будет появляться, но будет запрещенным для использования.

Помимо всего этого менеджер сертификатов имеет свой собственный API, который состоит из трех функций: чтения, записи и уничтожения информации, необходимой для аутентификации, из различных частей системного реестра.

Доменные аттестаты используются компонентами операционной системы и аутентифицируются с помощью LSA (Local Security Authority). Как правило, доменные аттестаты пользователя создаются, когда пакет безопасности (security package), такой как Kerberos, осуществляет аутентификацию пользователя. Аттестаты, полученные при входе пользователя в систему, кэшируются ОС, чтобы не требовать идентификационных данных пользователя каждый раз при попытке его доступа к различным ресурсам. Например, сетевое подключение к другому компьютеру осуществляется прозрачно (без дополнительного запроса идентификационных данных) на основе доменного аттестата.

ОС содержит средства, защищающие секретную часть аттестата (например, пароль). Только компоненты ОС, работающие в контексте LSA, могут считывать данные из доменных аттестатов. Все остальные компоненты могут лишь осуществлять их запись.

Общие аттестаты

Общие аттестаты создаются и проверяются приложениями, которые самостоятельно управляют идентификацией и авторизацией пользователя, не поручая эти задачи операционной системе. Например, приложение может потребовать у пользователя ввести идентификационные данные, чтобы создать сертификат для доступа к Web-сайту.

Аутентификация с помощью LSA (далее LSA-аутентификация) позволяет использовать такие средства аутентификации как GINA (Graphical Identification and Authentication), пакеты аутентификации (authentication package) или комбинированный провайдер безопасности/пакет аутентификации (SSP/AP).

Модель LSA-аутентификации обладает следующими свойствами:

• LSA-аутентификация поддерживает различные пакеты аутентификации, в том числе не входящие штатно в поставку ОС. Такой подход позволяет использовать различные виды идентификационных данных вместо стандартных имени пользователя и пароля.

• LSA поддерживает различные пакеты безопасности, которые функционируют как провайдеры безопасности для распределенных приложений. Все пакеты безопасности реализуют однотипный интерфейс (SSPI), который позволяет использовать как штатные пакеты безопасности, так и пакеты сторонних производителей.

• LSA поддерживает управление аттестатами пользователей в гетерогенных средах, то есть средах, построенных с использованием продуктов, отличных от продуктов Microsoft.

• Каждый класс устройств входа в систему работает в своем собственном процессе. Классы устройств обычно включают в себя такие уст-

ройства как считыватели смарт-карт. В том числе, сетевые подключения тоже рассматриваются как отдельные устройства.

Сессия пользователя начинается после его успешной идентификации.

Когда пользователь прошел процедуру идентификации и аутентификации, пакет аутентификации создает сессию пользователя (logon session) и использует LSA, который создает токен (token) для пользователя.

Когда токен создан, его счетчик ссылок увеличивается на единицу. Кроме того, этот счетчик увеличивается при создании копии токена и имперсонации. Когда использование копии завершено, счетчик уменьшается на единицу. При обнулении счетчика токен пользователя удаляется и сессия пользователя завершается.

Пакеты аутентификации – это вспомогательные модули, которые используются LSA, чтобы поддерживать различные способы входа в систему и различные протоколы безопасности.

Пакет аутентификации отвечает за выполнение следующих задач.

• Анализ идентификационных данных пользователя, чтобы разрешить/отклонить вход пользователя в систему.

• Создание новой сессии пользователя.

• Передача информации LSA.

Одним из способов получения пользователем доступа к средствам ОС Windows XP является интерактивный вход в систему. Он осуществляется непосредственным вводом имени пользователя и пароля или предъявлением какой-либо дополнительной идентификационной информации (например, смарт-карты).

Интерактивный вход пользователя осуществляется с помощью Winlogon, GINA и сетевых провайдеров.

Winlogon – это системный процесс, который полностью контролирует интерактивную идентификацию и аутентификацию пользователей. Winlogon обеспечивает процедуру аутентификации, независимую от конкретной модели идентификации пользователя (по паролю, ключу, смарт-карте и т.д.).

GINA – это динамическая библиотека, работающая в контексте процесса Winlogon и обеспечивающая процесс идентификации пользователя. GINA может показывать диалоги для ввода имени пользователя и пароля, считывать идентификационную информацию из дополнительных устройств (например, считывателей смарт-карт) и т.д. GINA является заменяемой частью ОС. Штатно в Windows XP содержится msgina.dll, показывающая привычные диалоги приветствия, но она может быть заменена на любую другую.

Для сетевой аутентификации пользователей в операционной системе Windows XP используются следующие защищенные протоколы:

• Kerberos v5;

• RADIUS;

• NTLM.

Протокол Kerberos определяет, как пользователь взаимодействует с сетевой службой аутентификации. Сетевые клиенты получают специ-

альные билеты (сетевые аттестаты) в центрах распространения ключей Kerberos и предоставляют их по запросу сервера при установлении соединения. Протокол Kerberos предоставляет механизмы взаимной аутентификации пользователей до того, как будет установлено защищенное соединение. Использование рассматриваемого протокола предполагает, что сетевые транзакции между клиентами и серверами имеют место в сети, которая не имеет физической защиты, то есть сетевые пакеты вполне могут быть отслежены, перехвачены или изменены. По этой причине данный протокол позволяет контролировать целостность сетевых пакетов, а также идентифицировать личность отправителя.

RADIUS (Remote Authentication Dial-In User Service) – промышленный стандарт, позволяющий производить безопасную аутентификацию пользователей при установке dial-up и других сетевых соединений. Суть протокола заключается в том, что обычно сетевые серверы (например, dial-up) являются клиентами общей сети аутентификации RADIUS. При попытке клиента зарегистрироваться в сети, сетевой сервер передает информацию о пользователе серверу RADIUS, а он уже в свою очередь берет на себя все обязательства по проверке этой информации и принятию решения о регистрации пользователя в сети.

Сетевой защищенный протокол NTLM (Windows NT Challenge/Response) используется при установке защищенных сетевых соединений между компьютерами с операционными системами Microsoft. Аттестаты NTLM основываются на данных, полученных в процессе интерактивного входа в систему, и содержит в себе такие данные, как имя домена, имя пользователя и хеш пользовательского пароля. NTLM использует специальный защищенный протокол передачи данных для того, чтобы избежать передачи пароля пользователя по сети в открытом виде.

7.5. Реализация требований к безопасности ОС. Подсистема разграничения доступа к объектам Windows XP

В ОС Microsoft Windows XP объект является отдельной переменной заранее определенного типа. Определение типа объекта состоит из регистрации набора внутренних данных, операций для работы с объектами этого типа и набора атрибутов объекта.

Внутренние данные базируются на более низкоуровневых объектах, поддерживаемых ядром ОС Microsoft Windows XP, и служат для хранения основных свойств объекта.

Атрибуты объекта – это поля данных в типе, которые определяют дополнительные, часто изменяющиеся свойства создаваемого объекта. Чаще всего атрибуты объекта используются в качестве параметров, передаваемых соответствующим функциям Win32 API или исполнительной системы (XP executive) при создании объекта. Объект типа «Стек», например, должен иметь указатель на область памяти, где размещаются его элементы. Этот указатель и является атрибутом стека, передаваемым функции создания объекта (конструктору).

Операции для работы с объектом – это набор сервисных процедур, предоставляемых для того, чтобы пользователь мог производить какие-либо действия с объектом, обычно состоящие в получении или изменении атрибутов объекта. Например, для стека операция Push может изменить значение указателя на область памяти, где хранятся его элементы.

Наиболее фундаментальное различие между объектом и структурированными данными – это наличие у объектов функций и данных, недоступных для внешнего использования. Для работы с объектом можно использовать только сервисные функции, предоставляемые для этого типа, прямое чтение или изменение внутренних данных исключено. Такой набор функций, доступных к внешнему использованию, является разделителем между внутренней организацией объекта и его представлением для пользователей. Таким образом, разработчики объекта могут изменять его внутреннюю организацию, не затрагивая интересов пользователей.

В силу изложенных причин, группой разработчиков XP executive было решено использовать объекты для представления системных ресурсов ОС Microsoft Windows XP. При этом учитывался и тот факт, что использование объектов позволяет эффективно решать основные задачи, возлагаемые на современную операционную систему:

• возможность использования читаемых названий для обозначения системных ресурсов;

• совместное использование ресурсов разными процессами;

• защита системных ресурсов от несанкционированного использования.

Таким образом, в ОС Microsoft Windows XP объектами стали не все внутренние структуры данных, а только лишь те, которые предполагают или их совместное использование, или возможность именования. Объектами также были сделаны все ресурсы, которые доступны на пользовательском уровне и нуждаются в защите от несанкционированного доступа к ним. Остальные структуры внутренних данных не были реализованы в виде объектов только потому, что по соображениям эффективности большая часть ОС Microsoft Windows XP написана на языке программирования C и дополнительная реализация объектов потребовала бы значительных усилий.

Для управления объектами был создан Object Manager (OM) – компонента ОС Microsoft Windows XP, которая отвечает за создание, уничтожение и защиту объектов ОС Microsoft Windows XP. Она предназначена для локализации системных ресурсов, т.к. в противном случае контроль использования ресурсов сильно усложняется ввиду их распределенности по разным компонентам ОС Microsoft Windows XP. Исходя из этого, разработчики Object Manager определили следующие функции, возлагаемые на данный модуль:

• проведение общего, формализованного механизма использования системных ресурсов;

• изоляция защиты системных объектов в одном модуле, с целью облегчения контроля механизмов защиты;

• предоставление удобной схемы поддержки и использования долговременных объектов, таких как внешние устройства, файлы, директории, файловые системы и т.п.;

• контроль количества системных ресурсов, используемых тем или иным процессом ОС Microsoft Windows XP;

• предоставление формализованных правил доступа к системным объектам;

• поддержка требований различных операционных систем к свойствам системных ресурсов, таких как возможность наследования свойств системных объектов от порождающего процесса, возможность создания строчно-прописных имен файлов и т.п.

ОС Microsoft Windows XP поддерживает два вида объектов: объекты XP executive и объекты ядра (будем называть их ИО и КО). ИО – это объекты, представляющие различные системные ресурсы, поддерживаемые модулями XP executive. Они доступны пользователям системы через объектный сервис XP executive и могут создаваться как защищенными подсистемами ОС Microsoft Windows XP, так и выполняемыми модулями. КО – это более примитивные объекты, представляющие системные ресурсы непосредственно ядра ОС Microsoft Windows XP. Эти объекты недоступны для пользователей системы и используются только внутри выполняемых модулей XP. КО используются для фундаментальной совместимости ОС и аппаратной платформы, на которой она функционирует. Объекты XP executive обычно инкапсулируют один или несколько объектов ядра. В дальнейшем будут рассматриваться только ИО в силу того, что системный интерфейс ОС Microsoft Windows XP зависит только от них.

Каждая защищенная подсистема ОС Microsoft Windows XP предоставляет свой API, отличный от других подсистем. Но все они при создании объектов API используют ИО как низкоуровневые объекты для представления своих ресурсов. Набор объектов, поддерживаемых той или иной подсистемой ОС Microsoft Windows XP, может быть как больше, так и меньше, чем набор объектов выполняемых модулей. Например, Win32, использует ИО для создания и поддержания объектов Win32, практически не изменяя исходные объекты (мутексы, семафоры и т.п.). Таким образом, все объекты, предоставляемые подсистемами на пользовательском уровне, либо непосредственно порождаются, либо наследуются от ИО (например, объекты Win32 API Pipe и Mailslots наследуются от File-объектов выполняемых модулей). В таблице 1 перечислены основные объекты XP executive и модули, которые их регистрируют:

Таблица 1. Объекты исполнительной системы

Тип объекта	Кто определяет	Описание
Процесс (Process)	Модуль управления процессами (Process manager)	Набор системных ресурсов, необходимых для выполнения программы
Поток (Thread)	Модуль управления процессами (Process manager)	Часть программы, способная выполняться независимо от других частей
Секция (Section)	Модуль управления памятью (Memory manager)	Выделенный объем памяти, доступный к совместному использованию несколькими процессами
Файл (File)	Модуль управления вводом/выводом (I/O manager)	Представление некоторого файла или устройства ввода/вывода
Порт (Port)	Модуль диспетчеризации системных запросов (LPC facility)	Объект для обмена сообщениями между параллельными процессами
Маркер доступа (Access token)	Подсистема безопасности (Security subsystem)	Идентификатор, содержащий классификационную информацию о пользователе
Событие (Event)	Модуль поддержки взаимодействия процессов (Executive support services)	Уведомление о происшедшем системном событии
Пара событий (Event pair)	Модуль поддержки взаимодействия процессов (Executive support services)	Объект, определяющий порядок выполнения двух взаимосвязанных потоков (используется только подсистемой Win32)
Семафор (Semaphore)	Модуль поддержки взаимодействия процессов (Executive support services)	Счетчик, регулирующий количество потоков, имеющих одновременный доступ к системному объекту
Мутант (Mutant)	Модуль поддержки взаимодействия процессов (Executive support services)	Механизм, обеспечивающий потокам взаимоисключающий доступ к системному объекту (используется только подсистемами Win32 и OS/2)
Таймер (Timer)	Модуль поддержки взаимодействия процессов (Executive support services)	Счетчик, отслеживающий определенный временной интервал
Директория объектов (Object directory)	Модуль управления объектами (Object manager)	Депозитарий в оперативной памяти, служащий для хранения имен объектов
Символьная связка (Symbolic link)	Модуль управления объектами (Object manager)	Механизм, поддерживающий косвенные ссылки на имена объектов
Профиль (Profile)	Ядро ОС Microsoft Windows XP (Kernel)	Механизм, предназначенный для измерения времени выполнения блока кода
Ключ (Key)	Модуль управления конфигурацией (Configuration manager)	Дескриптор, индексирующий ссылки в базу данных конфигурации ОС Microsoft Windows XP

Следует помнить, что ИО отличаются от объектов, поддерживаемых защищенными подсистемами ОС Microsoft Windows XP в рамках предоставления соответствующего API. Поэтому всюду ниже, где речь идет об объектах, имеются в виду именно ИО.

ИО создаются обычно или различными частями самой ОС, или защищенными подсистемами в ответ на соответствующие запросы прикладных программ. К примеру, для того, чтобы Win32 приложение получило доступ к файлу, вызывается функция CreateFile из Win32 API, а Win32, в свою очередь, вызывает функцию NtCreateFile объектного сервиса ОС Microsoft Windows XP, которая и создает объект выполняемого модуля типа «Файл». Когда приложение позже читает или пишет в этот файл, то Win32 использует этот объект для получения доступа к файлу.

Однако операции с файлами не являются типичным случаем использования объектов выполняемых модулей ОС Microsoft Windows XP в силу того, что файл – это долговременный системный ресурс, а не динамический, хранимый в оперативной памяти. Но, несмотря на это, файловые операции были выбраны для примера потому, что система организации и поддержки объектов выполняемых модулей ОС Microsoft Windows XP очень сильно напоминает организацию файловой системы, и принципы работы с ней можно распространить на систему поддержки ИО:

1. В большинстве языков программирования, до того как писать в файл или читать из него, файл необходимо открыть. Открытие файла подразумевает либо открытие уже существующего файла, используя полное его имя как спецификацию, либо создание нового файла и добавление его уникального полного имени к именам уже существующих файлов.

2. При открытии файла необходимо указать виды операций, которые будут затем производиться с этим файлом (читать, писать, добавлять и т.д.)

3. Файловая система, открыв файл, возвращает дескриптор, который используется приложением для идентификации этого файла. Когда работа с файлом завершена, дескриптор перестает быть его идентификатором и последующее использование этого дескриптора некорректно.

4. Несколько параллельно работающих программ могут совместно использовать файл, если каждая имеет в своем распоряжении его дескриптор. Некоторые файловые системы позволяют дополнительно создавать временные файлы, которые уничтожаются сразу же после закрытия.

С небольшими допущениями можно сказать, что система объектов ОС Microsoft Windows XP имитирует организацию файловой системы. Основное различие заключается в том, что объекты ОС Microsoft Windows XP хранятся в оперативной памяти, а не на долговременном носителе.

Аналогично многим другим операционным системам, ОС Microsoft Windows XP использует процессы для разделения работы. Каждый про-

цесс является набором системных ресурсов, позволяющих выполнять программу. Поток является непосредственно исполняемой частью процесса, имеющей необходимое адресное пространство для кода и для данных.

Когда поток выполняется, он может запросить у операционной системы дополнительные ресурсы, инициируя создание новых объектов или получая дескрипторы уже существующих. При этом дескрипторы объектов предстают, своего рода, идентификаторами системных ресурсов, используемых процессом. Имея дескриптор объекта, процесс может производить над ним любые допустимые действия, вызывая соответствующие системные функции.

Win32 – это тоже процесс (csrss.exe) ОС Microsoft Windows XP, и когда какое-нибудь приложение вызывает функцию из API, прямо или косвенно создающая объект ОС Microsoft Windows XP, Win32 вызывает соответствующую системную функцию, запрашивая Object Manager (OM, Менеджер объектов) создать этот объект и возвратить его дескриптор.

OM при этом должен выполнить следующие функции:

• зарезервировать оперативную память для хранения объекта;

• создать и закрепить за объектом дескриптор безопасности, определяющий, кто из пользователей может иметь доступ к этому объекту и какие действия с этим объектом допустимы;

• создать и поддерживать директорию объектов, в которой хранятся все имена доступных системных ресурсов;

• создать уникальный дескриптор этого объекта и вернуть его процессу, запросившему ресурс.

Все процессы пользовательского уровня, включая подсистемы, должны получить дескрипторы объектов для того, чтобы потоки этих процессов могли их использовать. Аналогично Win32 приложения используют дескрипторы различных типов для представления окон, курсоров и пиктограмм. В обоих случаях дескрипторы выступают как неявные указатели на системные ресурсы и, таким образом исключается прямой доступ и модификация системных данных.

В ОС Microsoft Windows XP дескрипторы имеют еще более широкое назначение. Первое отличие от дескрипторов объектов Win32 – это то, что в ОС Microsoft Windows XP нет различия между дескриптором файла, например, и дескриптором события (за исключением того, что они идентифицируют объекты разных типов). Таким образом, нет необходимости создавать и поддерживать десять различных способов для манипуляций с десятью различными типами объектов. И второе отличие объектов ОС Microsoft Windows XP – это то, что создает их лишь один системный модуль, а это позволяет OM полностью контролировать использование объектов приложениями пользователей. Такой механизм управления объектами позволяет OM эффективно решать важнейшие задачи защищенной операционной системы, такие, как:

1. Защита объектов. При использовании объекта приложением пользователя, OM может провести необходимые проверки прав на запрошенный тип доступа

2. Мониторинг объектов. ОМ в любой момент может узнать, сколько пользователей имеют доступ к данному объекту и своевременно уничтожать объекты, в которых никто не нуждается.

3. Мониторинг системных ресурсов. Каждый раз, при обработке запроса на создание объекта неким процессом, ОМ проверяет количество используемых этим процессом объектов и, в случае превышения установленных квот на системные ресурсы, не допускает открытия новых дескрипторов.

Механизм решения первой задачи многое унаследовал от модели файловой системы: при открытии файла мы должны определить тип доступа к нему и, например, если файл был открыт для чтения, то попытка записи в него приведет к ошибке. Аналогично, при создании или при открытии дескриптора объекта ОС Microsoft Windows XP, процесс, создающий этот объект, должен определить запрашиваемые типы доступа к этому объекту (то есть набор действий, которые процесс будет производить над этим объектом). Это могут быть как стандартные для всех объектов ОС Microsoft Windows XP типы доступа (чтение, запись, выполнение и т.д.), так и специфичные для объектов определенного типа (например, приостановка или завершение для потока). ОМ, в свою очередь, передает запрашиваемые типы доступа Монитору Ссылок (МС), который, проверив дескриптор безопасности этого объекта, возвращает ОМ гарантированные типы доступа, т.е. набор всех возможных видов действий, которые данный субъект может выполнять над этим объектом. Гарантированные типы доступа ОМ сохраняет в дескрипторе безопасности для ускорения выполнения дальнейших проверок. При использовании процессом дескриптора производится быстрая побитная проверка типа доступа и, если такой тип доступа процессу не разрешен, то вызывается соответствующее системное прерывание.

Рис. 2. Структура объектов ОС Microsoft Windows XP

Каждый объект ОС Microsoft Windows XP имеет определенный тип. Тип объекта определяет данные, содержащиеся в объекте, а так же набор действий, которые с этим объектом можно производить. Для того чтобы объектами различных типов можно было управлять однообразно,

ОМ поддерживает несколько полей данных, содержащих стандартную информацию об объекте. ОМ не известно назначение и расположение полей данных, зависящих от конкретного типа объекта. Таким образом, каждый объект состоит из двух частей: из заголовка и из тела. ОМ контролирует только заготовочную часть, а тело объекта находится под контролем того модуля ОС Microsoft Windows XP, который регистрирует этот тип объектов.

ОМ использует заголовочную часть для управления объектами независимо от их типа. На ***рис. 2*** приводится структура объектов ОС Microsoft Windows XP.

Таблица 2. Стандартные атрибуты заголовка объекта

Поле заголовка	Описание
Имя объекта	Предназначено для поддержки совместного использования объекта различными процессами
Директория объектов	Полный путь к данному объекту в иерархической структуре, содержащей имена всех существующих объектов
Дескриптор безопасности	Определяет права доступа к объекту, то есть, кто может использовать данный объект и какой тип доступа к этому объекту может быть предоставлен
Квоты	Набор установленных ограничений на используемые процессами объекты
Счетчик дескрипторов	Количество одновременно открытых дескрипторов данного объекта
База данных	Список процессов, получивших дескриптор данного объекта
Временный/постоянный	Определяет время жизни объекта (временные объекты уничтожаются, если счетчик открытых дескрипторов равен нулю)
Ядро/пользовательский	Определяет доступность объекта для процессов пользовательского уровня
Указатель типа	Указатель на тип объекта (набор атрибутов, одинаковых для всех объектов данного типа)

Помимо заголовка, ОМ поддерживает небольшой набор внутренних функций (методов) для управления объектами. Ниже приведен список этих методов.

Таблица 3. Набор внутренних функций для управления объектами

Метод	Назначение
Закрыть	Закрывает дескриптор объекта
Продублировать	Разделяет объект путем дублирования дескриптора и передачи его другому процессу

Запросить атрибуты	Запрашивает стандартные атрибуты объекта
Запросить права	Возвращает дескриптор безопасности объекта
Установить права	Изменяет права доступа к этому объекту
Ожидать объект	Синхронизирует выполнение потока с объектом
Ожидать объекты	Синхронизирует выполнение потока с несколькими объектами

Каждый модуль ОС Microsoft Windows XP может определить свой объект или объекты. Для этого ОМ должна быть передана информация о размерах тела объекта и набор функций для работы с ним (Все объекты одного типа должны иметь одинаковую структуру и одинаковый набор функций).

Заголовок объекта содержит данные, общие для всех объектов, но значения которых для различных копий объектов могут быть разными. Однако объекты могут содержать данные, одинаковые для всех копий объектов определенного типа. Например, набор возможных видов доступа к объекту конкретного типа или информация о возможности синхронизации. Таким образом, в целях экономии оперативной памяти и уменьшения затрат на поддержку объекта, эти данные хранятся в отдельной структуре, называемой «тип объекта», а в заголовке объекта содержится лишь указатель на эту структуру. «Тип объекта» служит также для связи между собой различных копий объектов одинакового типа (таким образом, ОМ может легко найти другие объекты этого же типа или перечислить их все, если потребуется).

«Тип объекта» не доступен для процессов пользовательского уровня, так как ОМ не предоставляет функций для манипулирования с этой структурой. Однако, значения некоторых атрибутов «типа объекта» доступны пользователям посредством соответствующих сервисных функций. В таблице 4 приведены атрибуты структуры «тип объекта».

Таблица 4. Атрибуты структуры «тип объекта»

Атрибут	Назначение
Имя типа	Название типа (порт, файл и т.д.)
Виды доступа	Список видов доступа к объектам этого типа (запись, чтение, удаление и т.п.)
Синхронизация	Возможность синхронизации исполняемого потока с объектом данного типа
Методы	Одна или несколько процедур, которые ОМ вызывает в определенные моменты существования объекта данного типа
Выгружаемость	Может ли объект данного типа выгружаться на диск из оперативной памяти

Синхронизация – пример атрибута, видимого на пользовательском уровне. Поток может синхронизироваться с объектами типа процесс, по-

ток, событие, пара событий, семафор, мутант и таймер. Секция, порт, маркер доступа, директория объектов, символьная связка, профиль и ключ не поддерживают механизм синхронизации.

Методы – это набор внутренних процедур, напоминающих конструкторы и деструкторы C++, то есть, это процедуры, автоматически вызываемые для объектов данного типа в определенное время. Методы вызываются, например, когда происходит открытие или закрытие дескриптора объекта или когда происходит изменение прав доступа к объекту. Заметим, что для различных типов объекта методы объекта различны. Ниже будет более подробно освещен вопрос назначения и механизмов поддержки методов объекта.

Подводя итог вышесказанному, можно сделать вывод, что каждый объект состоит из двух частей: заголовка, контролируемого ОМ и тела, контролируемого тем модулем, который определяет и поддерживает этот объект. Заголовок объекта содержит указатель на структуру «тип объекта», определяющую данные, общие для всех объектов одинакового типа. Любой модуль ОС Microsoft Windows XP может определить новый тип объекта, задав размер тела объекта и заполнив поля структуры «тип объекта».

Как отмечалось выше, ОМ поддерживает набор функций, общих для объектов всех типов. Функции, различные для разных типов, определяются и поддерживаются теми модулями, которые создавали эти типы объектов. При запросе модуля, предоставляющего объект, определенного типа доступа к нему от пользовательского приложения, вызывается ОМ. При этом используется внутренний интерфейс, недоступный приложениям пользовательского уровня. ОМ, проверив соответствующие права на такой тип доступа, вызывает соответствующую процедуру модуля, зарегистрированную им при определении типа, при этом используя внутренний интерфейс. Таким образом, все типы доступа к системным ресурсам находятся под контролем ОМ, являющимся своего рода шлюзом между приложениями и системными ресурсами.

Важным моментом при создании и управлении большим числом объектов является динамический контроль за использованием этих объектов, то есть возможность быстрого поиска нужного объекта и получение всех его необходимых спецификаций. Для адекватного выполнения этих функций ОМ должен быть способен решать следующие задачи:

- однозначно идентифицировать объект;
- найти определенный объект;

Первая задача решается с помощью уникальных имен объектов и, в отличие от большинства существующих ОС, ОС Microsoft Windows XP позволяет именовать все без исключения объекты, а не только файлы, блоки разделяемой памяти и каналы передачи данных. Вторая задача также может быть решена посредством имен. То есть, если ОМ хранит объекты в виде иерархического дерева имен, то для поиска нужного объекта достаточно лишь просмотреть это дерево.

Именование объектов позволяет дополнительно решить задачу разделения ресурсов. Ввиду того, что имена ресурсов доступны всем

процессам, создатель объекта, давая ему имя, позволяет другим процессам использовать его при открытии дескриптора этого объекта. В том случае, если создатель объекта не заинтересован в его совместном использовании другими процессами, то он может просто не именовать данный объект.

В целях повышения быстродействия ОС, ОМ не просматривает имя объекта при каждом обращении к нему, а делает это лишь в двух случаях:

1. При создании именованного объекта для разрешения конфликта имен.

2. При открытии дескриптора именованного объекта. (ОМ просматривает список имен объектов, находит нужный и возвращает дескриптор запросившему процессу).

В остальных случаях ОМ использует только лишь дескрипторы объектов для обслуживания системных запросов. Заметим, что ОМ позволяет использовать в именах объектов как строчные, так и прописные буквы, различая их между собой.

Список имен объектов доступен всем процессам, выполняющимся на одном локальном компьютере, но при удаленном доступе к объекту возникают определенные трудности. Для разрешения этой проблемы ОМ использует специальный механизм поиска имени удаленного объекта. Например, для доступа к удаленному файлу, ОМ вызывает специально зарегистрированную процедуру подсистемы ввода/вывода, которая через подсистему ввода/вывода того компьютера, на котором находится файл, вызывает удаленный ОМ, возвращая необходимую информацию о гарантированных видах доступа обратно.

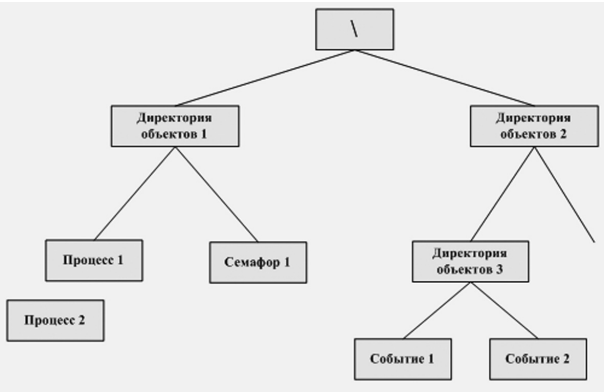


Рис. 3. Структура имен объектов

Для решения задачи формирования имен объектов разработчики ОМ взяли за основу файловую модель MS DOS. Таким образом, ОМ может понимать формат имен файлов и быстро находить нужный, при этом имена остальных объектов легко вписываются в эту модель. Имя

любого объекта ОС Microsoft Windows XP строится по аналогии с полным именем файла: от самого корневого каталога объектов разбор имени ведется вниз по дереву имен, используя промежуточные объекты типа «Директория объектов» в качестве узлов этого дерева. Для разделения этих имен используется тот же самый разделитель, что и в именах файлов – обратный слеш.

Объект типа «Директория объектов» – это специализированное средство поддержки иерархии объектов ОС Microsoft Windows XP. Этот тип создается самим ОМ и основным его свойством является способность связывать объекты, содержащиеся в этой директории. Это могут быть объекты опять того же типа, что позволяет реализовать описанную иерархическую модель. Любой процесс может создать свою собственную директорию объектов для хранения там своих имен, при этом любой другой процесс так же может хранить там свои объекты, если он имеет к этой директории соответствующий вид доступа. Например, подсистема ввода/вывода создает свою собственную директорию с именем \Devices, в которой хранятся все имена устройств ввода/вывода.

В таблице 5 приведены важнейшие характеристики директории объектов. При этом «тип объекта» – это ссылка на соответствующую структуру, подробно описанную выше. Тело объекта – это поля данных, индивидуальных для определенного типа объекта, но переменных для различных копий. Они содержат информацию для поддержки функциональности объектов данного типа. Сервис – это набор процедур, позволяющих производить определенные действия над объектом. Атрибуты заголовка не указаны ввиду того, что их подробное описание было выше.

Таблица 5. Основные характеристики директории объектов

Тип объекта	Директория объектов
Тело объекта	Список ссылок на имена объектов, содержащихся в этой директории
Сервис	Создать Открыть Запросить

Функции «Создать» и «Открыть» используются для создания новой директории объектов и для открытия дескриптора на уже существующую директорию. Если поток открывает дескриптор директории объектов с правом на запись, то он может помещать имена своих объектов в эту директорию. Функция «Запросить» позволяет просматривать список имен объектов, помещенных в эту директорию. Директория объектов позволяет даже получать другим модулям ОС Microsoft Windows XP непосредственно указатели на хранимые объекты, которые ОМ затем может преобразовать в дескрипторы для передачи их процессам пользовательского уровня.

Отметим, что три эти процедуры («Создать», «Открыть», «Запросить») повторяются для почти всех объектов ОС Microsoft Windows XP и,

поэтому их можно было бы отнести к стандартным методам, поддерживаемым ОМ, но из-за того, что для различных типов объектов эти функции требуют разного числа передаваемых параметров, их реализация была оставлена в ведении модулей, непосредственно определяющих тип. Таким образом, ОМ для каждого объекта определяет его тип, находит ссылку на требуемую процедуру и вызывает ее.

Имена объектов ОС Microsoft Windows XP посредством директорий образуют некую зонтичную структуру, в которую при необходимости легко встраиваются новые объекты, образующие вторичные иерархические структуры имен. Такие контейнерные объекты получили название домены объектов. Подсистема ввода/вывода, например, поддерживает такой домен, содержащий устройства ввода/вывода, директории и файлы дисков. ОМ, таким образом, позволяет встраивать в свою директорию объектов объекты, предоставляемые устройствами ввода/вывода. Предположим, что структура директорий некоторого диска выглядит следующим образом (***рис. 4***):

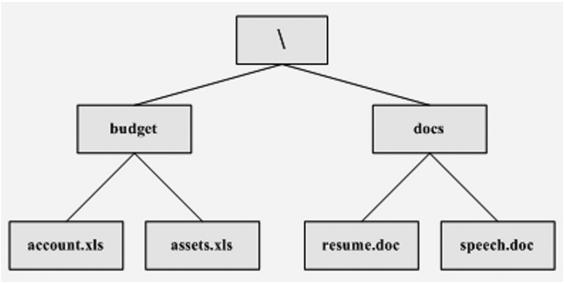


Рис. 4. Структура директорий диска

Со стороны ОМ эта структура выглядит приблизительно так (***рис. 5***):

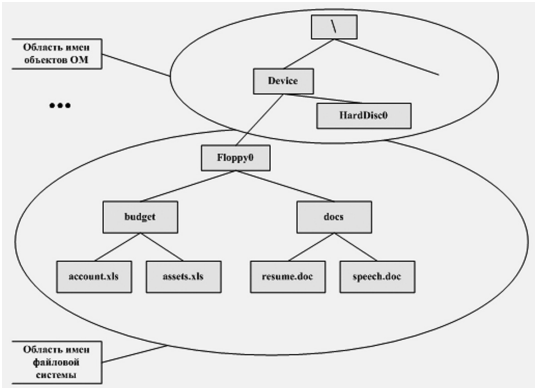


Рис. 5. Структура директорий диска в представлении ОМ

В этом дереве каждое имя представляет некоторый объект ОС Microsoft Windows XP. А область имен файловой системы встроена в директорию имен объектов под именем \Device\Floppy0.

Таким образом, когда пользователь посредством приложения Microsoft Excel открывает файл a:\budget\accounts.xls, ОМ открывает дескриптор на объект с именем \Device\Floppy0\budget\accounts.xls. Он просматривает дерево имен объектов пока не находит имя Floppy0. Это имя является именем специального объекта, который создается подсистемой ввода/вывода для организации вторичного домена объектов. С этим объектом зарегистрирован специальный метод «поиск», вызываемый ОМ с именем требуемого файла в качестве передаваемого параметра, а данная процедура затем находит и открывает соответствующий файл. Такой механизм организации структуры объектов позволяет ОМ хранить только те объекты, для которых открыты дескрипторы, не загромождая оперативную память компьютера неиспользуемыми долговременными объектами, такими как директории и файлы дисков.

Символьная связка – это второй тип объектов, которые создаются самим ОМ. Этот тип служит для организации перекрестных ссылок внутри области имен объектов ОМ. Когда процесс использует имя объекта, содержащее символьную связку, то ОМ переводит его в имя реально существующего объекта ОС Microsoft Windows XP. Для этого в первоначальное имя объекта, вместо имени символьной связки, подставляется строка символов, специально зарегистрированная при создании объекта-связки. Имя символьной связки может быть использовано в любом месте полного имени объекта ОС Microsoft Windows XP. В таблице 6 приведены атрибуты и сервис для объектов типа «символьная связка».

Таблица 6. Атрибуты и сервис для объектов типа «символьная связка»

Тип объекта	Символьная связка
Атрибуты тела объекта	Подставляемая символьная строка Время создания
Сервис	Создать Открыть Запросить

Символьная связка используется, например, для перевода имен DOS-устройств в имена объектов-устройств ОС Microsoft Windows XP. В MS DOS пользователи используют имена дисковых накопителей: A:, B: и т.п. Более того, пользователи могут добавлять новые имена устройств путем создания логических разделов, виртуальных дисков или для обозначения сетевых дисковых устройств. Имена этих устройств должны быть доступны всем системным процессам с момента их создания. Для разрешения этой проблемы подсистема Win32 создает специальную директорию в дереве объектов ОС Microsoft Windows XP под названием \??, в которую и помещает имена всех вышеупомянутых DOS-устройств, таким образом, они сразу же становятся доступными для всех процессов ОС Microsoft Windows XP.

При создании нового DOS-устройства Win32 помещает его имя в эту директорию. При этом имена реальных устройств ввода/вывода находятся в другой директории (\Devices), которую создает подсистема ввода/вывода.

Имена A:, B: и т.д. – это на самом деле имена объектов типа «символьная связка», в которых в качестве атрибута содержится имя реального устройства ввода/вывода. Таким образом, если пользователь посредством приложения Microsoft Excel открывает файл A:\budget\accounts.xls, то Win32 передает OM запрос на открытие файла \??\ A:\budget\accounts.xls. При поиске этого объекта OM встречает имя символьной связки «A:» и заменяет его на строку «\Devices\Floppy0», при этом, к строке, вставляемой вместо имени символьной связки, OM присоединяет оставшуюся часть начального имени объекта, а затем вновь начинает поиск от корневого каталога объектов.

Символьные связки позволяют защищенным подсистемам или другим процессам создавать имена объектов, которые могут изменяться динамически. Более того, подсистемы могут таким образом поддерживать аппаратно-независимое исполнение приложений, помещая имена устройств в специальную директорию объектов, вместо того, чтобы сохранять их в своем адресном пространстве.

Несмотря на то, что имена очень важны для сохранения и разделения объектов, они используются OM не часто: в случаях, когда объект создается или когда процесс открывает дескриптор разделяемого объекта. В остальных случаях, для работы с объектом используются их дескрипторы. Такой подход гораздо эффективнее и быстрее в силу того, что пропускается поиск имени в директории объектов и к нужному объекту OM обращается напрямую. Дескриптор объекта – это, на самом деле, индекс в специальной таблице объектов, которая прикрепляется к каждому процессу в системе. Эта таблица содержит указатели на все объекты, для которых у этого процесса имеются открытые дескрипторы. Процесс может получить дескриптор объекта четырьмя способами (***рис. 6***):

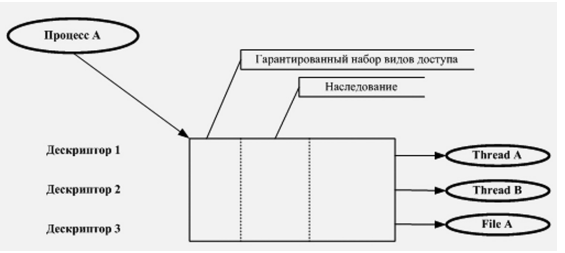


Рис. 6. Алгоритм получения дескриптора объекта

1. Создав этот объект;
2. Открыв дескриптор существующего объекта по его имени;
3. Унаследовав дескриптор от родительского процесса;

4. Получив продублированный дескриптор от другого процесса.

Каждая ячейка этой таблицы содержит поле, в котором находится набор гарантированных видов доступа и флаг наследования – то есть флаг, определяющий будут ли дочерние процессы иметь открытый дескриптор этого объекта.

Два процесса разделяют объект, если они оба имеют открытые дескрипторы данного объекта. При этом значения самих дескрипторов могут быть различными.

После завершения процесса, открытые дескрипторы объектов, используемых данными процессами, становятся претендентами на уничтожение, но они уничтожаются, только если ни у одного из процессов, использующих данные объекты, нет соответствующих открытых дескрипторов. Такая выборочность достигается с помощью вызова специального метода при уничтожении объекта типа «процесс».

Так как все процессы пользовательского уровня, прежде чем использовать объект, должны открыть его дескриптор, то ОМ в любой момент может легко установить: сколько и какие процессы используют тот или иной объект. Наличие этой информации позволяет ОМ своевременно уничтожать неиспользуемые временные объекты.

Время жизни любого объекта, с точки зрения ОМ, состоит из двух фаз. Первая – это время жизни имени объекта. Каждый раз, когда процесс открывает дескриптор объекта, ОМ увеличивает соответствующий счетчик в заголовке этого объекта, а при закрытии дескриптора уменьшает его. Когда счетчик открытых дескрипторов объекта уменьшается до нуля, ОМ уничтожает его имя из директории объектов. Это касается только временных объектов; для объектов долговременного использования, таких, как физические устройства ввода/вывода и т.п., имена продолжают храниться в директории объектов, даже если ни один процесс не имеет соответствующих открытых дескрипторов. Вторая фаза – время жизни самого объекта. Дело в том, что процессы ядра ОС Microsoft Windows XP используют непосредственно указатели на объекты для доступа к ним, и, в силу этого, ОМ должен сохранять количество используемых указателей на объект в специальном счетчике, называемым счетчиком ссылок. Для этого, при получении указателя на объект привилегированным процессом, ОМ увеличивает этот счетчик, а при отсутствии необходимости в дальнейшем использовании указателя соответствующий процесс уведомляет об этом ОМ.

В ОС Microsoft Windows XP учет использования системных ресурсов, как и время жизни объектов, определяется, в основном, счетчиком открытых дескрипторов на объект. Если какой-либо процесс имеет открытый дескриптор, то оперативная память и другие системные ресурсы, выделенные под этот объект, привязываются к контексту этого процесса и учитываются при проверке превышения квот на использование системных ресурсов данным процессом.

Многие ОС вводят определенные ограничения на использование системных ресурсов процессами пользовательского уровня. При этом контроль за соблюдением этих ограничений ложится на те системные

модули, которые отвечают за распределение соответствующих ресурсов. Подсистема ввода/вывода, например, контролирует количество открытых файлов, модуль управления памятью контролирует количество оперативной памяти, выделяемой пользовательскому процессу. В отличие от других ОС, в Microsoft Windows XP весь контроль за использованием объектов сосредоточен в ОМ. Для этого в каждом заголовке объекта ОМ хранит информацию об установленных квотах на использование пользователями этого объекта. То есть, каждый раз, когда поток открывает какой-нибудь объект, ОМ проверяет таблицу дескрипторов этого потока и, если установленные квоты на объекты такого типа превышены, не позволяет открывать запрашиваемый объект.

Следует отметить, что в добавление к мониторингу объектов, производимому ОМ, модуль управления процессами устанавливает ограничения на количество процессорного времени, используемого пользовательскими потоками.

ОМ заполняет и поддерживает заголовки объектов таким образом, чтобы управление ими было организовано одинаковым образом. Однако объекты различных типов имеют разные структуры и функции. Поэтому ОМ был бы гораздо сложнее и больше, если при управлении объектами учитывались все эти различия, а при появлении новых типов сам ОМ подвергался бы модификации. Для разрешения этой проблемы был введен специальный механизм, называемый «методы объектов», позволяющий выполнять функции, индивидуальные для объектов определенного типа, в тех модулях ОС Microsoft Windows XP, которые регистрируют этот тип.

Когда какой-нибудь модуль ОС Microsoft Windows XP создает новый тип объектов, он может зарегистрировать у ОМ один или несколько методов, связанных с этим типом. Затем, в определенные при регистрации моменты, ОМ вызывает эти методы для объектов соответствующего типа.

Хороший пример использования методов объекта – это метод «закрыть», зарегистрированный подсистемой ввода/вывода для объектов типа «файл». ОМ вызывает этот метод каждый раз, когда какой-нибудь процесс закрывает дескриптор файла. Подсистемой ввода/вывода этот метод используется для того, чтобы проверить наличие не сброшенных процессом блокировок файла и удалить их, при необходимости. Проверка блокировок файла – это задача, которую ОМ, сам по себе, не может выполнить, по причине отсутствия какой-либо информации об этих блокировках, а также в силу соображений концептуальной целостности ОС Microsoft Windows XP.

ОМ вызывает метод «уничтожить», если таковой зарегистрирован, перед тем как удалить из оперативной памяти временный объект. Модуль управления памятью, например, регистрирует такой метод для объектов типа «секция». Он необходим для освобождения физических страниц памяти, используемых этим объектом, а также для удаления внутренних структур модуля управления памятью, зависящих от размера выделенной памяти. ОМ ничего не известно об этих структурах и, поэтому для «секции» необходимо зарегистрировать данный метод.

Метод «поиск» (а также аналогичный ему метод «запросить») позволяет ОМ передать управление поиском объекта системному модулю, который поддерживает собственные директории объектов. Этот модуль находит объект, расположенный вне области видимости имен ОМ и передает назад его имя. Простейший пример использования данного метода можно найти при рассмотрении взаимодействия ОМ и подсистемы ввода/вывода:

Объект с именем Floppy0 – это объект специального типа «устройство», который регистрируется и поддерживается подсистемой ввода/вывода. В директории имен ОМ этот объект представляет некую точку входа во вторичную директорию объектов, обслуживаемую подсистемой ввода/вывода, и о которой ОМ ничего не известно. Когда подсистема ввода/вывода создает тип «устройство», она регистрирует ассоциированный с этим типом метод «поиск». Затем, когда ОМ просматривает какое-нибудь полное имя, то, дойдя до объекта, с которым зарегистрирован «поиск», вызывает этот метод, передавая ему в качестве параметра остаток полного имени.

Например, когда процесс открывает дескриптор объекта с именем \Devices\Floppy0\docs\resume.doc, ОМ просматривает путь к этому объекту до тех пор, пока не достигнет имени объекта Floppy0. Тогда ОМ вызывает метод «поиск», ассоциированный с объектами этого типа и передает ему в качестве параметра строку «\docs\resume.doc». Вызываемый метод является просто функцией подсистемы ввода/вывода, которая передает строку с именем файла соответствующей файловой системе, находит файл на диске и открывает его, передавая затем управление обратно ОМ.

«Символьная связка» – это еще один тип объектов, которые имеют ассоциированный с ними метод «поиск». При вызове этого метода ему передается полный путь к объекту, в котором содержится имя символьной связки. При выполнении этой процедуры вместо имени символьной связки подставляется ассоциированная с ней строка и управление передается обратно, с тем, чтобы ОМ продолжил поиск нужного объекта, но уже по новому пути в директории объектов. Если новый путь опять содержит символьную связку, то описанный процесс повторяется.

Метод «безопасность», используемый подсистемой ввода/вывода, очень похож на «поиск». Он вызывается каждый раз, когда какой-либо поток пытается изменить информацию безопасности файла. Такой подход необходим, так как, в отличие от других объектов, информация безопасности файла хранится в нем самом и ОМ не имеет к ней прямого доступа.

7.6. Реализация требований к безопасности ОС. Защита объектов доступа

Хотя именование, разделение и учет использования системных ресурсов являются достаточно вескими причинами для использования модели объектов, возможно наиболее весомой причиной создания ОМ

стал тот факт, что ОС Microsoft Windows XP – это защищенная операционная система, для которой модель объектов является наиболее простой реализацией политики безопасности.

Microsoft Windows XP – многопользовательская ОС, поэтому файлы, оперативная память и другие системные ресурсы, используемые одним пользователем, должны быть защищены от несанкционированного использования их другими пользователями системы. С другой стороны ОС Microsoft Windows XP должна защищать свои собственные данные от пользовательских программ. Поэтому защита системных ресурсов представляется весьма сложной и многоплановой задачей.

Следует отметить, что при разработке ОС Microsoft Windows XP конечной целью было создание защищенной ОС. Разработчики ОС Microsoft Windows XP при создании этой ОС использовали идею, состоящую в том, что все запросы пользователей к системным ресурсам должны осуществляться через один системный модуль, решающий все вопросы безопасности локального компьютера. Так как все системные ресурсы ОС Microsoft Windows XP представлены как объекты, то ОМ и была отведена роль упомянутого системного модуля. Таким образом, все операции с критичными с точки зрения безопасности системными ресурсами сосредоточены в одном модуле – ОМ, что в значительной мере облегчает контроль за доступом к объектам ОС Microsoft Windows XP.

Маркер доступа (access token) ОС Microsoft Windows XP однозначно идентифицирует процесс и потоки, в то время как список контроля доступа (ACL) объекта определяет, какие из этих процессов могут иметь доступ к объекту. Таким образом, когда процесс открывает дескриптор какого-нибудь объекта, ОМ и подсистема безопасности совместно используют эту информацию для определения правомочности такого запроса.

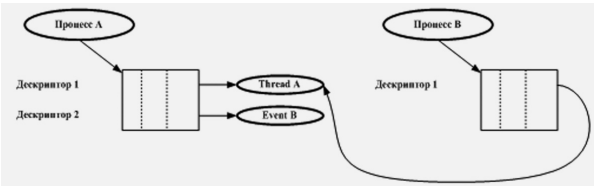


Рис. 7. Алгоритм совместного использования объекта

При проверке прав доступа подсистема безопасности последовательно просматривает записи из ACL объекта. Если встречается запись, разрешающая запрашиваемый тип доступа, то процедура проверки завершается и пользователю возвращается соответствующий дескриптор объекта. Если же первой встречается запись, запрещающая запрашиваемый тип доступа, то пользователю дескриптор не возвращается. Если просмотр всех записей ACL объекта не дал никаких результатов, то считается, что данный вид доступа пользователю запрещен. Заметим, что

записи в ACL объекта ОС Microsoft Windows XP отсортированы обычно таким образом, что записи, запрещающие доступ, располагаются в начале списка.

Так как за время выполнения процесс открывает и использует множество объектов, а ACL объекта может иметь очень много записей, то реализация проверок прав доступа при каждом обращении к объекту может сильно снизить производительность всей системы. Поэтому полная проверка прав доступа в ОС Microsoft Windows XP осуществляется только при открытии дескриптора объекта. Заметим, что, так как сами модули ОС Microsoft Windows XP используют указатели, а не дескрипторы объектов, то для системных модулей не производится вообще никаких проверок прав доступа. В этом смысле можно сказать, что ОС Microsoft Windows XP «доверяет» своим собственным системным модулям.

При использовании объекта каким-либо процессом, ОМ осуществляет лишь быструю проверку гарантированных прав доступа, которые хранятся непосредственно в дескрипторе объекта. При этом пользователь может иметь к объекту только те типы доступа, которые были им перечислены при открытии дескриптора.

7.7. Подсистема регистрации и протоколирования

Подсистема регистрации и протоколирования предназначена для фиксирования фактов возникновения ошибок, их причин, попыток несанкционированного доступа, а также иных событий, происходящих в системе.

В операционной системе Windows XP имеется компонента EventLog, которая отвечает за протоколирование событий, происходящих в системе. Данная служба имеет открытый интерфейс, поэтому любое выполняющееся в системе приложение, системный сервис или драйвер могут сделать запись о том или ином событии.

Схематически схема протоколирования события выглядит, как показано на ***рис. 8***.

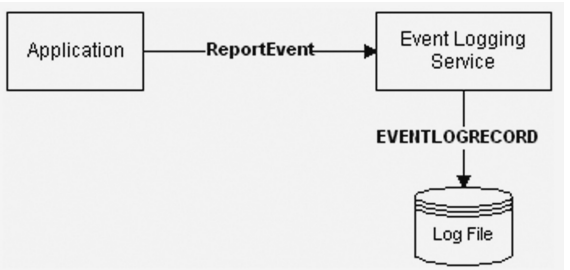


Рис. 8. Схема протоколирования события

Протоколирование событий может осуществляться как локально (в файл журнала, находящийся на данном компьютере), так и на любой другой компьютер локальной сети, к журналу событий которого протоколирующий компьютер может получить доступ. Такая возможность

позволяет «собирать» все события на выделенном рабочем месте администратора сети.

Служба событий имеет 3 стандартных журнала событий, предназначенных для событий от разных компонент операционной системы:

1. Журнал приложений (Application Log) – предназначен для регистрации событий, происходящих в приложениях 3-го кольца. В данном журнале регистрируются события, происходящие в пользовательских приложениях и системных сервисах.

2. Журнал системы (System Log) – предназначен для регистрации системных событий. В данном журнале регистрируются события, происходящие в драйверах, т.н. Native-приложениях и некоторых системных сервисах.

3. Журнал безопасности (Security Log) – предназначен для регистрации событий подсистемы контроля доступа и аудита. В данный журнал заносятся события, регистрирующие факты доступа к объектам операционной системы. Тип регистрируемых событий (успешный доступ или отказ в доступе), а также список объектов, доступ к которым протоколируется, задается администратором системы.

Кроме указанных выше стандартных журналов событий в операционной системе Windows XP имеется возможность создавать дополнительные журналы событий. Это позволяет выделить события от некоторой группы приложений в отдельный журнал. Подобная возможность позволяет также установить специальные права доступа к данному журналу, дополнительно контролируя тем самым доступ к нему.

Имеется пять типов событий, для которых предусмотрено протоколирование. Тип события определяется приложением, которое осуществляет запись в журнал событий, и может быть одним из указанных в таблице 7:

Таблица 7. Типы событий

Тип события	Описание
Информация	Информационное событие означает успешное завершение какой-либо операции. Например, в случае успешной установки сетевой конфигурации в программе «Администратор ключевой системы» в журнале событий появится запись «Сетевая конфигурация успешно установлена».
Предупреждение	Предупреждающее событие сообщает о проблеме, которая, возможно, не требует немедленного решения, но может породить нежелательные последствия в будущем. Например, в случае малого количества свободного места на диске (менее 10% от общего объема) в журнал регистрации событий заносится соответствующая запись.

Тип события	Описание
Успешный аудит (разрешение доступа)	Событие успешного аудита – это событие подсистемы безопасности, которое означает, что доступ к охраняемому объекту операционной системы успешно получен. Примером события успешного аудита может служить сообщение об успешном входе пользователя в систему.
Аудит отказа (запрещение доступа)	Событие, регистрирующее факт отказа в доступе, – это событие подсистемы безопасности, которое означает, что в доступе к охраняемому объекту отказано по причине отсутствия достаточных прав доступа. Примером аудита отказа может служить регистрация запрета доступа к некоторому файлу.
Ошибка	Сообщение об ошибке означает, как правило, серьезную проблему, о которой должен знать администратор системы. Например, если произошел сбой при запуске системного сервиса, то компонента Service Control Manager делает запись о произошедшей ошибке.

В отличие от событий типа «Информация», «Предупреждение» или «Ошибка» события аудита регистрируются только подсистемой ограничения доступа.

Приложения, системные сервисы и драйверы могут не только протоколировать происходящие события, но и добавлять в этот процесс новую функциональность путем регистрации собственных источников событий (Source). Источник события состоит из имени, файла сообщений и других параметров. Такой подход позволяет все возможные сообщения программы вынести в отдельный модуль, а системе регистрации событий передавать только идентификатор события и, если требуется, дополнительные данные. В результате в журнале сообщений хранятся только имя источника события, идентификатор события и его параметры (опционально). В данном случае чтение данных о событии будет иметь следующий вид, представленный на ***рис. 9***.

Наряду с журналами событий локального компьютера данная программа позволяет открывать журналы событий с других компьютеров локальной сети.

Программа просмотра поддерживает следующие операции по работе с журналами событий:

- сортировка событий по времени, источнику, компьютеру и другим параметрам;
- поиска событий;
- создания фильтров событий по источнику событий, временному интервалу, типу события, компьютеру, на котором произошло событие и числовому идентификатору события;
- просмотр расширенной информации о произошедшем событии;
- архивация журналов событий и просмотр сделанных ранее архивов журналов событий;

• очистка журналов событий.

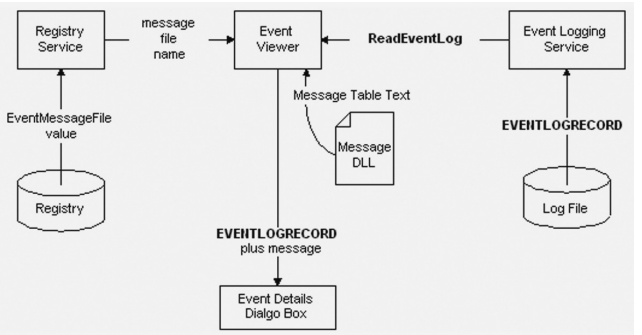


Рис. 9. Чтение данных о событии при наличии дополнительного файла сообщений

Программа просмотра позволяет также экспортировать файлы журналов событий в текстовые файлы и файлы, поля в разделены запятыми (Comma Delimited Text). Эта возможность позволяет сторонним разработчикам создавать собственные программы работы со списками событий, расширяя отсутствующие возможности стандартной программы.

8. СИСТЕМНЫЕ ПРОБЛЕМЫ И СПЕЦИАЛЬНЫЕ РАЗДЕЛЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

8.1. Методология создания защищенных компьютерных систем.

Кратко опишем методологию процесса проектирования защищенной КС. При этом будем опираться на сформулированные выше утверждения, методы и подходы. Изложенный выше понятийный и методический аппарат позволяет заказчику системы и владельцу, который будет потом ее использовать «говорить на одном языке» с разработчиками архитектур КС, программистами прикладных систем и специалистами по безопасности, понимать предложенные ими решения, правильно оценивать реальные параметры защищенной системы, самостоятельно и осознанно применять и совершенствовать организационные и оргнизационно-технические меры безопасности.

Первоначально, исходя из общей архитектуры и назначения КС, определяются существенно важные элементы, связанные с ее архитектурой, с распределенностью КС, с составом аппаратных компонент, с составом и свойствами «программного наполнения» (в первую очередь свойствами операционных сред КС). Например, если КС замкнута внутри корпорации и не допускает работу удаленных пользователей, то звено внешних пользователей не нужно реализовывать, что упрощает и удешевляет систему и упрощает реализацию функций безопасности для КС в целом.

Далее формулируется политика безопасности, реализуемая в КС (состоящая, как было указано, в выборе критерия различения потоков легального и несанкционированного доступа). Затем политика безопасности подвергается коррекции, учитывающей распределенность компьютерной системы. Затем уточненная политика безопасности подвергается содержательному анализу с целью определения ее адекватности целевой функции защищаемой КС.

Надо учитывать, что в современных операционных средах уже существуют, мониторы безопасности объектов и мониторы безопасности субъектов, а также криптографические системы, реализующие описанные выше методы защиты объектов. Эти средства называют **штатными средствами безопасности** операционных сред.

Следующей стадией является соотнесение скорректированной политики безопасности с возможностями, реализуемыми штатными средствами операционных сред КС.

Основным результатом данного этапа проектирования КС является вывод о необходимости приобретения или разработки дополнительных средств безопасности, либо о возможности решения всех проблем безопасности с использованием штатных средств безопасности операционных сред. Важным вопросом является использование криптографичес-

ких средств, в первую очередь необходимо ориентироваться на национальные криптографические средства, одобренные уполномоченными государственными органами (сертифицированные).

Необходимо также сделать выводы о субъектном наполнении КС – проанализировать используемые в КС программы и определить их разумный минимум для решения поставленных перед пользователями задач. Совершенно определенно нельзя допускать одновременной работы рядовых пользователей и программистов-разработчиков в одном локальном сегменте КС. Это связано с тем, что программисты работают с нестационарными субъектами и принципиально нарушают корректность используемого программного обеспечения. Также все программное обеспечение, не связанное с прямой функциональностью КС должно быть вынесено вне ее рамок.

Далее требуется сформулировать технологию управления в КС, уточнить структуры и реализовать субъекты управления, определить вопросы выработки и использования ключей для шифрования и электронной цифровой подписи, а также сформулировать необходимые организационно-технические меры безопасности. Затем, как правило, необходим этап опытной эксплуатации КС. К этому моменту КС содержит операционные среды и прикладное наполнение со свойствами корректности включенных субъектов и «инфраструктуру» (программы и данные) для управления защитой. Цель этапа опытной эксплуатации – убедиться в выполнении целевой функции КС и встроенных в нее защитных механизмов (т. е. решает ли КС и ее защита те задачи, для которых была спроектирована).

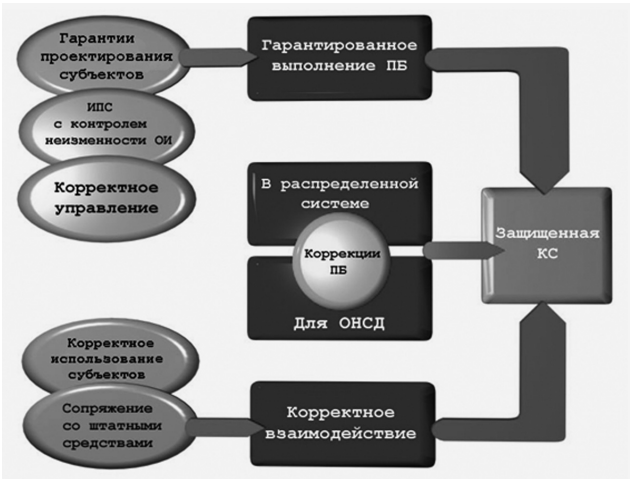


Рис. 1. Взаимосвязь методов проектирования защищенной КС

Наконец, прикладное наполнение КС должно быть замкнуто в изолированную программную среду. При этом либо полноценно реализует-

ся монитор безопасности субъектов, разрешающий порождение только разрешенного списка задач, либо создаются различные выделенные подсистемы (например, межсетевым экраном разделяются внутренний и внешний сегмент КС). Гарантирование политик безопасности может производиться и другими способами, зависящими от архитектуры, способа применения и целевой функции конкретной КС. Например, хорошей практикой является использование систем терминального доступа, в которых программное наполнение загружается с сервера и поэтому по определению замкнуто и проверено.

Результатом работ является КС (в виде документированного проекта и стенда или макета), предназначенная для выполнения предписанных заказчиком системных следований целевых функций, имеющая запас по производительности и надежности, в которой гарантированно выполнена заданная политика безопасности. Процесс проектирования схематически изображен на ***рис. 1***.

8.2. Типовые архитектуры безопасности

Для эффективного функционирования защищенной КС целесообразно наряду с политикой безопасности и целевой функцией также определить ее архитектуру. Здесь и далее под **архитектурой** будем понимать структуру компьютерной системы с выделением ее компонентов, элементов и связей между ними. Связи между элементами являются системообразующими, а сама архитектура часто представляется в виде изображения или схемы.

Рассмотрим в качестве показательного примера пятизвенную архитектуру защищенной КС. Пять звеньев включают в себя: сервер баз данных, сервер доступа, фронт-сервер, рабочие места локальных пользователей и рабочие места удаленных пользователей. Такая архитектура оправдана практически для любой защищенной системы. Например, система обработки безналичных платежей с такой архитектурой будет функционировать следующим образом. Пользователи системы будут присылать платежные поручения, подписанные электронной цифровой подписью, которые будут поступать на сервер доступа и проверяться там. При положительном результате проверки ЭЦП в сервере баз данных в счетах пользователей-клиентов будут отображаться изменения, вызванные осуществлением платежей. Рабочие места локальных пользователей необходимы для управления системой, контролем за состоянием серверов и оборудования. Фронт-сервер отображает суммарную информацию по платежам, количество обработанных документов, очередность платежей и т.д.

Для аналитической компьютерной системы первой компонентой является сервер базы данных (сервер БД), он осуществляет хранение базы данных, аналитической и вспомогательной информации и обработку запросов, направляемых со стороны клиентских программных приложений. Понятие сервер обозначает компьютер, предназначенный для длительной работы без выключения и отличается от персональных

компьютеров повышенной надежностью работы и хранения данных. Сервер БД должен представлять собой высоконадежную аппаратную платформу под управлением сертифицированных версий операционных сред (ОС), размещаться в отдельном сегменте локальной вычислительной сети и физически в отдельном защищенном помещении. Это помещение должно удовлетворять следующим требованиям:

- отсутствие окон;
- электронный кодовый замок с автономным питанием;
- система видеонаблюдения с видеозаписью;
- климатическая система;
- система бесперебойного электропитания.

Второй компонентой аналитической КС является **сервер доступа**, который, будучи посредником, обеспечивает доступ удаленных пользователей-аналитиков к защищаемой базе данных по каналу, обеспечивающему защиту конфиденциальности и целостности передаваемых данных (защищенному криптографическими методами, либо проложенному так, чтобы исключить подключение к нему). Сервер доступа представляет собой промышленный высоконадежный компьютер с необходимым программным обеспечением.

Фронт–сервер является третьей компонентой аналитической КС. На фронт–сервере размещается ресурс с персональными кабинетами для доступа пользователей к аналитической информации, выполнения запросов и размещения результатов работ аналитиков в БД. Четвертой компонентой являются **рабочие места пользователей** ресурсом фронт–сервера и информации из внешних сетей, к которым подключен фронт сервер. Пятой компонентой являются **удаленные места пользователей** и аналитиков, подключенные непосредственно к серверу доступа.

Таким образом, в корпоративной аналитической КС можно выделить **два информационных контура** – внутренний, содержащий сервер БД и саму базу данных с результатами аналитических исследований, и внешний – рабочие места пользователей и точку взаимодействия с внешними информационными ресурсами – **фронт–сервер**.

С точки зрения модели КС внешние сети естественным образом соответствуют внешнему сегменту КС относительно рабочих мест пользователей и аналитиков, а также относительно сервера БД. Необходимо обеспечить изолированность этих контуров и потоков между ними. Изоляция в первую очередь подразумевает невозможность переноса информации из внутреннего информационного контура во внешний с целью обеспечения конфиденциальности результатов аналитических исследований. Далее необходимо обеспечить работоспособность сервера БД при активном воздействии со стороны внешнего сегмента (сохранность информации и невозможность ее изменения). И, наконец, необходимо обеспечить безопасное взаимодействие пользователей и аналитиков между собой и с серверами системы. Изолированность информационных контуров между собой обеспечивают средства **межсетевой защиты**.

8.3. Защита объектов при изменении их формы

В настоящее время с развитием и удешевлением копировально-множительной техники и развитием программных средств обработки изображений проблема защиты подлинности бумажных печатных документов, равно как проблема восприятия их электронной формы является весьма актуальной. С точки зрения компьютерной безопасности решается задача обеспечения непрерывности защиты при преобразовании документа или объекта КС из электронной в визуально читаемую материальную форму на бумажном или другом носителе.

Попробуем уточнить задачу защиты подлинности бумажного документа или, в общем случае, произвольного документа на некотором носителе. Принципиальное отличие структурированных документов типа банкнот от документов произвольной формы состоит именно в меняющемся содержании последних.

Необходимо рассматривать возможность контроля подлинности не листа бумаги или иного носителя, на котором готовится документ, а подлинности содержимого листа бумаги и, соответственно, говорить о защитном признаке (атрибуте) подлинности бумажного документа. Сформулируем, каким должен быть защитный признак бумажного документа.

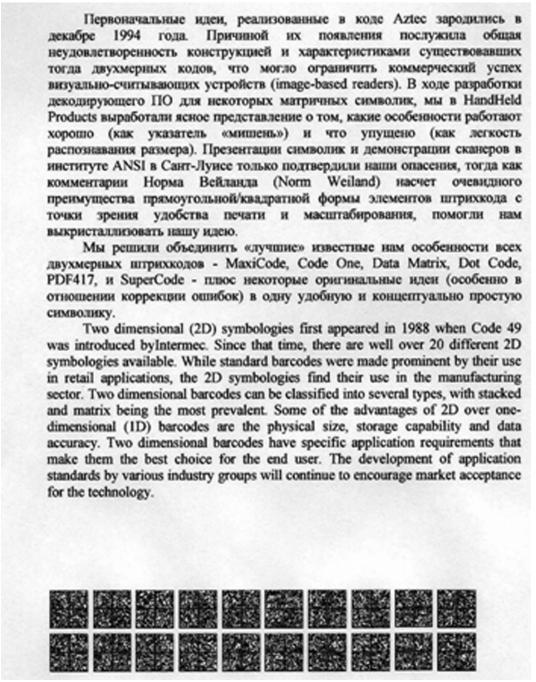
Во-первых, защитный признак должен быть неразрывно связан с тремя сущностями – с автором документа, с его содержанием и с основой документа. Собственно, подлинность и есть удостоверение авторства и содержания. Во-вторых, защитный признак должен быть проверяемым, т. е. практически любой человек должен иметь возможность убедиться в неизменности содержания.

Для электронных документов эту технологию, как известно, обеспечивает инфраструктура открытых ключей – когда электронный документ заверяется цифровой подписью, а проверка подлинности ведется при помощи заверенных уполномоченным органом сертификатов. Вполне логично распространить эту технологию на бумажные документы. Для того чтобы нанести цифровую подпись на документ воспользуемся технологией двумерного кодирования – когда информация наносится на носитель в виде матрицы из черных и белых квадратов. Матричный код обладает высокой информационной емкостью и устроен так, что может обнаруживать и исправлять ошибки при чтении, что важно для его практического применения.

Предположим, что имеется цифровой сертификат, который используется для заверения электронных документов электронной цифровой подписью (ЭЦП). Это означает, что пользователь обладает секретным ключом, которым он может подписывать сообщение или документ, и существует открытый ключ, заверенный уполномоченной организацией, которым возможно проверить подпись под электронным документом.

Теперь формируется образ бумажного документа и перед выводом его на печать создается маркер подлинности – совокупность архивиро-

ванного содержания документа, электронной цифровой подписи под ним и сертификата, позволяющего проверить электронную цифровую подпись, в виде матричного кода. Маркер подлинности выводится на печать вместе с текстом документа в свободном от текста поле документа. Таким образом, каждый бумажный документ сопровождается своим маркером подлинности. Примерный внешний вид такого документа приведен на ***рис. 2***.



Первоначальные идеи, реализованные в коде Aztec зародились в декабре 1994 года. Причиной их появления послужила общая неудовлетворенность конструкцией и характеристиками существовавших тогда двухмерных кодов, что могло ограничить коммерческий успех визуально-считывающих устройств (image-based readers). В ходе разработки декодирующего ПО для некоторых матричных символик, мы в HandHeld Products выработали ясное представление о том, какие особенности работают хорошо (как указатель «мишень») и что упущено (как легкость распознавания размера). Презентации символик и демонстрации сканеров в институте ANSI в Сант-Луисе только подтвердили наши опасения, тогда как комментарии Норма Вейланда (Norm Weiland) насчет очевидного преимущества прямоугольной/квадратной формы элементов штрихкода с точки зрения удобства печати и масштабирования, помогли нам выкристаллизовать нашу идею.

Мы решили объединить «лучшие» известные нам особенности всех двухмерных штрихкодов - MaxiCode, Code One, Data Matrix, Dot Code, PDF417, и SuperCode - плюс некоторые оригинальные идеи (особенно в отношении коррекции ошибок) в одну удобную и концептуально простую символику.

Two dimensional (2D) symbologies first appeared in 1988 when Code 49 was introduced byIntermec. Since that time, there are well over 20 different 2D symbologies available. While standard barcodes were made prominent by their use in retail applications, the 2D symbologies find their use in the manufacturing sector. Two dimensional barcodes can be classified into several types, with stacked and matrix being the most prevalent. Some of the advantages of 2D over one-dimensional (1D) barcodes are the physical size, storage capability and data accuracy. Two dimensional barcodes have specific application requirements that make them the best choice for the end user. The development of application standards by various industry groups will continue to encourage market acceptance for the technology.

Рис. 2. Вид документа с маркером подлинности

Теперь, прочитав маркер подлинности при помощи обычного сканера или специализированного устройства (функционально аналогичного тому, которым считывают штрих-коды с товаров), можно:

- быстро получить электронный образ документа, восстановив его из маркера подлинности;
- проверить цифровую подпись под ним;
- визуально сверить текстовое содержание документа с восстановленным из маркера подлинности и, следовательно, убедиться в неизменности документа;
- сканировав бумажный документ, автоматически сверить текстовое содержание документа с восстановленным из маркера подлинности и, следовательно, убедиться в неизменности документа.

Кроме того, механизм цифровой подписи точно определяет автора документа. Наконец, получаем у бумажных документов ряд полезных и оригинальных свойств: возможность проверять подлинность ксерокопий документов (поскольку и текст и маркер подлинности переносятся на копию), возможность передавать документы по факсу и проверять их подлинность на приеме.

Еще заметим, что в данном случае решается такая принципиальная проблема, как обеспечение *непрерывности* защиты электронного и бумажного документооборота. Электронная подпись под бумажным документом легко переходит в электронный и наоборот.

Маркер подлинности обладает следующими достоинствами:

• имеет низкую стоимость (она равна стоимости красящего состава, потраченного на печать двумерного кода, либо стоимости ламинированной самоклеящейся этикетки, если маркер наносится на нее);

• позволяет восстановить и проверить подлинность всех данных документа автоматизированным способом (при помощи обычного бытового сканера или фотокамеры мобильного телефона);

• содержит в себе электронную цифровую подпись (ЭЦП) всего заверенного им документа и не может быть подделан лицом, не владеющим секретным ключом;

• его нельзя испортить скрытно (в отличие от радиометок и смарт-карт) – любые намеренно внесенные искажения визуально видны, но при этом они исправляются при чтении маркера за счет применения кодов, исправляющих ошибки;

• опирается на российские сертифицированные средства и технологии.

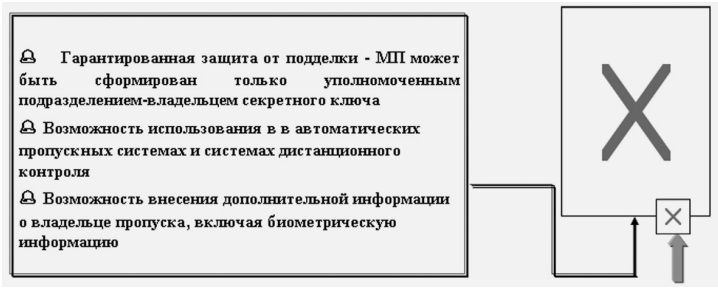


Рис. 3. Защита пропусков на режимные объекты

Описанная технология может быть применена при изготовлении пропусков (в том числе и содержащих фото- и биометрическую информацию), защите подлинности удостоверительных (паспорт, удостоверение личности) и правоустанавливающих документов (доверенность, свидетельство о праве на собственность). При этом маркер подлинности можно внести в любой действующий документ без изменения его формы и правил пользования им (внутренний паспорт, заграничный паспорт, включая визы и визовые отметки, водительские права и т. д.) в виде ла-

минированной наклейки или напечатав его непосредственно на бланке документа. После этого можно будет быстро (автоматизированным способом) и безошибочно считать и проверить подлинность всех установочных данных владельца документа, что устраняет необходимость ручного заполнения полей документа и позволяет полностью исключить ошибки при вводе. И, безусловно, в этом случае мы решаем поставленную в самом начале принципиальную проблему обеспечения непрерывности защиты электронного и бумажного документооборота. Маркер подлинности бумажного документа сохраняется и в электронном документе, и, наоборот, электронная подпись сохраняется в бумажной копии документа, сопровождая его на всех этапах жизненного цикла.

Весьма интересной является идея использования маркера для дистанционной продажи билетов или выдачи пропусков (***рис. 3***). Дело в том, что маркер легко воспроизводится на экранах мобильных устройств и может быть выведен на печать в домашних условиях.

8.4. Защищенный документооборот

В настоящее время категории «электронный документ» и «электронный документооборот» не имеют единого понимания среди IT-специалистов. Еще больше споров вызывает понятие «защищенный документооборот». Такая ситуация объективно объясняется тем, что документ как сущность существует в различных формах не обособленно, а является элементом некоторой технологии, которая предусматривает движение и преобразование документов, которые и целесообразно называть документооборотом.

Таким образом, документ отражает появление, движение, преобразование и исчезновение некоторых сведений, облеченных в ту или иную форму (для простоты условно их может быть две – бумажная и электронная). Соответственно, документооборот как технология содержит некоторые повторяющиеся операции с документами в различных формах. Свойство защищенности документооборота может складываться из классических категорий безопасности «конфиденциальность – целостность – надежность» в проекции их как на сам документ, так и на технологию его движения.

Технологию движения целесообразно рассматривать как жизненный цикл документов, определяемых целевой функцией системы, в которой существуют документы.

Так, например, система платежного документооборота будет обслуживать совершение безналичных денежных расчетов путем движения документов – распоряжений по счету, создаваемых в бумажном виде физическими и юридическими лицами, далее преобразуемых в электронную форму в системе организации, уполномоченной осуществлять безналичные операции, и в конечном итоге отражающих изменения на счетах клиентов кредитных организаций. В первую очередь защищенность такой системы должна рассматриваться с точки зрения соблюдения юридическо-правовых норм, регламентирующих функциональность

системы, т. е. в данном случае – проведение платежей. Следовательно, понятие «защищенность» здесь является регламентацией более высокого уровня по отношению к самой системе. Однако правовые нормы могут быть только опосредованно отражены в такой системе в виде средств их реализации разного уровня.

Таким образом, под *системой защищенного документооборота* (СЗДО) будем понимать автоматизированную информационная система (АКС) совместно с комплексом организационных, методических и нормативно-распорядительных мер, которые предназначены для ввода, обработки, хранения, учета, контроля движения, исполнения, выполнения иных функций, предусмотренных в СЗДО для документов различного назначения, с соблюдением свойств их защищенности (целостности, конфиденциальности, доступности), определяемых регламентацией высших уровней на всех этапах жизненного цикла документов, определенного технологией их обработки в указанной АКС.

Целесообразно выделить три класса СЗДО по их позиционированию в системе общественных отношений – предназначенные для индивидуального пользования (Д1), для корпоративного бизнес-пользования (Д2) и для органов государственной власти (ОГВ) и местного самоуправления (ОМС). При такой классификации системы класса Д1 будут предназначены для использования индивидуальными пользователями, которые применяют модель злоумышленника H_2 – внутренний нарушитель, не являющийся пользователем средств вычислительной техники (СВТ), на которых реализована АКС. Для индивидуального пользователя такой выбор модели злоумышленника единственно верен, поскольку для него все потенциальные злоумышленники (нарушители) являются внешними, отличными от него.

Системы класса Д2 предназначены для использования корпорациями и бизнес-консорциумами, которые применяют модель злоумышленника H_3 – внутренний нарушитель, являющийся пользователем СВТ, на которых реализована АКС СЗДО. Такой выбор обусловлен разумными предположениями о возможности злоумышленных действий кого-либо из сотрудников корпорации.

Системы класса Д3 предназначены для использования территориально-распределенными крупными корпорациями и ОГВ (ОМС) уровня министерства (ведомства, федеральной службы (агентства)), которые применяют модель злоумышленника H_4 – группа нарушителей (среди которых есть внутренние, являющиеся пользователями СВТ, на которых реализована АКС СЗДО), осуществляющая создание методов и средств реализации атак, а также реализующая атаки с привлечением отдельных специалистов, имеющих опыт разработки и анализа средств защиты.

Архитектурно, независимо от позиционирования по классу применения, СЗДО может представлять собой отдельное рабочее место (компьютер), ЛВС, размещенную в пределах одного здания (сооружения), территориально-распределенную АКС, иметь отношения информационного обмена с другими СЗДО и информационными ресурсами глобальной сети или ее национальных доменов. В СЗДО могут выделяться

клиентские места мобильных и стационарных пользователей, а также некоторая серверная компонента, предназначенная для хранения данных и реализации технологий, общих для всех пользователей.

Поскольку в настоящее время разработано значительное количество различных регламентирующих требований уполномоченных органов (ФСБ и ФСТЭК), определяющих требования к АКС и их компонентам, то часть требований, отнесенная к классам, может носить прямой отсылочный характер, т. е. содержать ссылки на конкретные нормы и классы требований ФСБ и ФСТЭК. Например, это могут быть *требования к криптографическому ядру и требования к защите от НСД.*

Другая группа требований может носить опосредованно отсылочный характер, т. е. ссылаться на документы, касающиеся требований к технологиям, в частности, требований к функционированию Удостоверяющего центра (центров) в том случае, если он необходим для работы СЗДО некоторого класса. Это могут быть *требования к управлению и аудиту.*

Требования прямого формулирования, связанные с особенностью работы и необходимостью защиты технологии именно для СЗДО, – это *требования к взаимодействию компонент, к взаимодействию с другими СЗДО, к преобразованию (конвертации) документов, к архивному хранению документов в СЗДО, к документации.*

Дальнейшее направление работ состоит в соотнесении классов и групп требований, которое логично может быть выполнено на основе сформулированных выше исходных посылок. Выделяют три класса СЗДО – Д1, Д2 и Д3.

Системы класса Д1 предназначены для использования индивидуальными пользователями и небольшими корпорациями, которые применяют модель злоумышленника H_2 – это внутренний нарушитель, не являющийся пользователем средств вычислительной техники (СВТ), на которых реализована АКС СЗДО (группа нарушителей, среди которых есть по крайней мере один указанный выше внутренний нарушитель), самостоятельно осуществляющий (-ая) создание методов и средств реализации атак, а также самостоятельно реализующий (-ая) атаки.

Системы класса Д2 предназначены для использования региональными ОГВ и крупными корпорациями, которые применяют модель злоумышленника H_3 – это внутренний нарушитель, являющийся пользователем СВТ, на которых реализована АКС СЗДО (группа нарушителей, среди которых есть, по крайней мере, один указанный выше внутренний нарушитель), самостоятельно осуществляющий (-ая) создание методов и средств реализации атак, а также самостоятельно реализующий (-ая) атаки.

Системы класса Д3 предназначены для использования территориально-распределенными крупными корпорациями и ОГВ уровня министерства (ведомства, федеральной службы (агентства)), которые применяют модель злоумышленника H_4 – группа нарушителей (среди которых есть внутренние, являющиеся пользователями СВТ, на которых реализована АКС СЗДО), осуществляющая создание методов и средств реа-

лизации атак, а также реализующая атаки с привлечением отдельных специалистов, имеющих опыт разработки и анализа средств защиты.

СЗДО может представлять собой отдельное рабочее место (компьютер), ЛВС, размещенную в пределах одного здания (сооружения), территориально-распределенную АКС, иметь отношения информационного обмена с другими СЗДО и информационными ресурсами глобальной сети или ее национальных доменов. В СЗДО могут выделяться клиентские места мобильных и стационарных пользователей и серверная компонента.

Требования к криптографическому ядру

Класс Д1. В систему должно быть интегрировано криптографическое ядро, сертифицированное по классу не ниже КС2 по «Требованиям к средствам криптографической защиты конфиденциальной информации». Все криптографические функции, реализованные в СЗДО, должны вызываться (экспортироваться) из интегрированного криптографического ядра.

Класс Д2. В систему должно быть интегрировано криптографическое ядро, сертифицированное по классу не ниже КС2 по «Требованиям к средствам криптографической защиты конфиденциальной информации» для клиентской части и по классу не ниже КС3 для серверной части. Все криптографические функции, реализованные в СЗДО, должны вызываться (экспортироваться) из интегрированного криптографического ядра. Все криптографические функции клиентской и серверной части должны быть встречно совместимы между собой.

Класс Д3. В систему должно быть интегрировано криптографическое ядро, сертифицированное по классу не ниже КС2 по «Требованиям к средствам криптографической защиты конфиденциальной информации» для мобильной клиентской части, по классу не ниже КС3 для стационарных клиентских мест и по классу не ниже КВ1 для серверной части. Все криптографические функции, реализованные в СЗДО, должны вызываться (экспортироваться) из интегрированного криптографического ядра. Все криптографические функции клиентской (мобильной и стационарной) и серверной частей должны быть встречно совместимы между собой.

Требования к защите от НСД

Класс Д3. В СЗДО должна быть реализована система защиты от НСД, соответствующая классу не ниже АК2 «Временных требований к защите от НСД к конфиденциальной информации в АКС, расположенных на территории РФ». Применение средств АПМДЗ обязательно в том случае, когда средства защиты от НСД являются наложенными и не интегрированы в сертифицированную по указанному классу операционную платформу. Дополнительно должно быть реализовано дискреционное управление (разграничение) доступа субъектов (пользователей) СЗДО к объектам СЗДО. В случае реализации СЗДО данного класса в раздельном исполнении как совокупности клиентской и серверной частей штатная работа СЗДО должна быть обеспечена таким образом, чтобы на клиентских местах работали только пользователи, не имеющих административных прав.

Класс Д2. В СЗДО должна быть реализована система защиты от НСД, соответствующая классу не ниже АК2 «Временных требований к защите от НСД к конфиденциальной информации в АКС, расположенных на территории РФ» для клиентских рабочих мест и классу АК3 для серверной части. Применение средств АПМДЗ обязательно для серверной части и/или для клиентской части в том случае, когда средства защиты от НСД клиентских рабочих мест являются наложенными и не интегрированы в сертифицированную по указанному классу операционную платформу. В случае реализации СЗДО данного класса в раздельном исполнении как совокупности клиентской и серверной частей штатная работа СЗДО должна быть обеспечена таким образом, чтобы на клиентских местах работали пользователи, не имеющих административных прав. При реализации средств удаленного управления средствами защиты от НСД для СЗДО данного класса должен использоваться выделенный или криптографически защищенный канал.

Класс Д3. В СЗДО должна быть реализована система защиты от НСД, соответствующая классу не ниже АК2 «Временных требований к защите от НСД к конфиденциальной информации в АКС, расположенных на территории РФ» для мобильных рабочих мест, АК3 для стационарных клиентских рабочих мест и классу АК4 для серверной части. Применение средств АПМДЗ обязательно для стационарной клиентской и серверной частей. В случае реализации СЗДО данного класса в раздельном исполнении как совокупности клиентской и серверной частей штатная работа СЗДО должна быть обеспечена таким образом, чтобы на клиентских местах работали пользователи, не имеющие административных прав. Права пользователей мобильных рабочих мест по доступу к ресурсам (объектам) СЗДО должны быть минимальны. При реализации средств удаленного управления средствами защиты от НСД для СЗДО данного класса должен использоваться выделенный или криптографически защищенный канал. Администрирование средств защиты от НСД для СЗДО данного класса с мобильных рабочих мест не допускается.

Требования к взаимодействию компонент

Класс Д1. Удаленные (размещенные не в рамках одного компьютера) компоненты СЗДО должны обмениваться данными, целостность которых зафиксирована с применением средств ЭЦП. Применение средств шифрования для обеспечения конфиденциальности взаимодействия удаленных компонент является опциональным. Взаимодействие удаленных компонент СЗДО данного класса должно происходить по стандартным либо оригинальным документированным интерфейсам (протоколам). Использование для информационного обмена сетей общего пользования регулируется формулярами (условиями применения) интегрированных в СЗДО указанного класса средств защиты информации.

Класс Д2. Требования включают требования предыдущего класса. Дополнительно требуется обязательное использование шифрования для обеспечения конфиденциальности взаимодействия удаленных компонент. При взаимодействии компонент СЗДО через сети общего

пользования, необходимо использование межсетевого экрана, сертифицированного по классу не ниже 4-го по «Требованиям к межсетевым экранам».

Класс Д3. Требования совпадают с требованиями предыдущего класса, за исключением того, что при взаимодействии компонент СЗДО данного класса через сети общего пользования необходимо использование межсетевого экрана, сертифицированного по классу не ниже 3-го по «Требованиям к межсетевым экранам».

Требования к взаимодействию с другими СЗДО

Класс Д1. Требования не предъявляются.

Класс Д2. СЗДО должна поддерживать получение электронных документов от других СЗДО с возможностью проверки ЭЦП под данными документами. Выполнение данного требования допускается путем преобразования передаваемых (получаемых) документов в единый формат обмена. Дополнительно получаемые документы должны проходить структурный (на соответствие формату) и антивирусный контроль с использованием сертифицированных антивирусных средств (АВС).

Класс Д3. Включает требования предыдущего класса. Дополнительно: ввод в систему документов с неверной ЭЦП запрещается; допускается ввод в СЗДО данного класса только таких документов, в которых отсутствуют исполняемые (интерпретируемые) фрагменты; требуется выделение отдельного рабочего места для получения документов от других СЗДО.

Требования к преобразованию (конвертации) документов

Класс Д1. Процедуры преобразования (конвертации) документов, обрабатываемых и хранимых в СЗДО данного класса, должны быть организованы таким образом, чтобы имелась возможность проверки ЭЦП под документами независимо от их форматов.

Класс Д2. Требования включают требования предыдущего класса. Дополнительно при преобразовании документов в бумажную форму обязательно нанесение на них маркера подлинности, включающего, как минимум, ЭЦП текста документа и, при необходимости, сам текст документа и сертификат для проверки ЭЦП. Возможно дополнительное нанесение на бумажную форму документов дополнительных признаков, обеспечивающих их учет. Должна быть предусмотрена возможность ввода бумажных документов в СЗДО с предварительным контролем ЭЦП под ними с использованием информации из маркера подлинности бумажного документа перед преобразованием в электронную форму.

Класс Д3. Требования совпадают с требованиями предыдущего класса.

Требования к архивному хранению документов в СЗДО.

Класс Д1. Требования не предъявляются.

Класс Д2. При архивном хранении документов в СЗДО они должны снабжаться и ЭЦП и меткой времени (time stamp), также заверенной ЭЦП. Хранение документов должно быть обеспечено в течение сроков, определенных действующим законодательством РФ, но не менее трех лет.

Класс Д3. Требования совпадают с требованиями предыдущего класса.

Требования к управлению

Класс Д1. СЗДО должна использовать механизмы управления ключами и сертификатами, опирающиеся на применение Удостоверяющего центра (УЦ), сертифицированного по классу не ниже КС2 «Требований к удостоверяющим центрам». УЦ для управления СЗДО данного класса может быть внешним.

Класс Д2. СЗДО должна использовать механизмы управления ключами и сертификатами, опирающиеся на применение Удостоверяющего центра, сертифицированного по классу не ниже КС3 «Требований к удостоверяющим центрам». УЦ для управления СЗДО данного класса может быть внешним и аккредитованным Федеральным уполномоченным органом.

Класс Д3. СЗДО должна использовать механизмы управления ключами и сертификатами, опирающиеся на применение Удостоверяющего центра, сертифицированного по классу не ниже КВ1 «Требований к удостоверяющим центрам». УЦ для управления СЗДО данного класса должен быть ведомственным и аккредитованным Федеральным уполномоченным органом.

Требования к аудиту

Класс Д1. Требования не предъявляются.

Класс Д2. В СЗДО должны быть реализованы меры контроля корректного функционирования прикладной компоненты СЗДО, обеспечивающей документооборот и средств защиты информации, интегрированных в СЗДО. Перечень и периодичность контроля согласуется с экспертной организацией в техническом задании на разработку и/или сертификацию СЗДО.

Класс Д3. Требования совпадают с требованиями предыдущего класса.

Требования к документации

Класс Д1. Документация на СЗДО должна включать документацию на все интегрированные в нее средства защиты информации.

Класс Д2. Требования включают требования предыдущего класса. Дополнительно требуется наличие руководства администратора информационной безопасности.

Класс Д3. Требования включают требования предыдущего класса. Дополнительно требуется наличие регламента применения СЗДО с интегрированными средствами защиты информации и профиля СЗДО, включающего рекомендации по настройке СЗДО с интегрированными средствами защиты информации на конкретном объекте использования (применения).

8.5. Электронная коммерция

Все задачи, возникающие при дистанционном осуществлении экономических отношений с использованием распределенных КС, можно разделить на два класса: «горизонтальные» и «вертикальные».

«Горизонтальные» задачи – это взаимоотношения между юридическими лицами, участвующими в различных сферах деловой деятельности: предприятиями – участниками рынка товаров и услуг, государственными органами и структурами, биржами, банками, фондами, страховыми компаниями и другими кредитно-финансовыми учреждениями. В англоязычной литературе такие отношения имеют обозначение B2B – business-to-business.

«Вертикальные» задачи – это взаимоотношения между государственными органами и частными предприятиями, а также физическими лицами, между поставщиками и потребителями, между продавцами и покупателями, т. е. всякие отношения, характеризующиеся неравноправностью, когда одна из сторон в целом определяет характер и правила взаимоотношений, диктует условия, параметры, исходную информацию для совершения коммерческой сделки или юридического действия. В англоязычной литературе такие отношения имеют обозначение B2C – business-to-consumer.

Можно выделить два уровня рассмотрения задач электронной коммерции с точки зрения обеспечения безопасности информации. Первый – это *единичные информационные взаимодействия* субъектов коммерческих отношений. Второй – это *процессы деловой деятельности (бизнес-процессы)* между субъектами коммерческих отношений, рассматриваемые в целом.

Начнем с первого уровня. В электронной коммерции рассматривают две базовые формы единичных информационных взаимодействий: односторонняя передача данных (transfer) и электронный обмен данными (exchange). Обе формы, рассматриваемые в совокупности, в англоязычной литературе нередко обозначают термином «electronic data interchange».

Под *односторонней передачей данных* понимают передачу одного или более фрагментов информации (данных, документов, платежей) от одной стороны к одному или нескольким другим получателям. В реальности одна логическая процедура односторонней передачи данных может потребовать нескольких физических пересылок сообщений: например, вся информация еще не известна отправителю в момент отсылки первого фрагмента данных, или разные фрагменты необходимо разослать по разным адресам, или требуется широковещательная рассылка одного фрагмента. Когда речь идет об односторонней передаче данных, очень важно иметь в виду, что каждый передаваемый фрагмент данных или каждая отдельная передача одного и того же фрагмента может быть индивидуально ассоциирована с различными атрибутами защиты (конфиденциальность, целостность, неотказуемость, подлинность и т. д.).

В электронной коммерции приходится иметь дело с тремя принципиально различными видами информации, пересылаемой между участниками протоколов:

• обычные электронные данные (не имеющие юридической значимости);

• подписанные электронные документы, имеющие юридический статус: платежные поручения, ордера, квитанции, расписки, контракты, сертификаты, лицензии и др.;

• «электронные деньги».

Таким образом, проблема защиты односторонней передачи информации в электронной коммерции оказывается далеко не такой простой, как может показаться на первый взгляд. Попытки ее решения порождают множество других задач, как-то:

• образование защищенных (секретных и/или аутентичных) логических каналов связи;

• широковещательное шифрование информации;

• защита авторских прав на электронные данные (защиты от несанкционированного копирования), в том числе отслеживание «пиратского» копирования данных;

• учет использования ресурсов участниками протоколов и др.

Когда говорят об электронном *обмене данными*, имеется в виду группировка нескольких односторонних передач данных для представления семантики неделимой операции, выполняемой двумя или более сторонами (участниками протокола). Основная и важнейшая задача электронного обмена данными – обеспечить гарантии честности обмена, т. е. невозможности только одностороннего выполнения обязательств и обмана одним участником протокола другого.

Рассматривая те же три вида информации (обычные данные, электронные документы, электронные деньги), получаем шесть типов обмена информацией (табл. 5.1):

• *обмен обычными данными*;

• обмен информации на деньги, т. е. *покупка информации*;

• обмен подписанных документов на информацию, т. е. *доступ к информации по условию*, при подписании какого-либо обязательства (conditional access);

• обмен подписанными документами, который может осуществляться в режиме реального времени либо в отложенном режиме (в общем случае обеспечение честности этой процедуры подразумевает решение задачи *одновременного подписания контракта*);

• обмен, при котором одна сторона передает другой электронные деньги, а в ответ передаются электронные документы, представляет собою *платеж с квитированием* (подтверждением об оплате);

• *обмен валюты* (возможно, представленной в электронном виде).

Таблица 1. Классификация задач электронного обмена данными

6 типов обмена:	3. Электронные деньги	2. Электронные документы	1. Электронные данные
1. Электронные данные	Покупка информации	Доступ к информации по условию	Обмен информацией
2. Электронные документы	Платеж с квитированием	Сертифицированная электронная почта Одновременное подписание контракта	–
3. Электронные деньги	Обмен валюты	–	–

Заметим, что среди перечисленных задач наибольший интерес с точки зрения обеспечения безопасности обмена данными представляют третий и четвертый случаи. Они порождают еще одну задачу, занимающую промежуточное положение между ними, – это так называемая задача о *сертифицированной электронной почте*. Она заключается в построении таких протоколов доставки электронной почты и – на их основе – такой системы электронной почты, которая обеспечивала бы невозможность отказа получателя от факта ознакомления с сообщением, отправителя сообщения – от факта его отправки, а самой системы – от факта принятия сообщения к доставке. Этим задачам в настоящее время уделяется очень большое внимание среди общего поля проблем электронной коммерции и электронного документооборота.

Второй уровень рассмотрения задач электронной коммерции с точки зрения обеспечения безопасности информации – это процессы деловой деятельности. Для решения задач обеспечения безопасности этого уровня применяется концепция транзакций, или дел (deal). Это последовательность шагов, цель которой – осуществить целостный бизнес-процесс, поддерживая взаимосвязь между отдельными его этапами по мере их выполнения. Ход выполнения транзакции записывается всеми участвующими в ней сторонами, причем запись включает получаемую и отправляемую информацию, а также общую информацию о процессе: согласованные атрибуты защиты, идентификаторы участников, логические связи между этапами транзакции и др. Все эти сведения являются основой для обработки исключительных ситуаций и разрешения споров и конфликтов, возникающих между участниками коммерческих отношений.

Примерами процессов деловой деятельности, для которых обеспечение безопасности информации требует целостного их рассмотрения, являются: электронные аукционы, электронные выборы (голосования), электронные биржи, электронные лотереи и т. п. процессы. Для всех них требуются специальные криптографические протоколы. Очень большую роль в обеспечении безопасности бизнес-процессов играют также двусторонние и многосторонние схемы цифровой подписи со специальными свойствами. Примером может являться оформление прав на недвижимое имущество, требующее, как известно, очень большого количества решений и согласований, а также оформление или регистрация каких-либо иных прав или обязанностей субъектов электронного рынка.

Для решения всех этих довольно непростых задач необходима разработка архитектуры информационных систем для электронной коммерции. Она будет являться базисом реализации целей, которые ставят электронная коммерция и электронный обмен данными, и сможет в конечном итоге обеспечить многостороннюю безопасность участников коммерческих и государственно-правовых отношений.

Одна из таких архитектур, наиболее развитая, – архитектура SEMPER (Secure Electronic Marketplace for Europe), разработанная в рамках общеевропейского проекта единого электронного рынка (***рис. 4***).

Спецификация SEMPER вводит модельное представление информационных систем для поддержки электронной коммерции, в которые введены, в том числе и функции обеспечения многосторонней безопасности. Она позволяет достичь следующих основных целей:

• реализовать множество сценариев бизнес-процессов;

• определить открытую платформу реализации функций безопасности;

• гарантировать безопасное управление пользователями своей информацией.

Рис. 4. Архитектура SEMPER

Архитектура SEMPER – пятиуровневая. Самый верхний уровень составляют бизнес-приложения – прикладные программы пользователей, выполняющие функции, специфические для электронной коммерции, которые пользуются услугами нижележащих уровней архитектуры.

Коммерческий уровень предоставляет бизнес-приложениям сервисы защиты и обеспечивает ориентацию бизнес-процессов, записывая статус и текущее состояние бизнес-процессов, включая открытие новых «дел», доступ к ним и отображение отдельных шагов бизнес-процесса, навигацию пользователей внутри отдельных «дел» и между «делами», экспорт «дел». Требования по безопасности «дел» отображаются посредством введения атрибутов защиты, относящихся либо к отдельным шагам бизнес-процесса (как, например, конфиденциальность), либо ко всему «делу» в целом (например, анонимность). Коммерческий уровень опирается на следующий – уровень передачи и обмена информацией, который реализует функции, связанные с односторонней передачей данных и обменом данными, включая и необходимые функции по защите этих процессов. Этот уровень, в свою очередь, опирается на уровень субъектов деловой деятельности, который управляет передачей сообщений сообразно с природой передаваемых и принимаемых данных: обычные данные, электронные документы, электронные деньги. Этот уровень имеет значение только для тех протоколов, в которых существенна природа пересылаемой информации. Сервисы поддержки включают доверенный пользовательский интерфейс, различные вспомогательные сервисы и инфраструктуру для криптографической защиты и

обеспечения безопасности каналов связи, локального учета и архивации данных пользователей.

Функции безопасности, предусмотренные моделью SEMPER, таким образом, оказываются структурированы по трем уровням: коммерческому, передачи и обмена информацией, уровню субъектов деловой деятельности. Необходимым условием применения функций защиты в архитектуре SEMPER является наличие соответствующего правового обеспечения.

8.6. Защита объектов интеллектуальной собственности

В настоящее время объективно назрела необходимость новой трактовки понятия «интеллектуальная собственность» (ИС), объединяющей практически все информационно-содержащие объекты КС (включая и «электронные деньги») и связанные с ними отношения, возникающие в производственной, финансовой, общественной, институционально-государственной и персональной сферах.

В современном научном обороте используют также термин «информационная собственность» с преобладающей «электронной» формой представления. Необходимо отметить, что этот термин является практически эквивалентным интеллектуальной собственности.

Интеллектуальная собственность порождается практически на всех этапах общественного производства, сопровождая появление и движение материальных ценностей. Исходя из реально существующего верхнего уровня объектов интеллектуальной собственности (федеральная ИС), целесообразно принять трехуровневое или трехзвенное деление «федеральная ИС» – «корпоративная ИС» – «индивидуальная ИС».

В текущем правовом толковании ИС представляет совокупность исключительных прав как личного, так и имущественного характера на результаты творческой деятельности человека, а также на некоторые иные приравненные к ним объекты, конкретный перечень которых устанавливается законодательством соответствующей страны с учетом принятых ею международных обязательств. Легко видеть, что это определение вполне допускает указанные выше расширения поля применения.

Исходя из постулата защиты прав добросовестного потребителя, необходимо совершенствовать систему защиты объектов КС, постепенно уходя от прямых технических мер защиты от копирования и иных технических ограничений в область более свободной циркуляции объектов КС, снабженных проверяемыми защитными атрибутами, в первую очередь ЭЦП или КА.

Сформулированные выше технологические и инфраструктурные подходы к защите КС позволяют глобализовать защиту КС, вписывая ее в новую систему отношения «личность – личность» и «личность – государство», а с другой – сделать ее технически «прозрачной» и реализуемой. В этой системе индивидуум, выступающий как автор, владелец и потребитель объектов КС, персонализируется своим

личным секретным (персональным) ключом (ЛСК), который существует только внутри защищенного хранилища, защищающего ЛСК как от несанкционированного доступа, так и от методов социальной инженерии. Индивидуум передает информацию о своем ЛСК в виде открытого ключа или запроса на сертификат, который воспринимается государством в лице своих уполномоченных органов и заверяется УЦ электронной цифровой подписью, образуя личный сертификат.

Таким образом, государственные институты признают индивидуума как владельца неотчуждаемой и никому неизвестной персональной информации. Сертификат и личный секретный ключ создают основу для обмена между индивидуумами различными объектами КС с обеспечением авторства, целостности и конфиденциальности. При помощи маркеров подлинности решается также проблема защиты объектов КС, существующих в «твердых копиях» и печатных формах.

С другой стороны, индивидуум таким же образом общается с государственными институтами, проверяя их управомоченность государством и выступает перед ними как полностью идентифицируемая личность, чьи права и статус также подтверждены государством. Государство может также придать некоторым объектам КС статус мер стоимости, накопления и обмена, инициируя принципиально новую экономическую сущность – «электронные деньги». Все это позволяет создать технологическую основу не только для защиты объектов КС, но и для принципиально новых экономических и социальных отношений.

8.7. Аудит компьютерной безопасности

Аудит в общем понимании представляет собой независимую экспертизу отдельных областей функционирования организации.

Аудит информационной безопасности заключается в организации периодического, независимого и документированного процесса получения свидетельств аудита и объективной их оценки с целью установления степени выполнения установленных требований по обеспечению информационной безопасности.

Внутренние аудиты («аудиты первой стороной») проводятся самой организацией или от ее имени для анализа менеджмента или других внутренних целей и могут служить основанием для самодекларации организации о соответствии требованиям по ИБ.

Внешние аудиты включают «аудиты второй стороной» и «аудиты третьей стороной». Аудиты второй стороной проводятся сторонами, заинтересованными в деятельности организации, например, потребителями или другими лицами от их имени. Аудиты третьей стороной проводятся внешними независимыми организациями.

Независимость при аудите предполагает полную свободу аудитора (самостоятельность) в отборе и анализе свидетельства аудита (изложение фактов или другой информации, связанной с критериями аудита) в отношении объекта аудита.

Итак, принципиально различают внешний и внутренний аудит. Внешний аудит – это, как правило, разовое мероприятие, проводимое по

инициативе руководства организации или акционеров. Рекомендуется проводить внешний аудит регулярно, для многих финансовых организаций и акционерных обществ это является обязательным требованием. Внутренний аудит представляет собой непрерывную деятельность, которая осуществляется на основании «Положения о внутреннем аудите» и в соответствии с планом, подготовка которого осуществляется подразделением внутреннего аудита и утверждается руководством организации. Аудит безопасности информационных систем является одной из составляющих общего аудита.

Целями проведения аудита компьютерной безопасности являются:

- анализ рисков, связанных с возможностью осуществления угроз безопасности в отношении ресурсов и объектов КС;
- оценка текущего уровня защищенности КС;
- локализация узких мест в системе защиты КС;
- оценка соответствия КС существующим стандартам в области информационной безопасности;
- выработка рекомендаций по внедрению новых и повышению эффективности существующих механизмов безопасности КС.

Этим в основном и исчерпывается набор целей проведения аудита безопасности, но только в том случае, если речь идет о внешнем аудите. В число дополнительных задач, стоящих перед внутренним аудитором, помимо оказания помощи внешним аудиторам, могут также входить:

- разработка политик безопасности и других организационно-распорядительных документов по защите информации и участие в их внедрении в работу организации;
- постановка задач для ИТ персонала, касающихся обеспечения защиты информации;
- участие в обучении пользователей и обслуживающего персонала КС вопросам обеспечения информационной безопасности;
- участие в разборе инцидентов, связанных с нарушением информационной безопасности.

Необходимо отметить, что все перечисленные выше «дополнительные» задачи, стоящие перед внутренним аудитором, за исключением участия в обучении, по существу аудитом не являются. Аудитор по определению должен осуществлять независимую экспертизу реализации механизмов безопасности в организации, что является одним из основных принципов аудиторской деятельности. Если аудитор принимает деятельное участие в реализации механизмов безопасности, то независимость аудитора утрачивается, а вместе с ней утрачивается и объективность его суждений, т.к. аудитор не может осуществлять независимый и объективный контроль своей собственной деятельности. Однако, на практике, внутренний аудитор, порой, являясь наиболее компетентным специалистом в организации в вопросах обеспечения информационной безопасности, не может оставаться в стороне от реализации механизмов защиты.

Деятельное участие во внедрении той же подсистемы аудита безопасности, которая смогла бы предоставлять аудитору исходные данные

для анализа текущей ситуации, он принять может и должен. Конечно, в этом случае, аудитор уже не сможет объективно оценить реализацию этой подсистемы и она естественным образом выпадает из плана проведения аудита. Точно также, внутренний аудитор может принять деятельное участие в разработке политик безопасности, предоставив возможность оценивать качество этих документов внешним аудиторам.

Аудит проводится не по инициативе аудитора, а по инициативе руководства компании, которое в данном вопросе является основной заинтересованной стороной. Поддержка руководства компании является необходимым условием для проведения аудита.

Аудит представляет собой комплекс мероприятий, в которых помимо самого аудитора, оказываются задействованными представители большинства структурных подразделений компании. Действия всех участников этого процесса должны быть скоординированы. Поэтому на этапе инициирования процедуры аудита должны быть решены следующие организационные вопросы:

• права и обязанности аудитора должны быть четко определены и документально закреплены в его должностных инструкциях, а также в положении о внутреннем (внешнем) аудите;

• аудитором должен быть подготовлен и согласован с руководством план проведения аудита;

• в положении о внутреннем аудите должно быть закреплено, в частности, что сотрудники компании обязаны оказывать содействие аудитору и предоставлять всю необходимую для проведения аудита информацию.

На этапе инициирования процедуры аудита должны быть определены границы проведения обследования. Одни информационные подсистемы компании не являются достаточно критичными и их можно исключить из границ проведения обследования. Другие подсистемы могут оказаться недоступными для аудита из-за соображений конфиденциальности.

Границы проведения обследования определяются в следующих терминах:

1. Список обследуемых физических, программных и информационных ресурсов;

2. Площадки (помещения), попадающие в границы обследования;

3. Основные виды угроз безопасности, рассматриваемые при проведении аудита;

4. Организационные (законодательные, административные и процедурные), физические, программно-технические и прочие аспекты обеспечения безопасности, которые необходимо учесть в ходе проведения обследования, и их приоритеты (в каком объеме они должны быть учтены).

План и границы проведения аудита обсуждается на рабочем собрании, в котором участвуют аудиторы, руководство компании и руководители структурных подразделений.

Этап сбора информации аудита, является наиболее сложным и длительным. Это связано с возможным отсутствием необходимой докумен-

тации на информационную систему и с необходимостью плотного взаимодействия аудитора со многими должностными лицами организации.

Компетентные выводы относительно положения дел в компании с информационной безопасностью могут быть сделаны аудитором только при условии наличия всех необходимых исходных данных для анализа. Получение информации об организации, функционировании и текущем состоянии КС осуществляется аудитором в ходе специально организованных интервью с ответственными лицами компании, путем изучения технической и организационно-распорядительной документации, а также исследования КС с использованием специализированного программного инструментария. Остановимся на том, какая информация необходима аудитору для анализа.

Обеспечение информационной безопасности организации – это комплексный процесс, требующий четкой организации и дисциплины. Он должен начинаться с определения ролей и распределения ответственности среди должностных лиц, занимающихся информационной безопасностью. Поэтому первый пункт аудиторского обследования начинается с получения информации об организационной структуре пользователей КС и обслуживающих подразделений. В связи с этим аудитору требуется следующая документация:

1. Схема организационной структуры пользователей;
2. Схема организационной структуры обслуживающих подразделений.

Обычно, в ходе интервью аудитор задает опрашиваемым следующие вопросы:

1. Кто является владельцем информации?
2. Кто является пользователем (потребителем) информации?
3. Кто является провайдером услуг?

Назначение и принципы функционирования КС во многом определяют существующие риски и требования безопасности, предъявляемые к системе. Поэтому на следующем этапе аудитора интересует информация о назначении и функционировании КС. Аудитор задает опрашиваемым примерно следующие вопросы:

1. Какие услуги и каким образом предоставляются конечным пользователям?
2. Какие основные виды приложений функционируют в КС?
3. Количество и виды пользователей, использующих эти приложения?

Ему понадобится также следующая документация, конечно, если таковая вообще имеется в наличии (что, вообще говоря, случается не слишком часто):

1. Функциональные схемы;
2. Описание автоматизированных функций;
3. Описание основных технических решений;
4. Другая проектная и рабочая документация на информационную систему.

Далее, аудитору требуется более детальная информация о структуре КС. Это позволит уяснить, каким образом осуществляется распреде-

ление механизмов безопасности по структурным элементам и уровням функционирования КС. Типовые вопросы, которые обсуждаются в связи с этим во время интервью, включают в себя:

1. Из каких компонентов (подсистем) состоит КС?
2. Функциональность отдельных компонент?
3. Где проходят границы системы?
4. Какие точки входа имеются?
5. Как КС взаимодействует с другими системами?
6. Какие каналы связи используются для взаимодействия с другими КС?
7. Какие каналы связи используются для взаимодействия между компонентами системы?
8. По каким протоколам осуществляется взаимодействие?
9. Какие программно-технические платформы используются при построении системы?

На этом этапе аудитору необходимо запастись следующей документацией:

1. Структурная схема КС;
2. Схема информационных потоков;
3. Описание структуры комплекса технических средств информационной системы;
4. Описание структуры программного обеспечения;
5. Описание структуры информационного обеспечения;
6. Размещение компонентов информационной системы.

Подготовка значительной части документации на КС, обычно, осуществляется уже в процессе проведения аудита. Когда все необходимые данные по КС, включая документацию, подготовлены, можно переходить к их анализу.

Используемые аудиторами методы анализа данных определяются выбранными подходами к проведению аудита, которые могут существенно различаться.

Первый подход, самый сложный, базируется на анализе рисков. Опираясь на методы анализа рисков, аудитор определяет для обследуемой КС индивидуальный набор требований безопасности, в наибольшей степени учитывающий особенности данной КС, среды ее функционирования и существующие в данной среде угрозы безопасности. Данный подход является наиболее трудоемким и требует наивысшей квалификации аудитора. На качество результатов аудита, в этом случае, сильно влияет используемая методология анализа и управления рисками и ее применимость к данному типу КС.

Второй подход, самый практичный, опирается на использование стандартов или нормативов по информационной безопасности. Стандарты определяют базовый набор требований безопасности для широкого класса КС, который формируется в результате обобщения мировой практики. Стандарты могут определять разные наборы требований безопасности, в зависимости от уровня защищенности КС, который требуется обеспечить, ее принадлежности (коммерческая организация, либо

государственное учреждение), а также назначения (финансы, промышленность, связь и т.п.). От аудитора в данном случае требуется правильно определить набор требований стандарта, соответствие которым требуется обеспечить для данной КС.

Третий подход, наиболее эффективный, предполагает комбинирование первых двух. Базовый набор требований безопасности, предъявляемых к КС, определяется стандартом. Дополнительные требования, в максимальной степени учитывающие особенности функционирования данной КС, формируются на основе анализа рисков. Этот подход является намного проще первого, т.к. большая часть требований безопасности уже определена стандартом, и, в то же время, он лишен недостатка второго подхода, заключающегося в том, что требования стандарта могут не учитывать специфики обследуемой КС.

Анализ рисков – это то, с чего должно начинаться построение любой системы информационной безопасности. Он включает в себя мероприятия по обследованию безопасности КС, с целью определения того какие ресурсы и от каких угроз надо защищать, а также в какой степени те или иные ресурсы нуждаются в защите. Определение набора адекватных контрмер осуществляется в ходе управления рисками. Риск определяется вероятностью причинения ущерба и величиной ущерба, наносимого ресурсам КС, в случае осуществления угрозы безопасности.

Анализ рисков состоит в том, чтобы выявить существующие риски и оценить их величину (дать им качественную, либо количественную оценку). Процесс анализа рисков можно разделить на несколько последовательных этапов:

• *Идентификация ключевых ресурсов КС;*

• *Определение важности тех или иных ресурсов для организации;*

• *Идентификация существующих угроз безопасности и уязвимостей, делающих возможным осуществление угроз;*

• *Вычисление рисков, связанных с осуществлением угроз безопасности.*

Ресурсы КС можно разделить на следующие категории:

• *Информационные ресурсы;*

• *Программное обеспечение;*

• *Технические средства (серверы, рабочие станции, активное сетевое оборудование и т. п.)*

• *Людские ресурсы*

В каждой категории ресурсы делятся на классы и подклассы. Необходимо идентифицировать только те ресурсы, которые определяют функциональность КС и существенны с точки зрения обеспечения безопасности.

Важность (или стоимость) ресурса определяется величиной ущерба, наносимого в случае нарушения конфиденциальности, целостности или доступности этого ресурса. Обычно рассматриваются следующие виды ущерба:

• *Данные были раскрыты, изменены, удалены или стали недоступны;*

• *Аппаратура была повреждена или разрушена;*

• *Нарушена целостность программного обеспечения.*

Ущерб может быть нанесен организации в результате успешного осуществления следующих видов угроз безопасности:

• *локальные и удаленные атаки на ресурсы КС;*

• *стихийные бедствия;*

• *ошибки, либо умышленные действия персонала КС;*

• *сбои в работе КС, вызванные ошибками в программном обеспечении или неисправностями аппаратуры.*

Под уязвимостями обычно понимают свойства КС, делающие возможным успешное осуществление угроз безопасности.

Величина риска определяется на основе стоимости ресурса, вероятности осуществления угрозы и величины уязвимости по следующей формуле:

$$Риск = \frac{(стоимость\ ресурса * вероятность\ угрозы)}{величина\ уязвимости}$$

Задача управления рисками заключается в выборе обоснованного набора контрмер, позволяющих снизить уровни рисков до приемлемой величины. Стоимость реализации контрмер должна быть меньше величины возможного ущерба. Разница между стоимостью реализации контрмер и величиной возможного ущерба должна быть обратно пропорциональна вероятности причинения ущерба.

Если для проведения аудита безопасности выбран подход, базирующийся на анализе рисков, то на этапе анализа данных аудита обычно выполняются следующие группы задач:

1. Анализ ресурсов КС, включая информационные ресурсы, программные и технические средства, а также людские ресурсы.

2. Анализ групп задач, решаемых системой, и бизнес процессов.

3. Построение (неформальной) модели ресурсов КС, определяющей взаимосвязи между информационными, программными, техническими и людскими ресурсами, их взаимное расположение и способы взаимодействия.

4. Оценка критичности информационных ресурсов, а также программных и технических средств.

5. Определение критичности ресурсов с учетом их взаимозависимостей.

6. Определение наиболее вероятных угроз безопасности в отношении ресурсов КС и уязвимостей защиты, делающих возможным осуществление этих угроз.

7. Оценка вероятности осуществления угроз, величины уязвимостей и ущерба, наносимого организации в случае успешного осуществления угроз.

8. Определение величины рисков для каждой тройки: угроза – группа ресурсов – уязвимость.

Перечисленный набор задач, является достаточно общим. Для их решения могут использоваться различные формальные и неформаль-

ные, количественные и качественные, ручные и автоматизированные методики анализа рисков. Суть подхода от этого не меняется.

Оценка рисков может даваться с использованием различных как качественных, так и количественных шкал. Главное, чтобы существующие риски были правильно идентифицированы и проранжированы в соответствии со степенью их критичности для организации. На основе такого анализа может быть разработана система первоочередных мероприятий по уменьшению величины рисков до приемлемого уровня.

При проведении аудита безопасности на соответствие требованиям стандарта, аудитор, полагаясь на свой опыт, оценивает применимость требований стандарта к обследуемой КС и ее соответствие этим требованиям. Данные о соответствии различных областей функционирования КС требованиям стандарта, обычно, представляются в табличной форме. Из таблицы видно, какие требования безопасности в системе не реализованы. Исходя из этого, делаются выводы о соответствии обследуемой КС требованиям стандарта и даются рекомендации по реализации в системе механизмов безопасности, позволяющих обеспечить такое соответствие.

Рекомендации, выдаваемые аудитором по результатам анализа состояния КС, определяются используемым подходом, особенностями обследуемой КС, состоянием дел с информационной безопасностью и степенью детализации, используемой при проведении аудита.

В любом случае, рекомендации аудитора должны быть конкретными и применимыми к данной КС, экономически обоснованными, аргументированными (подкрепленными результатами анализа) и отсортированными по степени важности. При этом мероприятия по обеспечению защиты организационного уровня практически всегда имеют приоритет над конкретными программно-техническими методами защиты.

В то же время, наивно ожидать от аудитора, в качестве результата проведения аудита, выдачи технического проекта подсистемы информационной безопасности, либо детальных рекомендаций по внедрению конкретных программно технических средств защиты информации. Это требует более детальной проработки конкретных вопросов организации защиты, хотя, внутренние аудиторы могут принимать в этих работах самое активное участие.

Аудиторский отчет является основным результатом проведения аудита. Его качество характеризует качество работы аудитора. Структура отчета может существенно различаться в зависимости от характера и целей проводимого аудита. Однако определенные разделы должны обязательно присутствовать в аудиторском отчете. Он должен, по крайней мере, содержать описание целей проведения аудита, характеристику обследуемой КС, указание границ проведения аудита и используемых методов, результаты анализа данных аудита, выводы, обобщающие эти результаты и содержащие оценку уровня защищенности АС или соответствие ее требованиям стандартов, и, конечно, рекомендации аудитора по устранению существующих недостатков и совершенствованию системы защиты.

Для примера, приведем образец структуры аудиторского отчета по результатам анализа рисков, связанных с осуществлением угроз безопасности в отношении обследуемой КС.

Структура отчета по результатам аудита безопасности КС и анализу рисков

Вводная часть

Введение

Цели и задачи проведения аудита

Описание КС

- Назначение и основные функции системы

- Группы задач, решаемых в системе

- Классификация пользователей КС

- Организационная структура обслуживающего персонала КС

- Структура и состав комплекса программно-технических средств КС

- Виды информационных ресурсов, хранимых и обрабатываемых в системе

- Структура информационных потоков

- Характеристика каналов взаимодействия с другими системами и точек входа

Границы проведения аудита

- Компоненты и подсистемы КС, попадающие в границы проведения аудита

- Размещение комплекса программно-технических средств КС по площадкам (помещениям)

- Основные классы угроз безопасности, рассматриваемых в ходе проведения аудита

Методика проведения аудита

- Методика анализа рисков

- Исходные данные

- Этапность работ

Структура документа

Оценка критичности ресурсов КС

Критерии оценки величины возможного ущерба, связанного с осуществлением угроз безопасности

Оценка критичности информационных ресурсов

- Классификация информационных ресурсов

- Оценка критичности по группам информационных ресурсов

Оценка критичности технических средств

Оценка критичности программных средств

Модель ресурсов КС, описывающая распределение ресурсов по группам задач

Анализ рисков, связанных с осуществлением угроз безопасности в отношении ресурсов КС

Модель нарушителя информационной безопасности

- Модель внутреннего нарушителя

- Модель внешнего нарушителя

Модель угроз безопасности и уязвимостей информационных ресурсов

- Угрозы безопасности, направленные против информационных ресурсов

-- Угрозы несанкционированного доступа к информации при помощи программных средств

-- Угрозы, осуществляемые с использованием штатных технических средств

-- Угрозы, связанные с утечкой информации по техническим каналам

- Угрозы безопасности, направленные против программных средств

- Угрозы безопасности направленные против технических средств

Оценка серьезности угроз безопасности и величины уязвимостей

- Критерии оценки серьезности угроз безопасности и величины уязвимостей

- Оценка серьезности угроз

- Оценка величины уязвимостей

Оценка рисков для каждого класса угроз и группы ресурсов

Выводы по результатам обследования

Рекомендации

Рекомендуемые контрмеры организационного уровня

Рекомендуемые контрмеры программно-технического уровня

В настоящее время имеется большое разнообразие как методов анализа и управления рисками, так и реализующих их программных средств. Приведем примеры некоторых, по мнению автора, наиболее распространенных.

Метод CRAMM (the UK Goverment Risk Analysis and Managment Method) был разработан Службой Безопасности Великобритании (UK Security Service) по заданию Британского правительства и взят на вооружение в качестве государственного стандарта. Он используется, начиная с 1985 г. правительственными и коммерческими организациями Великобритании. За это время CRAMM приобрел популярность во всем мире. Фирма Insight Consulting Limited занимается разработкой и сопровождением одноименного программного продукта, реализующего метод CRAMM.

Метод CRAMM выбран нами для более детального рассмотрения и это не случайно. В настоящее время CRAMM – это довольно мощный и универсальный инструмент, позволяющий, помимо анализа рисков, решать также и ряд других аудиторских задач, включая:

• Проведение обследования КС и выпуск сопроводительной документации на всех этапах его проведения;

• Проведение аудита в соответствии с требованиями Британского правительства, а также стандарта BS 7799:1995 – Code of Practice for Information Security Management BS7799;

• Разработка политики безопасности и плана обеспечения непрерывности бизнеса.

В основе метода CRAMM лежит комплексный подход к оценке рисков, сочетая количественные и качественные методы анализа. Метод является универсальным и подходит как для больших, так и для мелких организаций, как правительственного, так и коммерческого сектора. Версии программного обеспечения CRAMM, ориентированные на разные типы организаций, отличаются друг от друга своими базами знаний (profiles). Для коммерческих организаций имеется Коммерческий профиль (Commercial Profile), для правительственных организаций – Правительственный профиль (Government profile). Правительственный вариант профиля, также позволяет проводить аудит на соответствие требованиям американского стандарта ITSEC («Оранжевая книга»).

Грамотное использование метода CRAMM позволяет получать очень хорошие результаты, наиболее важным из которых, пожалуй, является возможность **экономического обоснования расходов организации на обеспечение информационной безопасности и непрерывности бизнеса**. Экономически обоснованная стратегия управления рисками позволяет, в конечном итоге, экономить средства, избегая неоправданных расходов.

CRAMM предполагает разделение всей процедуры на три последовательных этапа.

Задачей первого этапа является ответ на вопрос: «Достаточно ли для защиты системы применения средств базового уровня, реализующих традиционные функции безопасности, или необходимо проведение более детального анализа?» На втором этапе производится идентификация рисков и оценивается их величина. На третьем этапе решается вопрос о выборе адекватных контрмер.

Методика CRAMM для каждого этапа определяет набор исходных данных, последовательность мероприятий, анкеты для проведения интервью, списки проверки и набор отчетных документов.

Если по результатам проведения первого этапа, установлено, что уровень критичности ресурсов является очень низким и существующие риски заведомо не превысят некоторого базового уровня, то к системе предъявляется минимальный набор требований безопасности. В этом случае большая часть мероприятий второго этапа не выполняется, а осуществляется переход к третьему этапу, на котором генерируется стандартный список контрмер для обеспечения соответствия базовому набору требований безопасности.

На втором этапе производится анализ угроз безопасности и уязвимостей. Исходные данные для оценки угроз и уязвимостей аудитор получает от уполномоченных представителей организации в ходе соответствующих интервью. Для проведения интервью используются специализированные опросники.

На третьем этапе решается задача управления рисками, состоящая в выборе адекватных контрмер.

Решение о внедрении в систему новых механизмов безопасности и модификация старых принимает руководство организации, учитывая связанные с этим расходы, их приемлемость и конечную выгоду

для бизнеса. Задачей аудитора является обоснование рекомендуемых контрмер для руководства организации.

В случае принятия решения о внедрении новых контрмер и модификации старых, на аудитора может быть возложена задача подготовки плана внедрения новых контрмер и оценки эффективности их использования. Решение этих задач выходит за рамки метода CRAMM.

Процесс анализа и управления рисками по методу CRAMM

Процедура аудита в методе CRAMM является формализованной. На каждом этапе генерируется довольно большое количество промежуточных и результирующих отчетов.

Так, на первом этапе создаются следующие виды отчетов:

• Модель ресурсов, содержащая описание ресурсов, попадающих в границы исследования, и взаимосвязей между ними;

• Оценка критичности ресурсов;

• Результирующий отчет по первому этапу анализа рисков, в котором суммируются результаты, полученные в ходе обследования.

• На втором этапе проведения обследования создаются следующие виды отчетов:

• Результаты оценки уровня угроз и уязвимостей;

• Результаты оценки величины рисков;

• Результирующий отчет по второму этапу анализа рисков.

По результатам третьего этапа обследования создаются следующие виды отчетов:

• Рекомендуемые контрмеры;

• Детальная спецификация безопасности;

• Оценка стоимости рекомендуемых контрмер;

• Список контрмер, отсортированный в соответствии с их приоритетами;

• Результирующий отчет по третьему этапу обследования;

• Политика безопасности, включающая в себя описание требований безопасности, стратегий и принципов защиты КС;

• Список мероприятий по обеспечению безопасности.

Грамотно применять метод CRAMM в состоянии только высококвалифицированный аудитор, прошедший обучение. Если организация не может себе позволить содержать в штате такого специалиста, тогда самым правильным решением будет приглашение аудиторской фирмы, располагающей штатом специалистов, имеющих практический опыт применения метода CRAMM.

Обобщая практический опыт использования метода CRAMM при проведении аудита безопасности, можно сделать следующие выводы, относительно сильных и слабых сторон этого метода.

К сильным сторонам метода CRAMM относится следующее:

• CRAMM является хорошо структурированным и широко опробованным методом анализа рисков, позволяющим получать реальные практические результаты;

• Программный инструментарий CRAMM может использоваться на всех стадиях проведения аудита безопасности КС;

• В основе программного продукта лежит достаточно объемная база знаний по контрмерам в области информационной безопасности, базирующаяся на рекомендациях стандарта BS 7799;

• Гибкость и универсальность метода CRAMM позволяет использовать его для аудита КС любого уровня сложности и назначения;

• CRAMM можно использовать в качестве инструмента для разработки плана непрерывности бизнеса и политик информационной безопасности организации;

• CRAMM может использоваться в качестве средства документирования механизмов безопасности КС.

К недостаткам метода CRAMM можно отнести следующее:

• Использование метода CRAMM требует специальной подготовки и высокой квалификации аудитора;

• CRAMM в гораздо большей степени подходит для аудита уже существующих КС, находящихся на стадии эксплуатации, нежели чем для КС, находящихся на стадии разработки;

• Аудит по методу CRAMM – процесс достаточно трудоемкий и может потребовать месяцев непрерывной работы аудитора;

• Программный инструментарий CRAMM генерирует большое количество бумажной документации, которая не всегда оказывается полезной на практике;

• CRAMM не позволяет создавать собственные шаблоны отчетов или модифицировать имеющиеся;

• Возможность внесения дополнений в базу знаний CRAMM не доступна пользователям, что вызывает определенные трудности при адаптации этого метода к потребностям конкретной организации.

Программное обеспечение RiskWatch, разрабатываемое американской компанией RiskWatch, Inc., является мощным средством анализа и управления рисками. В семейство RiskWatch входят программные продукты для проведения различных видов аудита безопасности. Оно включает в себя следующие средства аудита и анализа рисков:

• RiskWatch for Physical Security –для физических методов защиты КС;

• RiskWatch for Information Systems – для информационных рисков;

• HIPAA-WATCH for Healthcare Industry – для оценки соответствия требованиям стандарта HIPAA;

• RiskWatch RW17799 for ISO17799 – для оценки требованиям стандарта ISO17799.

В методе RiskWatch в качестве критериев для оценки и управления рисками используются «предсказание годовых потерь» (Annual Loss Expectancy – ALE) и оценка «возврата от инвестиций» (Return on Investment – ROI).

Система COBRA (Consultative Objective and Bi-Functional Risk Analysis), разрабатываемая компанией Risk Associates, является средством анализа рисков и оценки соответствия КС стандарту ISO17799. COBRA реализует методы количественной оценки рисков, а также инструменты для консалтинга и проведения обзоров безопасности. При

разработке инструментария COBRA были использованы принципы построения экспертных систем, обширная база знаний по угрозам и уязвимостям, а также большое количество вопросников, с успехом применяющихся на практике. В семейство программных продуктов COBRA входят COBRA ISO17799 Security Consultant, COBRA Policy Compliance Analyst и COBRA Data Protection Consultant.

Программный продукт Buddy System, разрабатываемый компанией Countermeasures Corporation, является еще одним программным продуктом, позволяющим осуществлять как количественный, так и качественный анализ рисков. Он содержит развитые средства генерации отчетов. Основной акцент при использовании Buddy System делается на информационные риски, связанные с нарушением физической безопасности и управление проектами.

Спецификация SysTrust в настоящее время достаточно широко используется аудиторскими компаниями, традиционно выполняющими финансовый аудит для своих клиентов и предлагающих услугу ИТ аудита в качестве дополнения к финансовому аудиту.

Немецкий стандарт «BSI\IT Baseline Protection Manual» содержит наиболее содержательное руководство по обеспечению безопасности ИТ и представляет несомненную практическую ценность для всех специалистов, занимающихся вопросами информационной безопасности.

Практические стандарты и руководства по обеспечению информационной безопасности, разрабатываемые в рамках проекта SCORE, ориентированны на технических специалистов и являются в техническом плане наиболее совершенными в настоящее время.

Программа сертификации Интернет сайтов по требованиям информационной безопасности и соответствующая спецификация «SANS/GIAC Site Certification», предложенная институтом SANS, заслуживает рассмотрения в связи с неизменно возрастающей актуальностью вопросов защиты КС организаций от атак со стороны сети Интернет и увеличением доли соответствующих работ при проведении аудита безопасности.

Теперь обратимся к краткому обзору стандартов информационной безопасности, являющихся наиболее значимыми и перспективными с точки зрения их использования для проведения аудита безопасности КС *более подробно стандарты будут рассмотрены ниже).

Результатом проведения аудита, в последнее время, все чаще становится сертификат (или заключение), удостоверяющих соответствие обследуемой КС требованиям стандарта. Наличие такого сертификата позволяет организации получать конкурентные преимущества, связанные с большим доверием со стороны клиентов и партнеров.

Стандарты ISO17799 и ISO15408 служат основой для проведения работ в области информационной безопасности, в том числе и аудита. ISO17799 сосредоточен на вопросах организации и управления безопасностью, в то время как ISO15408 определяет детальные требования, предъявляемые к программно-техническим механизмам защиты информации.

Наиболее полно критерии для оценки механизмов безопасности организационного уровня представлены в международном стандарте ISO 17799: Code of Practice for Information Security Management (Практические правила управления информационной безопасностью), принятом в 2000 году. ISO 17799 был разработан на основе британского стандарта BS 7799.

ISO 17799 может использоваться в качестве критериев для оценки механизмов безопасности организационного уровня, включая административные, процедурные и физические меры защиты.

Практические правила разбиты на следующие 10 разделов:

1. Политика безопасности
2. Организация защиты
3. Классификация ресурсов и их контроль
4. Безопасность персонала
5. Физическая безопасность
6. Администрирование компьютерных систем и вычислительных сетей
7. Управление доступом
8. Разработка и сопровождение информационных систем
9. Планирование бесперебойной работы организации
10. Контроль выполнения требований политики безопасности

Десять разделов контроля, предлагаемых в ISO 17799 (они обозначены как ключевые), считаются особенно важными. Под средствами контроля в данном контексте понимаются механизмы управления информационной безопасностью организации.

При использовании некоторых из средств контроля, например, шифрования данных, могут потребоваться советы специалистов по безопасности и оценка рисков, чтобы определить, нужны ли они и каким образом их следует реализовывать. Для обеспечения более высокого уровня защиты особенно ценных ресурсов или оказания противодействия особенно серьезным угрозам безопасности, в ряде случаев могут потребоваться более сильные средства контроля, которые выходят за рамки ISO 17799.

Десять ключевых средств контроля, перечисленные ниже, представляют собой либо обязательные требования, например, требования действующего законодательства, либо считаются основными структурными элементами информационной безопасности, например, обучение правилам безопасности. Эти средства контроля актуальны для всех организаций и сред функционирования КС и составляют основу системы управления информационной безопасностью.

Ключевыми являются следующие средства контроля:

- документ о политике информационной безопасности;
- распределение обязанностей по обеспечению информационной безопасности;
- обучение и подготовка персонала к поддержанию режима информационной безопасности;
- уведомление о случаях нарушения защиты;

• средства защиты от вирусов;
• планирование бесперебойной работы организации;
• контроль над копированием программного обеспечения, защищенного законом об авторском праве;
• защита документации организации;
• защита данных;
• контроль соответствия политике безопасности.

Процедура аудита безопасности КС включает в себя проверку наличия перечисленных ключевых средств контроля, оценку полноты и правильности их реализации, а также анализ их адекватности рискам, существующим в данной среде функционирования. Составной частью работ по аудиту безопасности КС также является анализ и управление рисками.

Наиболее полно критерии для оценки механизмов безопасности программно-технического уровня представлены в международном стандарте ISO 15408: Common Criteria for Information Technology Security Evaluation (Общие критерии оценки безопасности информационных технологий), принятом в 1999 году. Подробно мы их рассмотрим в последней части книги.

Общие критерии оценки безопасности информационных технологий (далее «Общие критерии») определяют функциональные требования безопасности (security functional requirements) и требования к адекватности реализации функций безопасности (security assurance requirements).

При проведении работ по анализу защищенности КС, «Общие критерии» целесообразно использовать в качестве основных критериев, позволяющих оценить уровень защищенности АС с точки зрения полноты реализованных в ней функций безопасности и надежности реализации этих функций.

Хотя применимость «Общих критериев» ограничивается механизмами безопасности программно-технического уровня, в них содержится определенный набор требований к механизмам безопасности организационного уровня и требований по физической защите, которые непосредственно связаны с описываемыми функциями безопасности.

Первая часть «Общих критериев» содержит определение общих понятий, концепции, описание модели и методики проведения оценки безопасности ИТ. В ней вводится понятийный аппарат, и определяются принципы формализации предметной области.

Требования к функциональности средств защиты приводятся во второй части «Общих критериев» и могут быть непосредственно использованы при анализе защищенности для оценки полноты реализованных в КС функций безопасности.

Третья часть «Общих критериев», наряду с другими требованиями к адекватности реализации функций безопасности, содержит класс требований по анализу уязвимостей средств и механизмов защиты под названием AVA: Vulnerability Assessment. Данный класс требований определяет методы, которые должны использоваться для предупреждения, выявления и ликвидации следующих типов уязвимостей:

• Наличие побочных каналов утечки информации;

• Ошибки в конфигурации, либо неправильное использование системы, приводящее к переходу системы в небезопасное состояние;

• Недостаточная надежность (стойкость) механизмов безопасности, реализующих соответствующие функции безопасности;

• Наличие уязвимостей в средствах защиты информации, позволяющих пользователям получать НСД к информации в обход существующих механизмов защиты.

При проведении работ по аудиту безопасности, данные требования могут использоваться в качестве руководства и критериев для анализа уязвимостей КС.

По существу, аудит в области информационных технологий, хотя и не имеет прямого отношения к финансовому аудиту, часто является дополнением к нему в качестве коммерческой услуги, предлагаемой аудиторскими фирмами своим клиентам, в связи с повышением зависимости бизнеса клиентов от ИТ. Идея заключается в том, что использование надежных и безопасных ИТ-систем до определенной степени гарантирует надежность финансовой отчетности организации. Хорошие результаты ИТ-аудита в некоторых случаях позволяют проводить финансовый аудит в сокращенном варианте, экономя время и деньги клиентов.

Отвечая потребностям бизнеса, Американским Институтом Сертифицированных Публичных Бухгалтеров (American Institute of Certified Public Accountants (AICPA)) и Канадским Институтом Общественных Бухгалтеров (Canadian Institute of Chartered Accountants (CICA)) разработали стандарт SysTrust для проведения ИТ аудита, который является дополнением к финансовому аудиту. SysTrust позволяет финансовым аудиторам расширить область своей деятельности, путем использования простого и понятного набора требований для оценки надежности и безопасности КС.

В стандарте SysTrust КС оценивается в терминах ее доступности (Availability), безопасности (Security), целостности (Integrity) и эксплуатационной надежности (Maintainability).

Под доступностью традиционно понимается возможность КС предоставлять информационные сервисы в любых режимах функционирования и при любых нагрузках, предусмотренных условиями ее эксплуатации, с задержками, не превышающими установленные требования.

Под безопасностью понимается защищенность КС от физического и логического несанкционированного доступа. В качестве средств обеспечения безопасности в основном рассматриваются средства разграничения физического и логического доступа к ресурсам КС.

Под целостностью понимается возможность КС обеспечить сохранение таких свойств обрабатываемой в системе информации как полнота, точность, актуальность, своевременность и аутентичность.

Эксплуатационная надежность КС определяется возможностью изменения конфигурации и обновления системы для обеспечения таких ее свойств как доступность, безопасность и целостность.

Критерии для оценки описанных четырех свойств КС определены в документе «AICPA/CICA SysTrust Principles and Criteria for Systems

Reliability, Version 2.0» (Принципы и критерии для оценки надежности систем).

В ходе сертификации по требованиям стандарта SysTrust (SysTrust engagement) аудитор оценивает соответствие КС критериям доступности, безопасности, целостности и эксплуатационной надежности (SysTrust Principles and Criteria), проверяя наличие в системе необходимых механизмов контроля. Затем аудитор производит тестирование механизмов контроля с целью определения их работоспособности и эффективности. Если в результате тестирования подтверждается соответствие КС критериям SysTrust, аудитор выпускает отчет по аттестации (unqualified attestation report). В отчете формулируются выводы относительно полноты и эффективности реализации руководством организации механизмов контроля в аттестуемой КС. В дополнение к отчету по аттестации, аудитор готовит общее описание обследуемой КС. Во многих случаях также готовится утверждение руководства организации (management's assertion) относительно эффективности механизмов контроля, позволяющих обеспечить соответствие КС критериям SysTrust. Обследование КС и оценка ее соответствия критериям SysTrust производится в соответствии с «Руководством по Проведению Аттестации» («Statement on Standards for Attestation Engagements (SSAE) No. 10, Attestation Standards, AT sec. 101»Attest Engagements»».)

Примером проведения аудита крупнейшей компании на соответствие требованиям стандарта SysTrust может служить сертификация аудиторской фирмой Ernst&Young системы BeeOffice, принадлежащей компании АО «Вымпелком».)

Немецкий стандарт «Руководство по обеспечению безопасности ИТ базового уровня» (IT Baseline Protection Manual) разрабатывается Агенством Информационной Безопасности Германии (BSI – Bundesamt fur Sicherheit in der Informationstechnik (German Information Security Agency).

Этот документ является, пожалуй, самым содержательным руководством по информационной безопасности и по многим параметрам превосходит все остальные стандарты. Приятен также тот факт, что этот ценнейший для аудитора источник информации имеется в свободном доступе в сети Интернет. В нем содержатся подробные руководства по обеспечению информационной безопасности применительно к различным аспектам функционирования КС и различным областям ИТ.

«BSI\IT Baseline Protection Manual» постоянно совершенствуется с целью обеспечения его соответствия текущему состоянию дел в области безопасности ИТ. К настоящему времени накоплена уникальная база знаний, содержащая информацию по угрозам и контрмерам в хорошо структурированном виде.

Практические стандарты SCORE и программа сертификации SANS/GIAC Site Certification

SCORE (Security Consensus Operational Readiness Evaluation) является совместным проектом института SANS и Центра безопасности Интернет (Center for Internet Security(CIS)).

Профессионалы-практики в области информационной безопасности из различных организаций объединились в рамках проекта SCORE с целью разработки базового (минимально необходимого) набора практических стандартов и руководств по обеспечению безопасности для различных операционных платформ. Требования и рекомендации, предлагаемые для включения в стандарты, широко обсуждаются и проверяются участниками проекта SCORE, и только после их одобрения всеми участниками, передаются в CIS, который занимается их формализацией и оформлением, а также разрабатывает программные средства (minimum standards benchmarks) для оценки соответствия операционных платформ предложенным стандартам.

Разработанные базовые стандарты вместе с руководствами по обеспечению соответствия этим стандартам и средствами тестирования публикуются на Интернет сайте CIS.

Программа сертификации Интернет сайтов (GIAC Site Certification program), предложенная институтом SANS, позволяет организациям проводить аудит безопасности сегментов компьютерной сети, непосредственно подключенных к сети Интернет, в соответствии со стандартами SCORE.

Программа сертификации «GIAC Site Certification» определяет три уровня защищенности Интернет сайтов. На практике, в настоящее время, используются только первые два из них.

Сертификация сайта на первом уровне предполагает проверку внешних сетевых адресов организации, видимых из сети Интернет, на предмет уязвимости соответствующих хостов в отношении сетевых атак. На этом уровне должна быть обеспечена защита сайта от наиболее распространенных атак. Требуется отсутствие наиболее серьезных и часто встречающихся уязвимостей защиты. Предъявляются также определенные требования к уровню квалификации специалистов, отвечающих за обеспечение безопасности сайта.

На втором уровне требуется проведение всех проверок и соблюдение всех требований первого уровня, а, кроме того, требуется осуществлять периодический пересмотр политики и процедур обеспечения сетевой безопасности. Также на втором уровне производится проверка защищенности сайта от сетевых атак путем осуществления попыток проникновения и взлома систем, подключенных к сети Интернет.

На третьем уровне, помимо обеспечения соответствия всем требованиям второго уровня, требуется также регулярно проводить сканирование сети изнутри с целью защиты от угроз со стороны внутренних нарушителей, а также внешних злоумышленников, пытающихся преодолеть механизмы защиты внешнего периметра сети путем использования продвинутых методов, включая методы социального инженеринга.

От уровня к уровню ужесточаются требования, предъявляемые к квалификации специалистов, организационной структуре подразделений, занимающихся вопросами защиты, наличию формальных политик и процедур, а также строгости и глубине тестов, используемых для проверки механизмов защиты.

В заключение приведем типовой вопросник для проведения аудита:

1. Для чего предназначена КС организации? Если это конфиденциальная информация, ответ – в общих терминах (например, для проектирования, для перевода платежей, для хранения информации и т.д.).

2. Сколько рабочих мест содержит система, сколько обслуживает пользователей (клиентов), сколько в ней обслуживающего персонала?

3. Как система структурируется? (например, сколько она содержит самостоятельных подразделений (подобъектов, площадок), как они связаны друг с другом, сколько рабочих мест на каждом объекте?)

4. Создана система самостоятельно или интегратором (поставщиком)? Сколько времени она создавалась? Есть ли проектно-сметная документация? Возможно ли предоставить выдержки из нее?

5. Существует ли регламентная документация на КС общего характера (описание КС, описание ее назначения, описание ее функционирования, требования к ней)? Возможно ли предоставить выдержки из нее?

6. Каков состав аппаратной платформы КС? Каков состав аппаратных средств ЛВС и телекоммуникаций? Используются ли арендованные каналы связи? Используется ли оборудование в аутсорсинге, расположенное вне рамок КС? Каким образом обеспечено электропитание и климат-контроль?

7. Каков состав операционных платформ КС (какие применяются операционные системы)? Каким образом они закуплены (приобретены)?

8. Укажите перечень продуктов бизнес-логики (программ, предназначенных для выполнения основных функций КС)? Каким образом они закуплены (приобретены)? Есть ли среди них продукты собственной разработки?

9. Какие используются средства обеспечения информационной безопасности (средства разграничения доступа, антивирусные средств, средства криптографической защиты, межсетевые экраны и т.д.)? Каким образом они закуплены (приобретены)? Есть ли среди них продукты собственной разработки?

10. Есть ли в КС в целом или ее структурных подразделениях служба управления (IT-управления)? Каковы ее функции? Имеются ли регламенты ее работы? Можно ли предоставить выдержки из них?

11. Есть ли в КС в целом или ее структурных подразделениях служба обеспечения общей безопасности? Каковы ее функции? Имеются ли регламенты ее работы? Можно ли предоставить выдержки из них?

12. Есть ли в КС в целом или ее структурных подразделениях служба обеспечения информационной безопасности? Каковы ее функции? Имеются ли регламенты ее работы? Можно ли предоставить выдержки из них?

13. Имеются ли в организации, КС в целом или ее структурных подразделениях служба внутреннего контроля или аудита? Каковы ее функции? Имеются ли регламенты ее работы? Можно ли предоставить выдержки из них?

14. Используется ли при обслуживании системы аутстаффинг (привлекаются ли сторонние фирмы к проведению работ по обслуживанию, ремонту, модернизации)? В каких объемах?

15. Существует ли в организации представление о злоумышленнике для КС и угрозах, которые могут быть им реализованы? Каковы меры обеспечения надежности системы от повреждений и стихийных бедствий?

8.8. Эксплуатация защищенных систем и понятие системы обеспечения информационной безопасности (СОИБ)

Защищенную КС с подключением внешних пользователей, можно разделить на два связанных сегмента: внешний, относящийся к абонентам (клиентам) системы, взаимодействующих с внутренним, служебным (обрабатывающим) контуром. Принципиальным различием обоих контуров является вид действующего в них потенциального злоумышленника (нарушителя).

Во внешнем контуре действует **внешний нарушитель**, взаимодействующий с телекоммуникационной средой и серверами доступа, во внутреннем – пользователи и администраторы обрабатывающего контура, управляющие прикладными подсистемами и подсистемами безопасности данного контура, а также и правилами подключения внешних абонентов, среди которых возможны **внутренние нарушители**. Кроме того, потенциально возможен **сговор** внутреннего и внешнего нарушителей (злоумышленников).

Как мы рассматривали выше, механизмы безопасности в КС должны быть реализованы в виде подсистемы реализации ПБ и подсистемы гарантирования ПБ, а также в виде регистрации событий, происходящих в КС или в указанных подсистемах.

Таким образом, в КС возникают отдельные подсистемы информационной безопасности (ПИБ), решающие отдельные задачи. Каждая из ПИБ как некоторая законченная подсистема КС в целом характеризуется следующими принципиально важными свойствами:

1. Наличием Администратора (Администраторов) либо службы администрирования ПИБ, генерирующим управляющие воздействия на данную ПИБ и авторизуемых для легитимного управления конкретной ПИБ.

2. Информацией журналов ПИБ или КС, характеризующей результаты работы конкретной ПИБ и предотвращенные (или пропущенные) ей инциденты (нарушения).

При этом можно констатировать, что для решения задачи интегрального управления всеми ПИБ, входящими в КС необходимо обеспечить создание и функционирование **некоторой надстройки над всеми ПИБ**, которую мы назовем Системой обеспечения информационной безопасности (СОИБ), осуществляющую общие для всех ПИБ функции управления и аудита и решающую следующий набор задач:

1. Задача **интегрирования ПИБ**, состоящая в том, что для всех или некоторого подмножества ПИБ формируются управляющие и мониторинговые воздействия из единого центра, которым является СОИБ.

2. Задача **аудита ПИБ**, состоящая в необходимости обработки журналов всех или выделенного подмножества элементов ПИБ, рассматриваемых как единый массив данных в единых координатах времени КЦОИ, предназначенный для выявления корреляций событий различных ПИБ.

3. Задача **оптимизации управления ПИБ**, состоящая в снижении временных и стоимостных затрат на управление и аудит и решаемая, как известно из теории управления, консолидацией управляющих и аудирующих воздействий в центре, которым в данном случае является СОИБ.

4. Задача **поддержания актуальности состояния ИБ**, которая может быть реализована только как синергетическое сочетание актуальных состояний защищенности отдельных ПИБ и также реализуемая из единого центра управления.

5. Задача **проверки соответствия уровня защищенности** системы КЦОИ в целом уровням стандартов и требований, которая решается путем анализа защищенности отдельных ПИБ, путем сбора и интегральной оценки информации от них.

6. Задача **защиты от непреднамеренных ошибок и злоумышленных действий администраторов различных ПИБ**, решаемая путем интегрального аудита в СОИБ и созданием иерархии администраторов также в рамках СОИБ.

7. Задача **динамической коррекции политики безопасности** системы КЦОИ в целом и реакции на инциденты, решаемой в рамках единого центра управления передачей корректирующей ПБ информации различным ПИБ.

Наличие СОИБ является обязательным условием управляемости больших защищенных систем, либо систем гетерогенного характера, объединяющих разнородные аппаратные средства и операционные среды.

9. НОРМАТИВНЫЕ ДОКУМЕНТЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

Введение

Данная глава посвящена обзору основных положений нормативных документов, регламентирующих проектирование, разработку и сертификацию защищенных КС. Основные положения и обзор документов, изложенных в данной части, приводятся (в значительной части – цитируются или переводятся с минимальным редактированием) по [1] и первоисточникам [2-25], содержащим оригинальные тексты (см. литературу к данной части).

Рассмотрены руководящие документы Гостехкомиссии РФ, документы Министерства обороны США («Оранжевая книга»), Европейские критерии безопасности и Федеральные критерии США.

Необходимо заметить, что, несмотря на то, что ряд документов содержит в названии или в основном тексте понятие «критерий», рассматривать их как необходимые и достаточные условия для создания защищенных КС не представляется возможным. В основном в документах описаны **необходимые** условия защищенности в смысле указания основных механизмов и процедур обеспечения безопасности, а также требований к их разработке и сопровождению. Целесообразно рассматривать данные документы как совокупность необходимых достаточно общих требований, а также как основу классификации защищенных КС.

Наибольший интерес представляют «Федеральные критерии» как пример стандарта нового поколения, в котором совершен переход от функционального стандарта (описывающего предметную область защиты) к инвариантному объектно-ориентированному профилю, характерному для документов, регламентирующих жизненный цикл открытых систем.

9.1. Документы Государственной технической комиссии России

9.1.1. Введение

В 1992 году Гостехкомиссия (ГТК) при Президенте Российской Федерации разработала и опубликовала пять руководящих документов, посвященных вопросам защиты информации в автоматизированных системах ее обработки [2,3,4,5,6].

Основой этих документов является концепция защиты средств вычислительной техники от несанкционированного доступа к информации, содержащая систему взглядов ГТК на проблему информационной безопасности и основные принципы защиты КС. С точки зрения разработчиков данных документов основная задача средств безопасности – это обеспечение защиты от несанкционированного доступа к информации.

Определенный уклон в сторону поддержания секретности информации объясняется тем, что эти документы были разработаны в расчете на применение в информационных системах силовых структур РФ.

9.1.2. Структура требований безопасности

Руководящие документы ГТК предлагают две группы требований к безопасности – показатели защищенности средств вычислительной техники (СВТ) от НСД и критерии защищенности автоматизированных систем (АС) обработки данных. Первая группа позволяет оценить степень защищенности отдельно поставляемых потребителю компонентов КС, а вторая рассчитана на более сложные комплексы, включающие несколько единиц СВТ.

9.1.3. Показатели защищенности средств вычислительной техники от несанкционированного доступа

Данный руководящий документ устанавливает классификацию СВТ по уровню защищенности от НСД к информации на базе перечня показателей защищенности и совокупности описывающих их требований. Под СВТ понимается совокупность программных и технических элементов систем обработки данных, способных функционировать самостоятельно или в составе других систем.

Данные показатели содержат требования защищенности СВТ от НСД к информации и применяются к общесистемным программным средствам и операционным системам (с учетом архитектуры ЭВМ). Конкретные перечни показателей определяют классы защищенности СВТ и описываются совокупностью требований. Совокупность всех средств защиты составляет комплекс средств защиты (КСЗ).

Установлено семь классов защищенности СВТ от НСД к информации. Самый низкий класс – седьмой, самый высокий – первый.

Показатели защищенности и установленные требования к классам приведены в таблице.

Таблица 1. Требования к защите от НСД СВТ

Наименование показателя	Класс защищенности					
	6	5	4	3	2	1
Дискреционный принцип контроля доступа	+	+	+	=	+	=
Мандатный принцип контроля доступа	-	-	+	=	=	=
Очистка памяти	-	+	+	+	=	=
Изоляция модулей	-	-	+	=	+	=
Маркировка документов	-	-	+	=	=	=
Защита ввода и вывода на отчужденный физический носитель информации	-	-	+	=	=	=
Сопоставление пользователя с устройством	-	-	+	=	=	=
Идентификация и аутентификация	+	=	+	=	=	=
Гарантии проектирования	-	+	+	+	+	+
Регистрация	-	+	+	+	=	=
Взаимодействие пользователя с КСЗ	-	-	-	+	=	=

Надежное восстановление	-	-	-	+	=	=
Целостность КСЗ	-	+	+	+	=	=
Контроль модификации	-	-	-	-	+	=
Контроль дистрибуции	-	-	-	-	+	=
Гарантии архитектуры	-	-	-	-	-	+
Тестирование	+	+	+	+	+	=
Руководство пользователя	+	=	=	=	=	=
Руководство по КСЗ	+	+	=	+	+	=
Текстовая документация	+	+	+	+	+	=
Конструкторская (проектная) документация	+	+	+	+	+	+

Обозначения: «-» – нет требований к данному классу, «+» – новые или дополнительные требования, «=» – требования совпадают с требованиями к СВТ предыдущего класса.

Важно отметить, что требования являются классическим примером применения необходимых условий оценки качества защиты, т.е. если какой-либо механизм присутствует, то это является основанием для отнесения СВТ к некоторому классу.

Интересно, что защищенные СВТ содержат разделение только по двум классам политик безопасности: дискреционной и мандатной (см. часть 1).

Невлияние субъектов друг на друга описывается требованием «изоляция модулей» (требуется с 4 класса). Гарантии выполнения ПБ коррелированны с требованием «целостность КСЗ» (требуется с 5 класса) и «гарантии проектирования» (также требуется с 5 класса).

9.1.4. Классы защищенности автоматизированных систем

Документы ГТК устанавливают девять классов защищенности АС от НСД, каждый из которых характеризуется определенной совокупностью требований к средствам защиты. Классы подразделяются на три группы, отличающиеся спецификой обработки информации в АС. Группа АС определяется на основании следующих признаков:

- наличие в АС информации различного уровня конфиденциальности;
- уровень полномочий пользователей АС на доступ к конфиденциальной информации;
- режим обработки данных в АС (коллективный или индивидуальный).

В пределах каждой группы соблюдается иерархия классов защищенности АС. Класс, соответствующий высшей степени защищенности для данной группы, обозначается индексом *N***А**, где *N* – номер группы (от 1 до 3). Следующий класс обозначается *N***Б** и т.д.

Третья группа включает АС, в которых работает один пользователь, допущенный ко всей информации АС, размещенной на носителях одного уровня конфиденциальности. Группа содержит два класса – **3Б** и **3А**.

Вторая группа включает АС, в которых пользователи имеют одинаковые полномочия доступа ко всей информации, обрабатываемой и хранимой в АС на носителях различного уровня конфиденциальности. Группа содержит два класса – **2Б** и **2А**.

Первая группа включает многопользовательские АС, в которых одновременно обрабатывается и хранится информация разных уровней конфиденциальности. Не все пользователи имеют равные права доступа. Группа содержит пять классов – **1Д**, **1Г**, **1В**, **1Б** и **1А**.

В таблице приведены требования к подсистемам защиты для каждого класса.

Таблица 2. Требования к защите от НСД АС

Подсистемы и требования	Классы								
	3Б	3А	2Б	2А	1Д	1Г	1В	1Б	1А
1. Подсистема управления доступом 1.1. Идентификация. Проверка подлинности и контроль доступа субъектов в систему,	+	+	+	+	+	+	+	+	+
к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ,				+		+	+	+	+
к программам,				+		+	+	+	+
к томам, каталогам, файлам, записям, полям записей.				+		+	+	+	+
1.2. Управление потоками информации.				+			+	+	+
2. Подсистема регистрации и учета 2.1.Регистрация и учет: входа/выхода субъектов доступа в/из системы (узла сети),	+	+	+	+	+	+	+	+	+
выдачи печатных (графических) выходных документов,		+		+		+	+	+	+
запуска/завершения программ и процессов (заданий, задач),				+		+	+	+	+
доступа программ субъектов к защищаемым файлам, включая их создание и удаление, передачу по линиям и каналам связи,				+		+	+	+	+
доступа программ субъектов, доступа к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ, программам, каталогам, файлам, записям, полям записей,				+		+	+	+	+
изменения полномочий субъектов доступа,							+	+	+
создаваемых защищаемых объектов доступа.				+			+	+	+
2.2. Учет носителей информации	+	+	+	+	+	+	+	+	+
2.3. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей		+		+		+	+	+	+
2.4. Сигнализация попыток нарушения защиты							+	+	+

3. Криптографическая подсистема 3.1. Шифрование конфиденциальной информации				+				+	+
3.2. Шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах									+
3.3. Использование аттестованных (сертифицированных) криптографических средств				+				+	+
4. Подсистема обеспечения целостности 4.1. Обеспечение целостности программных средств и обрабатываемой информации	+	+	+	+	+	+	+	+	+
4.2. Физическая охрана средств вычислительной техники и носителей информации	+	+	+	+	+	+	+	+	+
4.3. Наличие администратора (службы) защиты информации в АС				+			+	+	+
4.4. Периодическое тестирование СЗИ НСД	+	+	+	+	+	+	+	+	+
4.5. Наличие средств восстановления СЗИ НСД	+	+	+	+	+	+	+	+	+
4.6. Использование сертифицированных средств защиты		+		+			+	+	+

Обозначения:

«+» – требование к данному классу присутствует, иначе – данное требование необязательно.

9.1.5. Выводы

Рассмотренные документы необходимо воспринимать как первую стадию формирования отечественных стандартов в области информационной безопасности.

На разработку этих документов наибольшее влияние оказала «Оранжевая книга» (рассматривается ниже), однако это влияние в основном отражается в ориентированности обоих документов на защищенные системы силовых структур и в использовании единой универсальной шкалы оценки степени защищенности.

К недостаткам данного стандарта относятся: ориентация на противодействие НСД и отсутствие требований к адекватности реализации политики безопасности. Понятие «политика безопасности» трактуется исключительно как поддержание режима секретности и отсутствие НСД [5]. Из-за этого средства защиты ориентируются исключительно на противодействие внешним угрозам, а к структуре самой системы и ее функционированию не предъявляется четких требований. Ранжирование требований по классам защищенности по сравнению с остальными стандартами информационной безопасности максимально упрощено и сведено до определения наличия или отсутствия заданного набора механизмов защиты, что существенно снижает гибкость требований и возможность их практического применения.

Несмотря на указанные недостатки документы ГТК заполнили «правовой вакуум» в области стандартов информационной безопасности в России и оперативно решили проблему проектирования и оценки качества защищенных КС.

9.2. Критерии безопасности компьютерных систем Министерства обороны США («Оранжевая книга»)

9.2.1. Цель разработки

«Критерии безопасности компьютерных систем» (Trusted Computer System Evaluation Criteria) [7], получившие неформальное, но прочно закрепившееся название «Оранжевая книга», были разработаны и опубликованы Министерством обороны США в 1983 году с целью определения требований безопасности, предъявляемых к аппаратному, программному и специальному программному и информационному обеспечению КС и выработки методологии и технологии анализа степени поддержки политики безопасности в компьютерных системах в основном военного назначения.

В данном документе были впервые формально (хотя и не вполне строго) определены такие понятия, как «политика безопасности», «корректность» и другие. Согласно «Оранжевой книге» безопасная КС – это система, поддерживающая управление доступом к обрабатываемой в ней информации, таким образом, что только соответствующим образом авторизованные пользователи или процессы (субъекты), действующие от их имени, получают возможность читать, записывать, создавать и удалять информацию. Предложенные в этом документе концепции защиты и набор функциональных требований послужили основой для формирования всех появившихся впоследствии стандартов безопасности.

9.2.2. Общая структура требований «Оранжевой книги»

В «Оранжевой книге» предложены три категории требований безопасности – политика безопасности, аудит и корректность, в рамках которых сформулированы шесть базовых требований безопасности. Первые четыре требования направлены непосредственно на обеспечение безопасности информации, а два последних – на качество самих средств защиты. Рассмотрим эти требования подробнее.

Политика безопасности

Требование 1. *Политика безопасности.* Система должна поддерживать точно определенную политику безопасности. Возможность осуществления субъектами доступа к объектам должна определяться на основании их идентификации и набора правил управления доступом. Там, где это необходимо, должна использоваться политика мандатного управления доступом, позволяющая эффективно реализовать разграничение доступа к информации различного уровня конфиденциальности.

Требование 2. *Метки.* С объектами должны быть ассоциированы метки безопасности (см. также аксиому 2 части 1), используемые в качестве исходной информации для процедур контроля доступа. Для реализации мандатного управления доступом система должна обеспечивать возможность присваивать каждому объекту метку или набор атри-

бутов, определяющих степень конфиденциальности (гриф секретности) объекта и режимы доступа к этому объекту.

Аудит

Требование 3. *Идентификация и аутентификация.* Все субъекты должны иметь уникальные идентификаторы. Контроль доступа должен осуществляться на основании результатов идентификации субъекта и объекта доступа, подтверждения подлинности их идентификаторов (аутентификации) и правил разграничения доступа. Данные, используемые для идентификации и аутентификации, должны быть защищены от несанкционированного доступа, модификации и уничтожения и должны быть ассоциированы со всеми активными компонентами компьютерной системы, функционирование которых критично с точки зрения безопасности.

Требование 4. *Регистрация и учет.* Для определения степени ответственности пользователей за действия в системе, все происходящие в ней события, имеющие значение с точки зрения безопасности, должны отслеживаться и регистрироваться в защищенном протоколе (т.е. должен существовать объект КС, потоки от которого и к которому доступны только субъекту администрирования). Система регистрации должна осуществлять анализ общего потока событий и выделять из него только те события, которые оказывают влияние на безопасность для сокращения объема протокола и повышения эффективности его анализа. Протокол событий должен быть надежно защищен от несанкционированного доступа, модификации и уничтожения.

Корректность

Требование 5. *Контроль корректности функционирования средств защиты.* Средства защиты должны содержать независимые аппаратные и/или программные компоненты, обеспечивающие работоспособность функций защиты. Это означает, что все средства защиты, обеспечивающие политику безопасности, управление атрибутами и метками безопасности, идентификацию и аутентификацию, регистрацию и учет, должны находиться под контролем средств, проверяющих корректность их функционирования. Основной принцип контроля корректности состоит в том, что средства контроля должны быть полностью независимы от средств защиты.

Требование 6. *Непрерывность защиты.* Все средства защиты (в т.ч. и реализующие данное требование) должны быть защищены от несанкционированного вмешательства и/или отключения, причем эта защита должна быть постоянной и непрерывной в любом режиме функционирования системы защиты и КС в целом. Данное требование распространяется на весь жизненный цикл компьютерной системы. Кроме того, его выполнение является одним из ключевых аксиом, используемых для формального доказательства безопасности системы.

9.2.3. Классы безопасности компьютерных систем

«Оранжевая книга» предусматривает четыре группы критериев, которые соответствуют различной степени защищенности: от минимальной (группа **D**) до формально доказанной (группа **A**). Каждая группа включает один или несколько классов. Группы **D** и **A** содержат по од-

ному классу (классы **D** и **A** соответственно), группа **C** – классы **C1**, **C2**, а группа **B** три класса – **B1**, **B2**, **B3**, характеризующиеся различными наборами требований безопасности. Уровень безопасности возрастает при движении от группы **D** к группе **A**, а внутри группы – с возрастанием номера класса.

Таблица 3. Требования «Оранжевой книги»

	Наименование	Класс защищенности					
		C1	C2	B1	B2	B3	A1
Security Policy							
1	Discretionary Access Control	+	+	+	=	=	=
2	Mandatory Access Control	-	-	+	+	=	=
3	Labels	-	-	+	+	=	=
4	Labels integrity	-	-	+	=	=	=
5	Working Label	-	-	-	+	=	=
6	Labels Frequency	-	-	+	=	=	=
7	Object Reuse	-	+	=	+	=	=
8	Resource Encapsulation	-	+	=	=	=	=
9	Export Machine Readable Output	-	-	+	=	=	=
10	Export Human-Readable Labels	-	-	+	=	=	=
Accountability							
11	Identification & Authentication	+	+	=	=	=	=
12	Audit	-	+	+	+	+	=
13	Trusted Path	-	-	-	+	=	=
Assurance							
14	Design Specification and verification	-	-	+	+	+	+
15	System Architecture	+	=	=	+	+	=
16	System Integrity	+	=	=	=	=	=
17	Security Testing	+	+	+	+	+	=
18	Trusted Recovery	-	-	-	-	+	=
19	Configuration Management	-	-	-	+	+	+
20	Trusted Facility Management	-	-	-	+	+	=
21	Trusted Distribution	-	-	-	-	+	=
22	Covert Channel Analysis	-	-	-	+	+	+
Documentation							
23	Users Guide	+	=	=	=	=	=
24	Facility Manual	+	+	+	+	+	=
25	Test Documentation	+	=	=	=	=	+
26	Design Documentation	+	=	+	+	=	+

Группа D. Минимальная защита

Класс **D**. Минимальная защита. К этому классу относятся все системы, которые не удовлетворяют требованиям других классов.

Группа С. Дискреционная защита

Группа **С** характеризуется наличием произвольного управления доступом и регистрацией действий субъектов.

Класс **С1**. Дискреционная защита. Системы этого класса удовлетворяют требованиям обеспечения разделения пользователей и информации и включают средства контроля и управления доступом, позволяющие задавать ограничения для индивидуальных пользователей, что дает им возможность защищать свою приватную информацию от других пользователей. Класс **С1** рассчитан на многопользовательские системы, в которых осуществляется совместная обработка данных одного уровня конфиденциальности.

Класс **С2**. Управление доступом. Системы этого класса осуществляют более гибкое управление доступом, чем системы класса **С1**, с помощью применения средств индивидуального контроля за действиями пользователей, регистрацией, учетом событий и выделением ресурсов.

Группа В. Мандатная защита

Основные требования этой группы – нормативное управление доступом с использованием меток безопасности, поддержка модели и политики безопасности, а также наличие спецификаций на функции доверенной вычислительной базы (см. ниже). Для систем этой группы монитор взаимодействий (МБО) должен контролировать все события (потоки) в системе.

Класс **В1**. Защита с применением меток безопасности. Системы класса **В1** должны соответствовать всем требованиям, предъявляемым к системам класса **С2,** и, кроме того, должны поддерживать определенную неформально модель безопасности, маркировку данных и нормативное управление доступом. При экспорте из системы информация должна подвергаться маркировке. Обнаруженные в процессе тестирования недостатки должны быть устранены.

Класс **В2**. Структурированная защита. Для соответствия классу **В2** доверенная вычислительная база КС должна поддерживать формально определенную и четко документированную модель безопасности, предусматривающую произвольное и нормативное управление доступом, которое распространяется по сравнению с системами класса **В1** на все субъекты. Кроме того, должен осуществляться контроль скрытых каналов утечки информации. В структуре ядра защиты должны быть выделены элементы, критичные с точки зрения безопасности. Интерфейс доверенной вычислительной базы должен быть четко определен, а ее архитектура и реализация должны быть выполнены с учетом возможности проведения тестовых испытаний. По сравнению с классом **В1** должны быть усилены средства аутентификации. Управление безопасностью осуществляется администраторами системы. Должны быть предусмотрены средства управления конфигурацией.

Класс **В3**. Домены безопасности. Для соответствия этому классу ядро защиты системы должно включать монитор взаимодействий, который контролирует все типы доступа субъектов к объектам, который невозможно обойти (т.е. требуется гарантирование заданной ПБ). Кроме того, ядро защиты должно быть структурировано с целью исключения из нее подсистем, не отвечающих за реализацию функций защиты, и быть достаточно компактным для эффективного тестирования и анализа. В

ходе разработки и реализации ядра защиты должны применяться методы и средства, направленные на минимизацию его сложности. Средства аудита должны включать механизмы оповещения администратора при возникновении событий, имеющих значение для безопасности системы. Требуется наличие средств восстановления работоспособности системы.

Группа А. Верифицированная защита

Данная группа характеризуется применением формальных методов верификации корректности работы механизмов управления доступом (произвольного и нормативного). Требуется дополнительная документация, демонстрирующая, что архитектура и реализация ядра защиты отвечают требованиям безопасности.

Класс **А1**. Формальная верификация. Системы класса **А1** функционально эквивалентны системам класса **В3,** и к ним не предъявляется никаких дополнительных функциональных требований. В отличие от систем класса **В3** в ходе разработки должны применяться формальные методы верификации, что позволяет с высокой уверенностью получить корректную реализацию функций защиты. Процесс доказательства адекватности реализации начинается на ранней стадии проектирования с построения формальной модели политики безопасности. Для обеспечения эффективности применения методов верификации системы класса **А1** должны содержать более мощные средства управления конфигурацией и защищенную процедуру дистрибуции (установки и распространения).

Приведенные классы безопасности надолго определили основные концепции безопасности и ход развития средств защиты.

9.2.4. Интерпретация и развитие «Оранжевой книги»

Опубликование «Оранжевой книги» стало важным этапом как в постановке, так и в указании направления решения основных теоретических проблем компьютерной безопасности. Тем не менее, в ходе применения ее основных положений выяснилось, что часть практически важных вопросов осталась за рамками данного стандарта, и, кроме того, с течением времени (с момента опубликования прошло пятнадцать лет) ряд положений устарел и потребовал пересмотра.

Круг специфических вопросов по обеспечению безопасности компьютерных сетей и систем управления базами данных нашел отражение в отдельных документах, изданных Национальным центром компьютерной безопасности США в виде дополнений к «Оранжевой книге» – «Интерпретация «Оранжевой книги» для компьютерных сетей» (Trusted Network Interpretation [8]) и «Интерпретация «Оранжевой книги» для систем управления базами данных» (Trusted Database Management System Interpretation [9]). Эти документы содержат трактовку основных положений «Оранжевой книги» применительно к соответствующим классам систем обработки информации.

Устаревание ряда положений «Оранжевой книги» обусловлено, прежде всего, интенсивным развитием компьютерных технологий и переходом с вычислительных комплексов типа IBM-360/370, (советский аналог – машины серии ЕС) к рабочим станциям, высокопроизводительным персональным компьютерам и сетевой модели вычислений. Именно

для того, чтобы исключить возникшую в связи с изменением аппаратной платформы некорректность некоторых положений «Оранжевой книги», адаптировать их к современным условиям и сделать адекватными нуждам разработчиков и пользователей программного обеспечения, и была проделана значительная работа по интерпретации и развитию положений этого стандарта. В результате возник целый ряд сопутствующих «Оранжевой книге» документов, многие их которых стали ее неотъемлемой частью. К наиболее часто упоминаемым относятся:

Руководство по произвольному управлению доступом в безопасных системах (A guide to understanding discretionary access control in trusted systems [10]).

Руководство по управлению паролями (Password management guideline[11]).

Руководство по применению «Критериев безопасности компьютерных систем» в специфических средах (Guidance for applying the Department of Defence Trusted Computer System Evaluation Criteria in specific environment[12]).

Руководство по аудиту в безопасных системах (A Guide to Understanding Audit in Trusted Systems) [13].

Руководство по управлению конфигурацией в безопасных системах (Guide to understanding configuration management in trusted systems [14]).

Количество подобных вспомогательных документов, комментариев и интерпретаций значительно превысило объем первоначального документа, и в 1995 году Национальным центром компьютерной безопасности США был опубликован документ под названием «Интерпретация критериев безопасности компьютерных систем»[15], объединяющий все дополнения и разъяснения. При его подготовке состав подлежащих рассмотрению и толкованию вопросов обсуждался на специальных конференциях разработчиков и пользователей защищенных систем обработки информации. В результате открытого обсуждения была создана база данных, включающая все спорные вопросы, которые затем в полном объеме были проработаны специально созданной рабочей группой. В итоге появился документ, проинтегрировавший все изменения и дополнения к «Оранжевой книге», сделанные с момента ее опубликования, что привело к обновлению стандарта и позволило применять его в современных условиях.

9.2.5. Выводы

«Критерии безопасности компьютерных систем» Министерства обороны США представляют собой первую попытку создать единый стандарт безопасности, рассчитанный на проектировщиков, разработчиков, потребителей и специалистов по сертификации систем безопасности компьютерных систем. В свое время этот документ явился значительным шагом в области безопасности информационных технологий и послужил отправной точкой для многочисленных исследований и разработок. Основной отличительной чертой этого документа, как уже отмечалось, является его ориентация на системы военного применения, причем в основном на операционные системы. Это предопределило доминирование требований, направленных на обеспечение конфиденциальности обрабатываемой информации и исключение возможностей ее

разглашения. Большое внимание уделено меткам конфиденциальности (грифам секретности) и правилам экспорта секретной информации.

Требования по гарантированию политики безопасности отражены достаточно поверхностно, соответствующий раздел по существу ограничивается требованиями контроля целостности средств защиты и поддержания их работоспособности, чего явно недостаточно (см. часть 1).

«Оранжевая книга» послужила основой для разработчиков всех остальных стандартов информационной безопасности и до сих пор используется в США в качестве руководящего документа при сертификации компьютерных систем обработки информации.

9.3. Европейские критерии безопасности информационных технологий

Вслед за выходом «Оранжевой книги» страны Европы разработали согласованные «Критерии безопасности информационных технологий» (Information Technology Security Evaluation Criteria, далее «Европейские критерии»). Данный обзор основывается на версии 1.2 данного документа, опубликованной в июне 1991 года от имени соответствующих органов четырех стран: Франции, Германии, Нидерландов и Великобритании [16].

9.3.1. Основные понятия

«Европейские Критерии» рассматривают следующие задачи средств информационной безопасности:

• защита информации от несанкционированного доступа с целью обеспечения конфиденциальности (поддержание конфиденциальности);

• обеспечение целостности информации посредством защиты от ее несанкционированной модификации или уничтожения (поддержание целостности);

• обеспечение работоспособности систем с помощью противодействия угрозам отказа в обслуживании (поддержание доступности).

Для того чтобы удовлетворить требованиям конфиденциальности, целостности и доступности, необходимо реализовать соответствующий набор функций безопасности, таких как идентификация и аутентификация, управление доступом, восстановление после сбоев и т. д.. Чтобы средства защиты можно было признать эффективными, требуется высокая степень уверенности в правильности их выбора и надежности функционирования. Для решения этой проблемы в «Европейских критериях» впервые вводится понятие адекватности (assurance) средств защиты.

Адекватность включает в себя два аспекта: эффективность, отражающую соответствие средств безопасности решаемым задачам, и корректность, характеризующую процесс их разработки и функционирования. Эффективность определяется соответствием между задачами, поставленными перед средствами безопасности, и реализованным набором функций защиты – их функциональной полнотой и согласованностью, простотой использования, а также возможными последствиями использования злоумышленниками слабых мест защиты. Под корректностью понимается правильность и надежность реализации функций безопасности (в принятой терминологии – гарантирование избранной ПБ).

Общая оценка уровня безопасности системы складывается из функциональной мощности средств защиты и уровня адекватности их реализации.

9.3.2. Функциональные критерии

В «Европейских критериях» средства, имеющие отношение к информационной безопасности, рассматриваются на трех уровнях детализации. На первом уровне рассматриваются цели, которые преследует обеспечение безопасности, второй уровень содержит спецификации функций защиты, а третий – реализующие их механизмы. Спецификации функций защиты предлагается рассматривать с точки зрения следующих требований:

- идентификация и аутентификация;
- управление доступом;
- подотчетность;
- аудит;
- повторное использование объектов;
- целостность информации;
- надежность обслуживания;
- безопасность обмена данными.

Большинство из перечисленных требований совпадает с аналогичными требованиями «Оранжевой книги». Остановимся лишь на специфичных для «Европейских критериев» моментах.

Требования безопасности обмена данными регламентируют работу средств, обеспечивающих безопасность данных, передаваемых по каналам связи, и включают следующие разделы:

- аутентификация;
- управление доступом;
- конфиденциальность данных;
- целостность данных;
- невозможность отказаться от совершенных действий.

Набор функций безопасности может специфицироваться с использованием ссылок на заранее определенные классы-шаблоны. В «Европейских критериях» таких классов десять. Пять из них (**F-C1**, **F-C2**, **F-B1**, **F-B2**, **F-B3**) соответствуют классам безопасности «Оранжевой книги» с аналогичными обозначениями. Рассмотрим подробнее другие пять классов, т. к. их требования отражают точку зрения разработчиков стандарта на проблему безопасности.

Класс **F-IN** предназначен для систем с высокими потребностями в обеспечении целостности, что типично для систем управления базами данных. Его описание основано на концепции «ролей», соответствующих видам деятельности пользователей, и предоставлении доступа к определенным объектам только посредством доверенных процессов. Должны различаться следующие виды доступа: чтение, запись, добавление, удаление, создание, переименование и выполнение объектов (имеется в виду порождение субъекта из соответствующего объекта-источника).

Класс **F-AV** характеризуется повышенными требованиями к обеспечению работоспособности. Это существенно, например, для систем управления технологическими процессами. В требованиях этого класса

указывается, что система должна восстанавливаться после отказа отдельной аппаратной компоненты таким образом, чтобы все критически важные функции постоянно оставались доступными. В таком же режиме должна происходить и замена компонентов системы. Независимо от уровня загрузки должно гарантироваться определенное максимальное время реакции системы на внешние события.

Класс **F-DI** ориентирован на распределенные системы обработки информации. Перед началом обмена и при получении данных стороны должны иметь возможность провести идентификацию участников взаимодействия и проверить ее подлинность. Должны использоваться средства контроля и исправления ошибок. В частности, при пересылке данных должны обнаруживаться все случайные или намеренные искажения адресной и пользовательской информации. Знание алгоритма обнаружения искажений не должно позволять злоумышленнику производить нелегальную модификацию передаваемых данных. Должны обнаруживаться попытки повторной передачи ранее переданных сообщений.

Класс **F-DC** уделяет особое внимание требованиям к конфиденциальности передаваемой информации. Информация по каналам связи должна передаваться только в зашифрованном виде. Ключи шифрования должны быть защищены от несанкционированного доступа.

Класс **F-DX** предъявляет повышенные требования и к целостности и к конфиденциальности информации. Его можно рассматривать как функциональное объединение классов **F-DI** и **F-DC** с дополнительными возможностями шифрования и защиты от анализа трафика. Должен быть ограничен доступ к ранее переданной информации.

9.3.3. Критерии адекватности

«Европейские критерии» уделяют адекватности средств защиты значительно больше внимания, чем функциональным требованиям. Адекватность складывается из двух компонент – эффективности и корректности работы средств защиты.

«Европейские критерии» определяют семь уровней адекватности – от **E0** до **E6** (в порядке возрастания). Уровень **E0** обозначает минимальную адекватность. При проверке адекватности анализируется весь жизненный цикл системы – от начальной фазы проектирования до эксплуатации и управления. Уровни адекватности от **E1** до **E6** выстроены по нарастанию требований тщательности контроля. Так, на уровне **E1** анализируется лишь общая архитектура системы, а адекватность средств защиты подтверждается функциональным тестированием. На уровне **E3** к анализу привлекаются исходные тексты программ и схемы аппаратного обеспечения. На уровне **E6** требуется формальное описание функций безопасности, общей архитектуры, а также политики безопасности.

Степень безопасности системы определяется самым слабым из критически важных механизмов защиты. В «Европейских критериях» определены три уровня безопасности – базовый, средний и высокий. Безопасность считается базовой, если средства защиты способны противостоять отдельным случайным атакам (злоумышленник – физическое лицо). Безопасность считается средней, если средства защиты способ-

ны противостоять злоумышленникам, обладающим ограниченными ресурсами и возможностями (корпоративный злоумышленник). Наконец, безопасность можно считать высокой, если есть уверенность, что средства защиты могут быть преодолены только злоумышленником с высокой квалификацией, набор возможностей и ресурсов которого выходит за рамки возможного (злоумышленник – государственная спецслужба).

9.3.4. Выводы

«Европейские критерии безопасности информационных технологий» появившиеся, вслед за «Оранжевой книгой» оказали существенное влияние на стандарты безопасности и методику сертификации.

Главное достижение этого документа – введение понятия адекватности средств защиты и определение отдельной шкалы для критериев адекватности.

Необходимо отметить, что «Европейские критерии» тесно связаны с «Оранжевой книгой», что делает их не вполне самостоятельным документом.

На первый взгляд довольно странным представляется факт, что «Европейские критерии» признают возможность наличия недостатков в сертифицированных на некоторый класс защищенных системах (критерий возможности использования недостатков защиты), однако на самом деле данный подход свидетельствует о реалистичном взгляде на существующее положение дел.

9.4. Федеральные критерии безопасности информационных технологий

9.4.1. Цель разработки

«Федеральные критерии безопасности информационных технологий» (Federal Criteria for Information Technology Security) разрабатывались как одна из составляющих «Американского федерального стандарта по обработке информации» (Federal Information Processing Standard), призванного заменить »Оранжевую книгу». Разработчиками стандарта выступили Национальный институт стандартов и технологий США (National Institute of Standards and Technology) и Агентство национальной безопасности США (National Security Agency). Данный обзор основан на версии 1.0 этого документа, опубликованной в декабре 1992 года [17].

Этот документ разработан на основе результатов многочисленных исследований в области обеспечения безопасности информационных технологий 80-х – начала 90-х годов, а также на основе анализа опыта использования «Оранжевой книги». Документ представляет собой основу для разработки и сертификации компонентов информационных технологий с точки зрения обеспечения безопасности.

9.4.2. Основные положения

«Федеральные критерии безопасности информационных технологий» (далее «Федеральные критерии») охватывают практически полный спектр проблем, связанных с защитой и обеспечением безопасности, т.к. включают все аспекты обеспечения конфиденциальности, целостности и доступности.

Основными объектами применения требований безопасности «Федеральных критериев» являются продукты информационных технологий (Information Technology Products) и системы обработки информации (Information Technology Systems). Под продуктом информационных технологий (далее – ПИТ) понимается совокупность аппаратных и программных средств, которая представляет собой поставляемое конечному потребителю готовое к использованию средство обработки информации. Как правило, ПИТ эксплуатируется не автономно, а интегрируется в систему обработки информации, представляющую собой совокупность ПИТ, объединенных в функционально полный комплекс, предназначенный для решения прикладных задач (т.е. для реализации некоторой целевой функции КС). В ряде случаев система обработки информации может состоять только из одного ПИТ, обеспечивающего решение всех стоящих перед системой задач и удовлетворяющего требованиям безопасности. С точки зрения безопасности принципиальное различие между ПИТ и системой обработки информации определяется средой их эксплуатации. Продукт информационных технологий обычно разрабатывается в расчете на то, что он будет использован во многих системах обработки информации, и, следовательно, разработчик должен ориентироваться только на самые общие предположения о среде эксплуатации своего продукта, включающие условия применения и общие угрозы. Напротив, система обработки информации разрабатывается для решения прикладных задач в расчете на требования конечных потребителей, что позволяет в полной мере учитывать специфику воздействий со стороны конкретной среды эксплуатации.

Положения «Федеральных критериев» касаются только собственных средств обеспечения безопасности ПИТ, т.е. механизмов защиты, встроенных непосредственно в эти продукты в виде соответствующих программных, аппаратных или специальных средств. Для повышения их эффективности могут дополнительно применяться внешние системы защиты и средства обеспечения безопасности, к которым относятся как технические средства, так и организационные меры, правовые и юридические нормы. В конечном счете, безопасность ПИТ определяется совокупностью собственных средств обеспечения безопасности и внешних средств, являющихся частью КС.

Ключевым понятием концепции информационной безопасности «Федеральных критериев» является понятие «Профиля защиты» (Protection Profile). Профиль защиты – это нормативный документ, который регламентирует все аспекты безопасности ПИТ в виде требований к его проектированию, технологии разработки и сертификации. Как правило, один Профиль защиты описывает несколько близких по структуре и назначению ПИТ. Основное внимание в Профиле защиты уделяется требованиям к составу средств защиты и качеству их реализации, а также их адекватности предполагаемым угрозам безопасности.

«Федеральные критерии» представляют процесс разработки систем обработки информации, начинающийся с формулирования требований потребителями и заканчивающийся введением в эксплуатацию, в виде следующих основных этапов:

1. Разработка и анализ Профиля защиты. Требования, изложенные в Профиле защиты, определяют функциональные возможности ПИТ по

обеспечению безопасности и условия эксплуатации, при соблюдении которых гарантируется соответствие предъявляемым требованиям. Кроме требований безопасности Профиль содержит требования по соблюдению технологической дисциплины в процессе разработки, тестирования, анализа и сертификации ПИТ. Профиль безопасности анализируется на полноту, непротиворечивость и техническую корректность.

2. Разработка и сертификация ПИТ. Разработанные ПИТ подвергаются независимому анализу, целью которого является определение степени соответствия характеристик продукта сформулированным в Профиле защиты требованиям и спецификациям.

3. Компоновка и сертификация системы обработки информации в целом. Успешно прошедшие второй этап ПИТ интегрируются в систему обработки информации. Полученная в результате система должна удовлетворять заявленным в Профиле защиты требованиям при соблюдении указанных в нем условий эксплуатации.

«Федеральные критерии» регламентируют только первый этап этой схемы – разработку и анализ Профиля защиты, процесс создания ПИТ и компоновка систем обработки информации остаются вне рамок этого стандарта.

9.4.3. Профиль защиты

Как уже говорилось, Профиль защиты является центральным понятием «Федеральных критериев», большая часть содержания которых представляет собой описание разделов Профиля защиты, включающее набор требований безопасности и их ранжирование. Рассмотрим назначение, структуру и этапы разработки Профиля защиты.

Назначение и структура Профиля защиты

Профиль защиты предназначен для определения и обоснования состава и содержания средств защиты, спецификации технологии разработки и регламентации процесса сертификации ПИТ. Профиль защиты состоит из следующих пяти разделов:

- описание;
- обоснование;
- функциональные требования к ПИТ;
- требования к технологии разработки ПИТ;
- требования к процессу сертификации ПИТ.

Описание Профиля содержит классификационную информацию, необходимую для его идентификации в специальной картотеке. «Федеральные критерии» предлагают поддерживать такую картотеку на общегосударственном уровне. Это позволит любой организации воспользоваться созданными ранее Профилями защиты непосредственно, или использовать их в качестве прототипов для разработки новых. В описании Профиля защиты должна быть охарактеризована основная проблема или группа проблем обеспечения безопасности, решаемых с помощью применения данного Профиля.

Обоснование содержит описание среды эксплуатации, предполагаемых угроз безопасности и методов использования ПИТ. Кроме того, этот раздел содержит подробный перечень задач по обеспечению безопасности, решаемых с помощью данного Профиля. Эта информация

дает возможность определить, в какой мере данный Профиль защиты пригоден для применения в той или иной ситуации. Предполагается, что данный раздел ориентирован на службы безопасности организаций, которые изучают возможность использования ПИТ, соответствующего данному Профилю защиты.

Раздел *функциональных требований к ПИТ* содержит описание функциональных возможностей средств защиты ПИТ и определяет условия, в которых обеспечивается безопасность в виде перечня угроз, которым успешно противостоят предложенные средства защиты. Угрозы, лежащие вне этого диапазона, должны быть устранены с помощью дополнительных, не входящих в состав продукта, средств обеспечения безопасности. Очевидно, что чем сильнее и опаснее угрозы, тем более мощными и стойкими должны быть средства, реализующие функции защиты.

Раздел *требований к технологии разработки ПИТ* охватывает все этапы его создания, начиная от разработки проекта и заканчивая вводом готовой системы в эксплуатацию. Раздел содержит требования как к самому процессу разработки, так и к условиям, в которых она проводится, к используемым технологическим средствам, а также к документированию этого процесса. Выполнение требований этого раздела является непременным условием для проведения сертификации ПИТ.

Раздел *требований к процессу сертификации ПИТ* регламентирует порядок сертификации в виде типовой методики тестирования и анализа. Объем и глубина требуемых исследований зависят от наиболее вероятных типов угроз, среды применения и планируемой технологии эксплуатации.

«Федеральные критерии» содержат подробное описание всех трех разделов Профиля защиты, включающее набор требований и их ранжирование для каждого раздела. В данном обзоре основное внимание уделено функциональным требованиям, т.к. именно в этой области «Федеральные критерии» проделали значительный шаг вперед по сравнению с предшествующими стандартами.

Этапы разработки Профиля защиты

Разработка Профиля защиты осуществляется в три этапа: анализ среды применения ПИТ с точки зрения безопасности, выбор Профиля-прототипа и синтез требований.

На первой стадии проводится анализ информации о среде предполагаемого применения ПИТ, действующих в этой среде угрозах безопасности и используемых этими угрозами недостатках защиты. Анализ проводится с учетом технологии использования продукта, а также существующих стандартов и нормативов, регламентирующих его эксплуатацию.

Второй этап состоит в поиске Профиля, который может быть использован в качестве прототипа. Как уже говорилось, «Федеральные критерии» предусматривают создание специальной, доступной для разработчиков ПИТ, картотеки, в которую должны быть помещены все разработанные когда-либо Профили защиты. Это позволит минимизировать затраты на создание Профилей и учесть опыт предыдущих разработок.

Этап синтеза требований включает выбор наиболее существенных для условий функционирования продукта функций защиты и их ранжирование по степени важности с точки зрения обеспечения качества за-

щиты. Выбор специфичных для среды требований безопасности должен быть основан на их эффективности для решения задачи противодействия угрозам. Разработчик Профиля должен показать, что выполнение выдвинутых требований ведет к обеспечению требуемого уровня безопасности посредством успешного противостояния заданному множеству угроз и устранения недостатков защиты.

При разработке Профиля защиты необходимо анализировать связи и взаимозависимости, существующие между функциональными требованиями и требованиями к процессу разработки, а также между отдельными требованиями внутри этих разделов. По завершению разработки Профиль защиты подвергается проверке, целью которой является подтверждение его полноты, корректности, непротиворечивости и реализуемости.

9.4.4. Функциональные требования к продукту информационных технологий

«Федеральные критерии» предлагают набор функциональных требований, реализация которых позволяет противостоять наиболее распространенным угрозам безопасности, относящимся к широкому спектру ПИТ, и областей их применения. Данные требования разработаны с учетом возможности расширения и адаптации к конкретным условиям эксплуатации ПИТ, и допускают возможность совершенствования параллельно процессу развития информационных технологий. Требования, изложенные в «Федеральных критериях», разработаны на основе обобщения существовавших на момент их создания стандартов информационной безопасности – «Оранжевой книги» и «Европейских критериев» (см. выше).

Функциональные требования, приведенные в «Федеральных критериях», определяют состав и функциональные возможности доверенной вычислительной базы (Trusted Computer Base, TCB) или в принятой терминологии, ядра защиты (ЯЗ). Ядро защиты объединяет все компоненты ПИТ (аппаратные, программные и специальные средства), реализующие функции защиты. Таким образом, функциональные требования, направленные на обеспечение безопасности, относятся либо к внутренним элементам ЯЗ, либо к ее внешним функциям, доступным через специальные интерфейсы.

Для того чтобы расширить спектр потенциального применения Профиля защиты в «Федеральных критериях», при описании функциональных требований предполагается, что ЯЗ является единственной частью ПИТ, которая нуждается в защите и обладает такой характеристикой, как *уровень защищенности*. По этой причине предполагается достаточным установить множество требований, касающихся только безопасности ЯЗ.

Функциональные требования Профиля защиты задаются в виде общих положений и косвенным образом определяют множество угроз, которым может успешно противостоять удовлетворяющий им ПИТ.

Структура функциональных требований

Функциональные требования «Федеральных критериев» разделены на восемь классов и определяют все аспекты функционирования ЯЗ. Реализация политики безопасности должна быть поддержана средствами, обеспечивающими надежность функционирования как самой ЯЗ, так

и механизмов осуществления политики безопасности. Эти средства также входят в состав ЯЗ, хотя, с точки зрения противодействия угрозам, вносят только косвенный вклад в общую защиту ПИТ.

Поскольку «Федеральные критерии» по сравнению с «Оранжевой книгой» являются стандартом нового поколения, и, кроме того, никогда не рассматривались в отечественных публикациях, остановимся на функциональных требованиях более подробно.

Требования к реализации политики безопасности описывают функции ЯЗ, реализующие политику безопасности, и состоят из четырех групп: требования к политике аудита, политике управления доступом, политике обеспечения работоспособности и управлению безопасностью. Эти требования носят весьма общий характер, что позволяет рассматривать их в качестве прототипа, обеспечивающего поддержку широкого спектра политик и моделей безопасности.

Политика аудита включает разделы, относящиеся к идентификации и аутентификации, процедуре регистрации пользователя в системе, обеспечению прямого взаимодействия с ЯЗ, а также регистрацию и учет событий. Основная задача политики управления аудитом – обеспечить возможность однозначной идентификации субъекта, ответственного за те или иные действия в системе.

Идентификация и аутентификация позволяют установить однозначное соответствие между пользователями и представляющими их в КС субъектами (т.е. субъектами, инициированными от имени конкретного пользователя), а также подтвердить подлинность этого соответствия.

Регистрация пользователя в системе означает создание субъекта взаимодействия, с идентификатором которого будут ассоциироваться все последующие действия пользователя. К процедуре регистрации также относится учет места, времени и других параметров подключения к системе и ее блокирование во время отсутствия пользователя.

Обеспечение прямого взаимодействия с ЯЗ гарантирует, что пользователь взаимодействует с компонентами ЯЗ напрямую, т.е. информация, которая передается в ЯЗ и обратно, не подвергается перехвату или искажению (см. часть 2). Поддержка прямого взаимодействия с ЯЗ особенно важна для управления безопасностью (например, при администрировании прав доступа и полномочий пользователей).

Регистрация и учет событий в системе позволяет распознавать потенциально опасные ситуации и сигнализировать о случаях нарушения безопасности. Регистрация событий включает распознавание, учет и анализ действий пользователя, представляющих интерес с точки зрения безопасности.

Политика управления доступом содержит следующие разделы: произвольное управление доступом (дискреционное), нормативное управление доступом и контроль скрытых каналов утечки информации. Политика управления доступом является основным механизмом защиты, т.к. непосредственно обеспечивает конфиденциальность и целостность обрабатываемой информации.

Произвольное управление доступом позволяет осуществлять назначение прав доступа с точностью до идентифицируемых субъектов и объектов, а также поддерживаемых типов доступа и, кроме того, обеспечивает контроль за распространением прав доступа среди субъектов.

Нормативное управление доступом, в отличие от произвольного, основано на контроле информационных потоков между субъектами и объектами и их атрибутах безопасности, что позволяет регламентировать порядок использования информации в системе.

Контроль скрытых каналов утечки информации включает технические и административные меры, направленные на ликвидацию таких каналов посредством минимизации объема совместно используемых ресурсов и введения активных «шумовых помех».

Политика обеспечения работоспособности системы включает контроль за распределением ресурсов и обеспечение отказоустойчивости. Обеспечение работоспособности позволяет гарантировать доступность ресурсов и сервиса системы, а также корректное восстановление системы после сбоев.

Контроль за распределением ресурсов осуществляется посредством введения ограничений (квот) на их потребление или приоритетной системы распределения ресурсов.

Обеспечение отказоустойчивости входит в сферу безопасности наравне с другими требованиями, т. к. противостоит угрозам работоспособности.

Управление безопасностью регламентирует следующие аспекты функционирования системы:

- компоновка, установка, конфигурация и поддержка ЯЗ;
- администрирование атрибутов безопасности пользователей (идентификаторов, полномочий, доступных ресурсов и т.д.);
- администрирование политики управления доступом;
- управление потреблением ресурсов системы;
- аудит действий пользователей.

Мониторинг взаимодействий. Требования этого раздела регламентируют порядок взаимодействия между компонентами системы и прохождения информационных потоков через ЯЗ. Реализация политики безопасности будет эффективна только в том случае, если все без исключения взаимодействия в системе, т. е. доступ к объектам, ресурсам и сервису осуществляются при обязательном посредничестве ТСВ (следовательно, требуется, чтобы ядро защиты представляло собой МБО – см. часть 1).

Логическая защита ЯЗ. Требования данной группы устанавливают порядок доступа к внутренним компонентам ЯЗ (данным и программам). ЯЗ должно быть защищено от внешних воздействий со стороны непривилегированных пользователей, в противном случае искажение программ и данных, находящихся в ЯЗ, может привести к полному подавлению функций защиты (данное требование включает требование корректности внешних субъектов относительно субъектов ядра защиты и требование на интерфейсы взаимодействия – см. части 1 и 2).

Необходимо подчеркнуть, что политика безопасности, мониторинг взаимодействий и логическая защита ЯЗ являются обязательными компонентами всех Профилей защиты вне зависимости от назначения и среды применения ПИТ.

Физическая защита ЯЗ. Требования этой группы задают ограничения на физический доступ к компонентам ЯЗ, а также допустимые физические параметры среды функционирования КС.

Самоконтроль ЯЗ. Требования, касающиеся самоконтроля ЯЗ, определяют возможности обеспечения контроля корректности выполнения функций ЯЗ и целостности программ и данных, входящих в ЯЗ. Выполнение этих требований позволяет вовремя обнаруживать нарушения целостности компонентов ЯЗ, произошедшие либо в результате целенаправленного воздействия, либо вследствие сбоя в работе аппаратных или программных средств, и осуществлять восстановление целостности ЯЗ.

Инициализация и восстановление ЯЗ. Требования данной группы устанавливают возможности ЯЗ по контролю за процессом собственной инициализации и способности к самовосстановлению после сбоев. Процесс восстановления после сбоя должен происходить без нарушений функционирования, даже временного, средств защиты. Восстановленное состояние ЯЗ должно соответствовать требованиям политики безопасности, мониторинга взаимодействий и самоконтроля целостности.

Ограничение привилегий при работе с ЯЗ. Требования этой группы устанавливают порядок назначения полномочий для работы с ЯЗ. Основным принципом назначения таких полномочий является принцип минимальной достаточности. Это обеспечивается посредством постоянного контроля и, при необходимости, автоматического понижения привилегий пользователей при обращении к компонентам или сервису ЯЗ. Соблюдение этого принципа позволяет минимизировать нарушения целостности в случае возникновения сбоев или нарушений безопасности.

Простота использования ЯЗ. Эти требования обеспечивают удобство пользования возможностями ЯЗ как для высококвалифицированных администраторов, ответственных за функционирование и безопасность системы, так и для рядовых пользователей, а также для разработчиков прикладных программ, взаимодействующих с ЯЗ. К этому классу требований относятся: порядок реагирования ЯЗ на ошибки в действиях пользователей и попытки нарушения безопасности, устанавливаемые по умолчанию полномочия, интерфейс пользователей и администратора.

Объем и глубина реализации функциональных требований зависит от того, какую степень защищенности должно обеспечивать ядро защиты конкретного ПИТ, а также от того, какие угрозы безопасности возможны в среде его эксплуатации. Степень обеспечения требуемого уровня защищенности зависит от реализованной политики безопасности, от квалификации ответственного за безопасность персонала, от правильности администрирования ЯЗ и соблюдения рядовыми пользователями правил политики безопасности.

Ранжирование функциональных требований

Состав и содержание включенных в Профиль защиты функциональных требований определяются средой эксплуатации ПИТ. Чтобы обосновать выбор тех или иных требований и не вступать в противоречие с существующими стандартами в области безопасности ПИТ, функциональные требования, приведенные в «Федеральных критериях», ранжируются по уровням с помощью следующих четырех критериев: широта сферы применения, степень детализации, функциональный состав средств защиты, обеспечиваемый уровень безопасности.

Широта сферы применения определяется множеством сущностей, к которому могут быть применены данные требования, а именно:

- пользователи системы, субъекты и объекты доступа;
- функции ЯЗ и интерфейс взаимодействия с ЯЗ;
- аппаратные, программные и специальные компоненты ЯЗ;
- множество параметров конфигурации ЯЗ.

Например, требования из разделов управления доступом, аудита, обеспечения работоспособности, мониторинга взаимодействий и простоты использования ЯЗ могут относиться только к определенному подмножеству объектов доступа и параметров конфигурации ЯЗ. Обеспечение прямого взаимодействия с ЯЗ требуется только для некоторого подмножества функций ЯЗ.

Степень детализации требований определяется множеством атрибутов сущностей, к которым применяются данные требования – либо ко всем атрибутам пользователей, субъектов или объектов, либо только к некоторому подмножеству этих атрибутов. Например, требования из разделов управления доступом, аудита и мониторинга взаимодействий могут относиться только к некоторому подмножеству атрибутов субъектов и объектов – к правам доступа, групповым идентификаторам пользователей, но не к атрибутам состояния субъектов и объектов и не к индивидуальным идентификаторам.

Функциональный состав средств защиты определяется множеством функций, включенных в ЯЗ для реализации той или иной группы функциональных требований. Например, политика управления доступом может включать либо произвольное, либо нормативное управление доступом, или и то, и другое одновременно.

Обеспечиваемый уровень безопасности определяется условиями, в которых функциональные компоненты ЯЗ способны противостоять заданному множеству угроз, отказам и сбоям. Например, нормативное управление доступом обеспечивает более высокий уровень безопасности, чем произвольное.

Ранжирование всегда предполагает установление некоторого отношения порядка. Однако независимое ранжирование функциональных требований по каждому из описанных критериев, хотя и дает некоторое представление о различиях между функциональными возможностями средств защиты, но не позволяет установить четкую, линейную шкалу уровней безопасности. Строгого отношения порядка, определенного на множестве функциональных требований, не существует, т.к. значение требований и уровень обеспечиваемой ими защиты зависят не только от их содержания, но и от назначения ПИТ и среды его эксплуатации. Для одних систем наиболее важными будут идентификация и аутентификация пользователей, а для других – реализация политики управления доступом или обеспечение доступности.

В связи с этим в «Федеральных критериях» отсутствуют рекомендации как по выбору и применению тех или иных функциональных требований, так и по определению их роли в системе обеспечения безопасности. Вместо жестких указаний этот документ содержит согласованный с предшествующими ему стандартами («Оранжевая книга», «Европейские критерии») ранжированный перечень функциональных требований и предоставляет разработчикам Профиля защиты возможность самостоятельно сделать выбор необходимых методов и средств обеспечения безопасности, основанный на назначении и специфике среды эксплуатации ПИТ.

Приводимое ранжирование не противоречит предшествующим стандартам и вводится для исключения ошибок в определении степени защищенности системы из-за неправильной оценки значимости отдельных групп требований. Кроме того, ранжирование предоставляет разработчикам и пользователям возможность для обоснованной оценки реально обеспечиваемого уровня безопасности.

Применение критериев ранжирования к различным группам функциональных требований представлено в таблице. Знак «*» указывает, на какие значимые свойства КС (широта сферы применения, степень детализации и т.д.) влияет реализация того или иного защитного механизма.

Таблица 4. Применение критериев ранжирования

Функциональные требования	Широта сферы применения	Степень детализации	Функциональный состав средств защиты	Обеспечиваемый уровень безопасности
Реализация политики безопасности				
Политика аудита				
Идентификация и аутентификация			*	*
Регистрация в системе			*	
Обеспечение прямого взаимодействия с ЯЗ	*		*	
Регистрация и учет событий			*	*
Политика управления доступом	*	*	*	
Контроль скрытых каналов	*		*	
Политика обеспечения работоспособности				
Контроль за распределением ресурсов	*		*	
Отказоустойчивость	-	-	-	-
Управление безопасностью			*	*
Мониторинг взаимодействий	*	*	*	
Логическая защита ЯЗ			*	
Физическая защита ЯЗ			*	*
Самоконтроль ЯЗ	*		*	
Инициализация и восстановление ЯЗ			*	
Ограничение привилегий при работе с ЯЗ		*		
Простота использования ЯЗ	*		*	

9.4.5. Требования к процессу разработки продукта информационных технологий

Основное назначение требований к технологии разработки ПИТ – обеспечить адекватность условий разработки функциональным требованиям, выдвинутым в соответствующем разделе Профиля защиты, и установить ответственность разработчика за корректность реализации этих требований. Данный раздел регламентирует процесс создания, тестирования, документирования и сопровождения ПИТ. Требования к технологии разработки ПИТ включают четыре раздела: требования к процессу разработки, к среде разработки, документированию и сопровождению.

Требования к процессу разработки содержат подразделы, относящиеся к проектированию, реализации, тестированию и анализу ПИТ. Особую роль играют требования адекватности реализации функций ЯЗ, обеспечивающие корректность выполнения функциональных требований Профиля защиты.

Требования к среде разработки позволяют обеспечить качество процесса создания ПИТ с помощью применения современных технологий проектирования, программирования и тестирования, а также регламентируют управление процессом разработки и дистрибуцию конечного продукта.

Требования к документированию определяют состав и содержание технологической документации, позволяющей производителю ПИТ доказать соответствие самого продукта и технологии его изготовления выдвинутым требованиям.

Требования к сопровождению ПИТ содержат обязательства производителя перед пользователями, выполнение которых позволяет обеспечить эффективную и надежную эксплуатацию ПИТ. Данные требования регламентируют состав пользовательской и административной документации, процедуру обновления версий и исправления ошибок, а также инсталляцию продукта.

«Федеральные критерии» содержат ранжированный перечень типовых требований к технологии разработки ПИТ [17]. Выполнение требований к технологии разработки является необходимым условием для проведения процедуры сертификации.

9.4.6. Требования к процессу сертификации продукта информационных технологий

Требования к процессу сертификации ПИТ призваны обеспечить надежность и корректность процесса анализа ПИТ на соответствие выдвинутым функциональным требованиями и требованиям к технологии разработки. Раздел содержит три группы требований, регламентирующих анализ, контроль и тестирование ПИТ. Раздел требований к анализу ПИТ содержит требования к проведению независимого анализа предложенного решения (архитектуры) и его реализации как конкретного средства.

Раздел требований к контролю регламентирует проверку соответствия среды разработки ПИТ и обеспечиваемого производителем сопровождения требованиям к технологии разработки.

Требования к тестированию описывают процедуру проведения тестирования функций ЯЗ как самим разработчиком ПИТ, так и независимыми экспертами.

Эти требования регламентируют процесс сертификации только в общих чертах и, по замыслу разработчиков стандарта, должны послужить основой для разработки специализированных методик сертификации, ориентированных на различные области применения и классы ПИТ.

9.4.7. Выводы

«Федеральные критерии безопасности информационных технологий» являются первым стандартом в области безопасности систем обработки информации, в котором определены и рассмотрены три независимые группы требований: функциональные требования к средствам защиты, требования к технологии разработки и к процессу сертификации. Авторами этого документа предложена концепция Профиля защиты – документа, содержащего полное описание всех требований безопасности, к процессу разработки, сертификации и эксплуатации ПИТ.

Функциональные требования к средствам защиты четко структурированы и описывают все аспекты функционирования КС. Требования к технологии разработки, впервые появившиеся в этом документе, позволяют разработчикам использовать современные технологии программирования в качестве основы для подтверждения безопасности своего продукта. Требования к процессу сертификации носят довольно общий характер и не содержат конкретных методик тестирования и исследования ПИТ.

Разработчики «Федеральных критериев» отказались от используемого в «Оранжевой книге» подхода к оценке уровня безопасности ПИТ путем введения обобщенной универсальной шкалы классов безопасности. Вместо этого предлагается независимое ранжирование требований по каждой группе, т.е. вместо одной шкалы используется множество частных критериев, характеризующих обеспечиваемый уровень безопасности. Данный подход позволяет разработчикам и пользователям ПИТ выбрать наиболее приемлемое решение и определить необходимый и, в ряде случаев, достаточный набор требований для каждого конкретного случая.

9.5. Общие критерии

Как было сказано выше, в начале 80-х годов в США был разработан документ «Критерии оценки доверенной компьютерной системы» (TCSEC). В следующем десятилетии начали проводиться исследования по разработке критериев оценки, которые построены на концепциях TCSEC, но являются более гибкими и приспособленными к различным аспектам использования ПИТ.

В Европе Европейской комиссией после совместной разработки с участием Франции, Германии, Нидерландов и Англии в 1991г. была опубликована версия 1.2 «Критериев оценки безопасности информационной технологии» (ITSEC). В Канаде к началу 1993 г. была опубликована версия 3.0 «Канадских критериев оценки доверенных компьютерных продуктов» (CTCPEC) как объединение подходов ITSEC и TCSEC. В

США в начале 1993 г. публикуется описанная выше версия 1.0 «Федеральных критериев для безопасности информационной технологии» (FC) как второй подход комбинирования Североамериканской и Европейской Концепций для критериев оценки безопасности.

Параллельно в 1990 г. была начата работа в Международной Организации Стандартов (ISO) по разработке международных критериев оценки безопасности КС для общего использования. Новые критерии должны были удовлетворять условиям взаимного признания результатов оценки безопасности на глобальном рынке ПИТ. Задача была поставлена Рабочей группе 3 (WG3) 27 подкомитета (SC27) Объединенного технического комитета 1 (JTC1). Первоначально в рамках WG3 работа протекала довольно медленно из-за большого объема первичной работы по согласованию различных документов и интенсивных многосторонних переговоров.

В июне 1993 г. организации-участники проектов CTCPEC, FC, TCSEC и ITSEC (перечислены ниже) объединили усилия по созданию частных нормативных документов по информационной безопасности и начали совместную деятельность по упорядочению своих критериев в одно множество критериев безопасности ПИТ, которые могли бы использоваться в рамках международного сотрудничества в данной области. Эта деятельность была названа Проектом Общих критериев (ОК). Его целью должно было стать разрешение концептуальных и технических отличий, обнаруженных в предыдущих нормативных документах-источниках критериев, и представление результатов в ISO как исходных положений по разработке международного стандарта. Представители организаций-участников сформировали для разработки ОК Издательский совет ОК (ОКЕВ). Затем было установлено взаимодействие между ОКЕВ и Рабочей группой-3 (WG3), и Издательский совет выдал несколько ранних версий ОК в WG3. Как результат взаимодействий между WG3 и ОКЕВ эти версии были приняты в качестве успешных рабочих проектов различных Частей критериев ISO, начиная с 1994 г.

Версия 1.0 ОК была завершена ОКЕВ в январе 1996 г. и была одобрена ISO в апреле 1996 г. для рассылки как Проект Комитета (CD). Затем Проект ОК претерпел ряд изменений, основанных на отзывах, полученных от экспертных организаций. Работа по учету отзывов и замечаний была проведена преемником ОКЕВ, теперь называемым Советом по выполнению ОК (CCIB).

Совет по выполнению ОК завершил версию 2.0 ОК «Бета» в октябре 1997 г. и представил ее в WG3, который одобрил ее в качестве Второго проекта. Последующие промежуточные версии проекта сопровождались неофициально экспертами WG3 для поддержки обратной связи, так как они формально выпускались CCIB. CCIB принимал отзывы и отвечал на те отзывы, которые приходили как от специалистов WG3, так и из национальных организаций ISO в процессе голосования. Результатом этого процесса стало появление Версии 2 Общих Критериев.

В целях сохранения историчности и непрерывности развития ISO/IEC JTC 1/SC 27/WG3 сохранил в документе использование термина «Общие Критерии» (ОК), однако его официальное название – «Критерии Оценки безопасности информационной технологии».

Организации-участники Проекта Общих Критериев

Семь европейских и североамериканских государственных организаций, перечисленных ниже, являются организациями-участниками и организаторами Проекта ОК. Они обеспечили почти весь цикл работ от начала разработки стандарта до их завершения. Указанные организации являются, как правило, также «оценивающими авторитетными организациями» для своих национальных правительств. Кроме того, в настоящее время данными организациями приняты неформальные обязательства заменить свои критерии оценки версией 2.0 ОК при переходе к окончательной стадии принятия их в качестве международного стандарта.

Канада

Координатор установления критериев безопасности связи подразделение 12А «Безопасность сетей и компьютеров».

Германия

Федеральная служба безопасности в области информационной техники.

Агентство информационной безопасности Германии (GISA).

Франция

Центральная служба безопасности информационных систем.

Центр сертификации безопасности информационных технологий.

Нидерланды

Национальное агентство безопасности связи Нидерландов.

Англия

Группа безопасности связи – электроники.

Служба оценки безопасности компьютеров.

США-NSA

Агентство национальной безопасности.

США-NIST

Национальный институт стандартов и технологии.

Отделение безопасности компьютеров.

Международные стандарты проходят стадии разработки в соответствии с правилами, данными в Директивах ISO/IEC, Часть 3.

В области информационной технологии ISO и IEC образовали объединенный технический комитет ISO/IEC JTC 1. Проекты международных стандартов, принятые объединенным техническим комитетом, передаются национальным организациям для обсуждения и последующего голосования. Публикация в качестве Международного стандарта требует одобрения по меньшей мере 75% Национальных организаций, участвующих в голосовании.

Международный стандарт ISO/IEC 15408, тождественный Общим Критериям был подготовлен объединенным техническим комитетом

ISO/IEC JTC 1, **Информационная технология**, в сотрудничестве с организациями, спонсирующими Проект Общих Критериев. Идентичный текст ISO/IEC 15408-1 опубликован организациями – спонсорами Проекта Общих Критериев как **«Общие критерии оценки безопасности информационных технологий»**.

Документ ISO/IEC 15408 состоит из следующих частей:

Часть 1: Введение и общая модель.

Часть 2: Функциональные требования безопасности.

Часть 3: Гарантийные требования безопасности.

Семь правительственных организаций (называемые «Организации, спонсирующие Проект Общих Критериев»), определенные в Приложении А, ISO/IEC 15408-1, как объединенные держатели копий «Общих критериев для оценки безопасности информационных технологий», части с 1 по 3 (называемые «ОК»), предоставляют свободную лицензию для ISO/IEC на использование ОК в разработке международного стандарта ISO/IEC 15408. Однако, организации – участники Проекта Общих Критериев, оставляют за собой права по использованию, копированию, распределению или модификации ОК по своему усмотрению.

9.5.1. Область применения

Стандарт ISO/IEC 15408, состоящий из нескольких частей, определяет критерии, которые должны использоваться как основа для оценки свойств безопасности продуктов информационных технологий (ПИТ). Посредством установления такой базы общих критериев результаты оценки безопасности ПИТ должны быть понятны широкой аудитории.

ОК должны позволять сравнивать результаты независимых оценок свойств безопасности КС. Эта возможность достигается путем построения общего множества требований для функций безопасности продуктов и систем ПИТ, а также мер по гарантиям выполнения ПБ (в тексте ОК-гарантиям), применяемым к ним при оценке безопасности. Процесс оценки устанавливает уровень доверия тому, что функции безопасности оцениваемых продуктов и систем и меры гарантии, применяемые к ним, отвечают требованиям. Результаты оценки могут помочь потребителю определить, являются ли ПИТ достаточно безопасными для их предполагаемого использования.

ОК полезны как руководство для разработки защищенных ПИТ или систем с функциями безопасности и для закупки коммерческих продуктов и систем с такими функциями. Во время оценки такой продукт ПИТ или система называются Объектом Оценки (ООц) (англоязычный исходный термин Target Of Evaluation, TOE). Объекты оценки включают, в частности, операционные системы, вычислительные сети, распределенные системы и прикладные приложения.

ОК рассматривают защиту информации от несанкционированного доступа, модификации или потери возможности использования (потерю доступности). Категории защиты, относящиеся к этим трем типам угроз безопасности, обычно называются конфиденциальностью, целостностью и доступностью. ОК также могут применяться к аспектам безопасности ПИТ вне этой тройки. ОК рассматривает угрозы информации, возникающие от человеческой деятельности (преднамеренной или непреднамеренной). Кроме того, ОК могут применяться и против

некоторых угроз, не связанных с человеческим фактором. ОК могут применяться и в других областях ПИТ, но не претендуют на корректность вне области безопасности ПИТ.

ОК применяются к мерам безопасности ПИТ, реализуемым в виде аппаратной компоненты, встроенных программ, программного обеспечения. Там, где конкретные аспекты оценки предназначаются для применения только к определенным методам реализации, это указано в соответствующих пунктах критериев. Ниже изложены естественные ограничения на область действия ОК.

a) ОК не содержат критерии оценки безопасности, относящиеся к административным мерам безопасности. Однако признается, что значительная часть мер безопасности ООц часто может составлять административные меры. Административные меры безопасности в среде ООц трактуются как **предположения** (т.е. исходные положения) безопасного обращения там, где они оказывают влияние на качество мер безопасности ПИТ. Например, исходным положением надежности криптографической системы является сохранение в тайне ключа шифрования, которое реализуется комплексом организационных мер.

b) Оценка технических, физических аспектов безопасности ПИТ таких, как контроль побочных излучений, специально не охватывается, хотя многие рассматриваемые концепции документа могут применяться и к этой области. В частности, эти ОК рассматривают некоторые аспекты физической защиты ООц.

c) ОК не рассматривают ни методологию оценки, ни административные и юридические структуры, в рамках которых критерии могут применяться специалистами по оценке. Однако ожидается, что эти ОК будут использованы для целей оценки в контексте таких структур и такой методологии.

d) Процедуры для использования результатов оценки при аккредитации (сертификации) продукта или системы находятся вне области действия ОК. Аккредитация продукта или системы является административным процессом, посредством которого гарантируется безопасность операций продукта или системы ПИТ в их полном жизненном цикле. Оценка сосредотачивается на тех сторонах безопасности ПИТ, которые могут прямо воздействовать на безопасное использование элементов ПИТ. Результаты процесса оценки впоследствии являются входными данными для процесса аккредитации.

e) Область критериев для оценки качеств криптографических алгоритмов не охватывается в ОК. Должна быть произведена отдельная оценка математических свойств криптографических модулей, встроенных в Ооц.

9.5.2. Общие сокращения

Следующие сокращения являются общими для разных частей ОК:

ОК Общие Критерии – название, исторически используемое для стандарта ISO/IEC 15408, состоящего из нескольких частей, вместо его официального названия ISO «Критерии оценки безопасности информационных технологий» (Common Criteria)

EAL Уровень гарантии оценки (УГ) (Evaluation Assurance Level)

IT	Информационная технология (Information Technology)
PP	Профиль Защиты (ПЗ) (Protection Profile)
SF	Функция безопасности (Security Function)
SFP	Политика функции безопасности (Security Function Policy)
SOF	Стойкость (надежность) функции безопасности (Strength Of Function)
ST	Задание по Безопасности (ЗБ)(Security Target)
TOE	Объект оценки (Target Of Evaluation)
TSC	Область действий TSF (TSF Scope of Control)
TSF	Функции безопасности Ооц (ООц Security Functions)
TSFI	Интерфейс функций безопасности (TSF) (TSF Interface)
TSP	Политика безопасности (ПБ) ООц (ООц Security Policy)

9.5.3. Сфера глоссария

Приводимый ниже раздел 9.5.4 содержит только те термины, которые используются в ОК специальным образом. Большинство терминов в ОК используется в соответствии с принятыми определениями в словарях или в соответствии с общепринятыми определениями, которые можно найти в глоссариях по безопасности ISO или в других хорошо известных перечнях терминов по безопасности. Некоторые комбинации общих терминов, используемые в ОК, хотя и не включены в глоссарий, но для ясности объясняются в том контексте, в котором они используются. Объяснения терминов и концепций, используемых специальным образом в ISO/IEC 15408-2 и ISO/IEC 15408-3, могут быть найдены в соответствующих разделах.

9.5.4. Глоссарий

Авторизованный пользователь - (Authorised user)	Пользователь, который может, в соответствии с Политикой безопасности (TSP), выполнять операции.
Атрибут безопасности - (Security attribute)	Информация, связанная с субъектами, пользователями и/или объектами, используемая для реализации TSP.
Внешние объекты ПИТ - (External IT Entity)	Любой ПИТ, доверенный или не доверенный, вне ООц, которые взаимодействуют с ООц.
Внутренний канал связи - (Internal communication channel)	Канал связи между отдельными частями ООц.
Внутренняя передача в ООц - (Internal TOE transfer)	Передача данных между отдельными частями ООц.
Выбор- (Selection)	Выделение одного или нескольких элементов из перечня в компоненте.

Высокая SOF - (SOF-high)	Уровень стойкости функции безопасности ООц, на котором в соответствии с результатами анализа обеспечивается адекватная защита против преднамеренно запланированного или организованного нарушения безопасности ООц нарушителями с высоким потенциалом атаки.
Гарантия - (Assurance)	Основание для уверенности, что объект удовлетворяет целям его безопасности.
Данные TSF - (TSF data)	Данные, созданные ООц и для ООц, которые могут воздействовать на работу ООц.
Данные аутентификации - (Authentication data)	Информация, используемая для аутентификации пользователя.
Данные пользователя - (User data)	Данные, созданные пользователем и для (другого) пользователя, которые не воздействуют на работу TSF.
Зависимость - (Dependency)	Отношение между требованиями – такое, что одно из них необходимо для того, чтобы другие требования были выполнимы.
Доверенный канал - (Trusted channel)	Средства, через которые TSF и удаленный доверенный продукт ПИТ могут связаться (с гарантией выполнения ПБ).
Доверенный маршрут - (Trusted path)	Средства, через которые пользователь и TSF могут связаться (с гарантией выполнения ПБ).
Задание по Безопасности (ST) - (Security Target)	Множество требований безопасности и спецификаций, которые должны использоваться в качестве основы для оценки определенного ООц.
Идентификатор - (Identity)	Представление (например, строка), однозначно определяющее авторизованного пользователя, которое может быть либо полным, либо сокращенным именем этого пользователя или его псевдонимом.
Имущество - (Assets)	Информация или ресурсы (в приводимых выше моделях – объекты), которые должны быть защищены.
Интерфейс функции безопасности ООц (TSFI) - (TOE Security Functions Interface)	Множество интерфейсов, либо интерактивных (интерфейс человек-машина), либо программных (интерфейс прикладного программирования), через которые осуществляется доступ к ресурсам ООц, используемым TSF, или к информации, получаемой от TSF (см. также главу 2).

Итерация (повторение) - (Iteration)	Использование компонента более одного раза различными операциями.
Класс - (Class)	Совокупность объектов, объединенных общим назначением (целевой функцией).
Компонент - (Component)	Минимальное выделяемое множество элементов, которое может быть включено в ПЗ, ЗБ или пакет.
Механизм реализации ссылок - (Reference validation mechanism)	Реализация концепции монитора ссылок (МБО), которая обладает следующими свойствами: она защищена от проникновения, всегда активизирована и достаточно проста для осуществления ее тщательного анализа и тестирования.
Модель политики безопасности ООц - (TOE Security Policy model)	Формализованное представление политики безопасности, которая должна быть реализована Ооц (см. также главу 1).
Монитор ссылок - (Reference monitor)	Концепция абстрактной машины, реализующей политику контроля доступа в ООц.
Назначение - (Assignment)	Спецификация некоторого заданного параметра в компоненте.
Неформальный - (Informal)	Выраженный на естественном языке.
Номинальная SOF - (SOF-basic)	Уровень стойкости функции безопасности ООц, на котором в соответствии с результатами анализа обеспечивается адекватная защита от случайного нарушения безопасности ООц нарушителями с низким потенциалом атаки.
Область действия TSF (TSC) - (TSF Scope of Control)	Множество взаимодействий, которые могут произойти с или внутри ООц и подчиняются правилам реализации ПБ.
Объект - (Object)	Некоторая сущность внутри TSC, который содержит или получает информацию, над которой субъект выполняет операции.
Объект оценки(ООц) - (Target of Evaluation)	ПИТ и связанная с ними документация в виде руководства администратора и руководства пользователя, которые являются предметом оценки.
Орган оценки - (Evaluation authority)	Организация, которая посредством системы оценки применяет ОК для определенного сообщества, устанавливает стандарты и проверяет качество оценок, проводимых другими органанизациями внутри этого сообщества.

Организационные политики безопасности (Organisational security policies)	Одно или более правил безопасности, процедур, указаний или руководств, накладываемых организацией на свои операции.
Оценка - (Evaluation)	Оценка ПЗ, ЗБ или ООц по определенным критериям.
Пакет - (Package)	Многократно используемое множество функциональных или гарантийных компонентов (например, EAL), собранных вместе, чтобы удовлетворять определенному множеству целей безопасности.
Передача за пределы области действия TSF - (Transfers outside TSF control)	Передача данных к элементам системы, находящимся вне области действия TSF.
Передачи между TSF - (Inter – TSF transfers)	Передачи данных между ООц и функциями безопасности других доверенных продуктов ПИТ.
Политика безопасности ООц (TSP) - (TOE Security Policy)	Множество априорно заданных правил, которые регулируют существование объекта защиты в ООц.
Политика функции безопасности (SFP) - (Security Function Policy)	Политика безопасности, реализуемая SF.
Пользователь - (User)	Любой элемент (человек или внешний элемент) вне ООц, который взаимодействует с ним.
Полуформальный - (Semiformal)	Выраженный в языке ограниченного синтаксиса с помощью определенных семантик.
Потенциал атаки - (Attack potential)	Осознанная возможность успешной атаки, которая может быть предпринята злоумышленником, выраженная в терминах квалификации атакующего, его ресурсов и мотивации.
Продукт - (Product)	Совокупность программных, программно-аппаратных и/или аппаратных средств, предоставляющая определенные функциональные возможности и предназначенная как для непосредственного использования, так и для включения в различные системы.
Профиль Защиты (ПЗ) (Protection Profile)	Независимое от реализации множество требований безопасности для некоторой категории ООц, которые отвечают некоторой целевой функции (например, профиль межсетевого экрана – см. главу 7).

Расширение - (Extension)	Добавление к ПЗ или ЗБ функциональных и/или гарантийных требований.
Ресурс ООц - (TOE resource)	Сущности, используемые в ООц.
Роль - (Role)	Предопределенное множество правил, устанавливающих разрешенное взаимодействие между пользователем и ООц.
Связность - (Connectivity)	Свойство ООц, которое позволяет взаимодействовать с объектами ПИТ, внешними по ноошению к ООц.
Секрет - (Secret)	Информация, которая должна быть известна только авторизованным пользователям и/или TSF, чтобы реализовать определенную SFP.
Семейство - (Family)	Совокупность компонентов, имеющих общие цели безопасности, но которые могут отличаться в акцентах или строгости.
Система - (System)	Определенная установка (инсталляция) ПИТ с определенной целью и операционной средой.
Система оценки - (Evaluation scheme)	Организационно-правовая структура, в рамках которой применяются ОК для оценки экспертной организацией в определенном сообществе.
Средняя SOF - (SOF-medium)	Уровень стойкости функции безопасности ООц, на котором в соответствии с результатами анализа обеспечивается адекватная защита против прямого или намеренного нарушения безопасности ООц нарушителями с умеренным потенциалом атаки.
Стойкость функции безопасности (SOF) - (Strength of Function)	Характеристика функции безопасности ООц, выражающая минимально необходимо воздействие непосредственно на ее механизмы безопасности, в результате которого нарушается ПБ в части этой функции.
Субъект - (Subject)	Элемент системы внутри TSC, который инициирует выполнение операций (активная компонента КС – см. также главу 1).
Уровень гарантии оценки (EAL) - ((Evaluation Assurance Level (EAL))	Пакет требований, состоящий из компонентов гарантий выполнения ПБ.
Усиление - (Augmentation)	Добавление одного или более компонентов гарантии к EAL (уровню гарантии оценки) или пакету гарантий.
Уточнение - (Refinement)	Дополнение деталей к компоненту.

Формальный - (Formal)	Выраженный в языке с ограниченным синтаксисом с помощью определенных семантик, основанных на точных математических концепциях.
Функция безопасности (SF) - (Security Function)	Часть или части ООц, на которые возлагается реализация тесно связанного подмножества правил из TSP.
Функции безопасности ООц (TSF) - (TOE Security Functions)	Множество, состоящее из всего аппаратного оборудования, программного обеспечения и встроенных программ ООц, которые ответственны за правильную реализацию TSP.
Цель безопасности - (Security objectives)	Сформулированное намерение противостоять определенным угрозам и/или удовлетворить определенной ПБ.
Человек-пользователь - (Human user)	Любое лицо, которое взаимодействует с ООц.
Элемент (требований) - (Element)	Элементарное («неделимое») требование безопасности.

Как видно из глоссария, в ОК четко выделяются задачи реализации ПБ и ее гарантирования. Кроме того, сделано четкое различение субъекта и человека-пользователя, введены понятия безопасного межсубъектного обмена. Таким образом, ОК являются максимально адекватным современным моделям безопасности КС.

9.5.5. Введение в Общие критерии

Продукты или системы ПИТ должны выполнять свои функции, одновременно с выполнением априорно заданной ПБ. Таким образом, термин «безопасность ПИТ» используется для обозначения защищенности отразличного рода нарушений ПБ.

Многие потребители ПИТ не имеют ни знаний, ни опыта, ни ресурсов, необходимых для того, чтобы судить о степени защищенности их продуктов или систем ПИТ. Им приходится полностью полагаться только на заявления разработчиков. Потребители могут пожелать повысить свою уверенность в мерах безопасности продукта или системы ПИТ, заказав анализ их безопасности (т.е. оценку безопасности).

ОК могут быть использованы для выбора соответствующих мер безопасности ПИТ и содержат критерии для оценки требований безопасности.

Пользователи ОК

Существуют три группы с общими интересами в оценке свойств безопасности продуктов и систем ПИТ: потребители ООц, разработчики ООц и оценивающие ООц (эксперты). Критерии, приводимые ниже, построены так, чтобы поддерживать интересы всех трех групп. Принципиально все они считаются пользователями ОК.

Потребители

ОК играют важную роль при выборе потребителем требований безопасности ПИТ для выражения их нужд. ОК пишутся для гарантии того, что оценка удовлетворяет нужды потребителей, так как это является основной целью процесса оценки.

Потребители могут использовать результаты оценок, чтобы решить, действительно ли оцениваемый ПИТ удовлетворяют их нужды безопасности. Эти потребности в безопасности обычно определяются в результате как анализа риска, так и выбора направления политики безопасности. Потребители могут также использовать результаты оценки для сравнения различных продуктов или систем. Представление требований по гарантиям в виде иерархии поддерживает эту потребность.

ОК дают потребителям – особенно в группах потребителей, объединенных общими интересами – структуру, независимую от реализации, называемую Профилем Защиты (PP), в котором выражаются их специальные требования по мерам безопасности ПИТ в ООц.

Разработчики

ОК предназначены для поддержки разработчиков в подготовке и помощи в оценке их продуктов и систем, а также в определении требований безопасности, которым должны удовлетворять каждый из их продуктов или систем. Возможно также, что соответствующая методология оценки, потенциально сопровождаемая взаимным соглашением по признанию результатов оценки, в дальнейшем должна позволить ОК поддержать каждого, не разработчика ООц, в подготовке и помощи в оценке ООц разработчика.

Конструкции ОК могут тогда использоваться для утверждения того, что ООц удовлетворяет определенным для него требованиям посредством определенных функций безопасности и гарантий, которые должны быть оценены. Требования каждого ООц содержатся в зависимой от реализации конструкции, называемой Заданием по Безопасности (ST). Один или более Профилей Защиты в ЗБ могут обеспечить требования широкой аудитории потребителей.

ОК описывают функции безопасности, которые разработчик может включить в ООц. ОК могут использоваться для определения ответственностей и действий, необходимых для поддержки оценки ООц. Они также определяют содержание и представление этой оценки.

Оценивающие (эксперты)

ОК содержат критерии, которые должны использоваться оценивающими (экспертами) при формировании суждений о соответствии ООц требованиями по их безопасности. ОК описывают множество общих действий, которые должен выполнить эксперт, и функции безопасности, в соответствии с которыми выполняются эти действия. Отметим, что ОК не определяют процедуры, которым надо следовать в проведении этих действий.

Другие

Хотя ОК ориентированы на определение и оценку свойств безопасности ПИТ в различных ООц, они также могут быть полезны в качестве

справочного материала для всех, кто заинтересован в безопасности или отвечает за безопасность ПИТ. Некоторыми из дополнительных групп по интересам, которые могут получить пользу от информации, содержащейся в ОК, являются:

a) офицеры (служащие) безопасности системы, ответственные за определение и выполнение ПБ и других требований безопасности ПИТ;

b) аудиторы, как внутренние, так и внешние, ответственные за оценку адекватности безопасности системы;

c) создатели и разработчики систем безопасности, ответственные за спецификацию содержания безопасности систем и продуктов ПИТ;

d) аккредиторы, ответственные за ввод (принятие) системы ПИТ в эксплуатацию в конкретной среде (обстановке);

e) спонсоры оценки (организации-заказчики сертификации), ответственные за запрос и поддержку оценки;

f) должностные лица по оценке, ответственные за управление и надзор за программами по оценке безопасности ПИТ.

Контекст оценки

Для того, чтобы достичь большей сравнимости результатов оценки между собой, оценки должны выполняться в рамках согласованной системы (схемы) оценки, которая учитывает стандарты, следит за качеством оценок и управляет правилами, которым должны соответствовать средства оценки и оценивающие.

Использование общей методологии оценок содействует воспроизводимости и объективности результатов, но само по себе не является достаточным. Многие из критериев оценки требуют применения экспертного заключения и дополнительных знаний и навыков, по которым труднее достичь согласованности. Чтобы повысить согласованность полученных данных по оценке, окончательные результаты оценки можно подвергнуть процессу сертификации. Процесс сертификации является независимой проверкой результатов оценки, ведущей к выработке конечного сертификата или подтверждения. Сертификат обычно публично доступен. Надо отметить, что процесс сертификации является средством получения большей согласованности в применении критериев безопасности ПИТ.

Система оценки, методология и процессы сертификации определяются организациями по оценке, которые применяют схемы оценки, и находятся вне сферы ОК.

9.5.6. Состав Общих Критериев (ОК)

ОК представляются как множество связанных частей:

a) Часть 1, Введение и общая модель, является введением в ОК и определяет общие концепции и принципы оценки безопасности ПИТ и представляет общую модель оценки. Часть 1 также представляет конструкции для выражения целей безопасности ПИТ, для выбора и определения требований безопасности ПИТ и для написания спецификаций высокого уровня для продуктов и систем.

b) Часть 2, Функциональные требования безопасности, устанавливает множество функциональных компонентов как стандартный способ выражения функциональных требований для ООц. В части 2 дается каталог множеств функциональных компонентов, семейств и классов.

c) Часть 3, Гарантийные требования безопасности, устанавливает множество компонентов по гарантиям как стандартный способ выражения требований по гарантиям для ООц. В Части 3 дается каталог множеств компонентов, семейств и классов по гарантиям. Часть 3 также определяет критерии оценки для ПЗ и ЗБ и представляет уровни гарантии оценки, которые определяют шкалу ОК рейтинга гарантии для ООц, которая называется шкалой Уровней Гарантии Оценки (EAL).

Предполагается, что в поддержку этих трех частей ОК, перечисленных выше, должны быть опубликованы другие виды документов, включающие материалы технического обоснования и руководящие документы.

Следующая таблица представляет для трех групп потенциальных потребителей их интересы в каждой части ОК.

Путеводитель по Общим Критериям

	Потребители	Разработчики	Оценивающие
Часть 1	Используют как базовую информацию и для ссылок. Руководство по структуре ПЗ.	Используют как базовую информацию и для ссылок при разработке требований и формулировании спецификаций безопасности для ООц.	Используют как базовую информацию и для ссылок. Руководство по структуре ПЗ и ЗБ.
Часть 2	Используют для руководства и ссылок при формулировании утверждений требований для функций безопасности.	Используют для ссылок при интерпретации утверждений функциональных требований и формулировании функциональных спецификаций для ООц.	Используют как обязательный документ по критериям оценки при определении эффективности заявленных функций безопасности для ООц.
Часть 3	Используют для руководства при определении требуемых уровней гарантии.	Используют для ссылок при интерпретации заявленных требований по гарантиям и определении подходов по гарантиям ООц.	Используют как обязательный документ по критериям оценки при определении гарантий ООц и при оценке ПЗ и ЗБ.

9.5.7. Общая модель

Данная часть ОК представляет общие концепции, используемые во всех частях ОК, включая контекст и ограничения, в котором должны быть использованы концепции и подход ОК для применения концепций. Часть 2 и Часть 3 расширяют использование этих концепций и предполагают, что используется описанный подход. Эта глава предполагает знание теории безопасности ПИТ и может использоваться как учебник в этой области.

ОК описывают безопасность, используя множество концепций безопасности и терминологию безопасности. Понимание этих концепций и терминологии является обязательным для эффективного использования ОК. Однако концепции сами по себе являются весьма общими и не предназначены для ограничения класса проблем безопасности ПИТ, для которых применимы ОК.

9.5.8. Контекст безопасности

9.5.8.1. Общий контекст безопасности

Безопасность касается защиты имущества (активов) от угроз, где угрозы понимаются как потенциальная возможность нанесения ущерба защищаемому имуществу. Все категории угроз должны рассматриваться, но в области информационной безопасности большее внимание уделяется тем угрозам, которые связаны с преднамеренной деятельностью людей. На ***рис.1*** представлены понятия безопасности и отношения между ними.

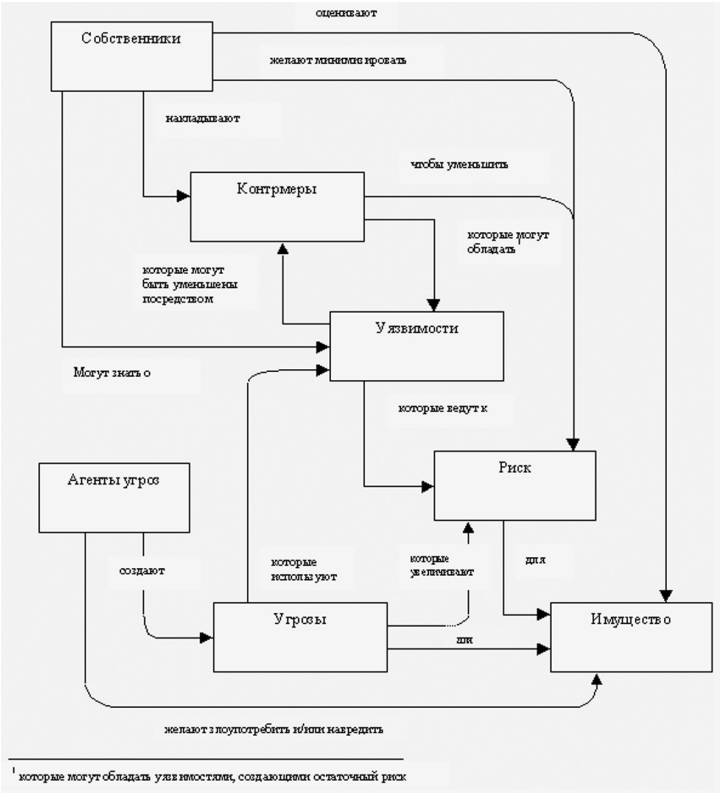


Рис. 1. Взаимосвязь категорий, относящихся к безопасности

Защита имущества является областью ответственности собственников, знающих стоимость этого имущества. Действительные или предполагаемые агенты угроз могут иметь свои оценки стоимости этого имущества и стремятся использовать его в интересах, противоположных интересам собственников. Собственники воспринимают такие угрозы как потенциальную возможность нанесения ущерба имуществу, в ре-

зультате которого оценка стоимости имущества собственников может снизиться. Угрозы безопасности имуществу обычно включают: раскрытие содержимого имущества неавторизованным получателем (потеря конфиденциальности), неавторизованную модификацию (потеря целостности), неавторизованное предотвращение доступа к имуществу (потеря доступности).

Собственники имущества должны анализировать возможные угрозы, чтобы определить какие из них применимы в их условиях (условиях собственников). Результаты такого анализа понимаются как риск (степень риска). Этот анализ может помочь в выборе контрмер, для того чтобы противодействовать риску и уменьшить его до приемлемого уровня.

Чтобы уменьшить риск и удовлетворить ПБ собственников имущества, принимаются некоторые меры (меры безопасности). После их принятия могут оставаться дефекты в защите (уязвимости). Такие дефекты могут использоваться агентами угроз, создавая некоторый уровень риска для имущества. Собственники стремятся минимизировать этот риск.

Собственники хотят быть уверенными, что меры безопасности адекватны угрозам имуществу до того, как противник подвергнет их имущество определенным атакам. Сами собственники могут не обладать способностью оценить все аспекты контрмер и поэтому могут потребовать их оценки. Результаты оценки определяют степень доверия к контрмерам, предназначенным снизить риск до допустимого уровня. Гарантии – есть свойство контрмер, дающее основание для уверенности, что они будут работать должным образом. Результаты оценки могут использоваться собственниками для принятия решения о приемлемости риска. ***Рис. 2*** иллюстрирует эти отношения.

Собственники имущества обычно должны нести ответственность за имущество и должны быть в состоянии отстаивать решение о принятии риска подвергнуть свое имущество угрозам. Результаты оценки должны позволять это делать. Таким образом, оценка должна приводить к объективным и воспроизводимым результатам, на которые можно ссылаться, как на доказанные.

9.5.8.2. Контекст безопасности информационной технологии

Как правило, большая часть имущества находится в форме информации, хранимой, обрабатываемой или передаваемой продуктами или системой ПИТ в интересах собственников информации. Собственники информации желают, чтобы их информация строго контролировалась при распространении и модификациях из одной формы представления в другую. Для этого требуется в продукте или системе ПИТ реализовать специальные меры как часть общих мер безопасности, направленных против угроз информации.

Системы ПИТ по экономическим соображениям в максимальной степени используют существующие продукты широкого применения, такие как операционные системы, прикладные компоненты общего назначения, аппаратные платформы. Меры безопасности ПИТ, реализуемые системой, могут использовать функции, заложенные в этих продуктах, и таким образом, зависеть от правильного функционирования функций безопасности продукта ПИТ. Следовательно, продукты ПИТ могут быть объектом оценки как часть оценки безопасности системы ПИТ. В частности, при проведении анализа на недокументированные возможности в ПО ПИТ.

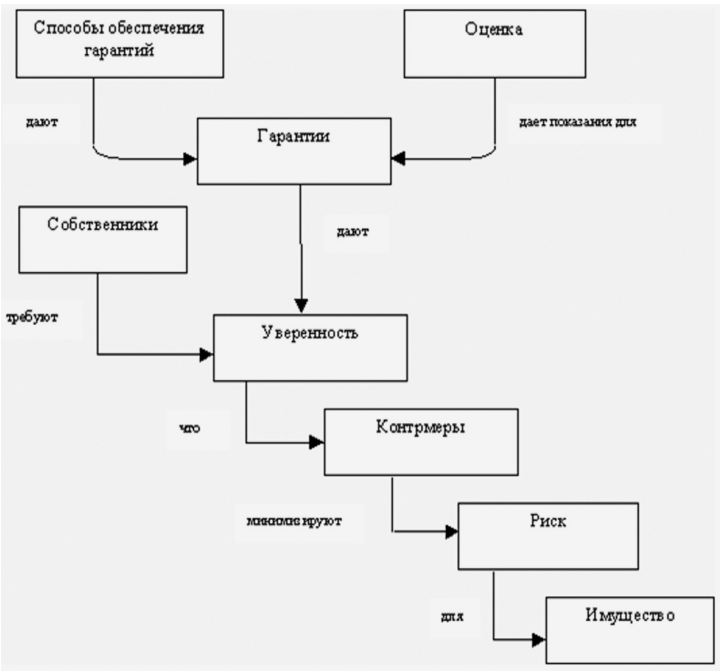


Рис. 2. Отношения оценки

С экономической точки зрения выгодно заранее провести оценку продуктов, независимо от систем, и составить каталог оцененных продуктов. Результаты оценки должны быть выражены в такой форме, которая позволяет использовать продукты каталога во многих системах ПИТ без повторения работы, проделанной перед включением продукта в каталог.

Собственник информации доверяет аккредитацию системы ПИТ специализированной организации, которая определяет адекватность контрмер безопасности для защиты информации и принимает решение о запуске системы в эксплуатацию. Специализированная организация может назначить дополнительную проверку адекватности мер безопасности ПИТ и правильности реализации специальных контрмер в системе ПИТ. Эта оценка может принимать различные формы и степени строгости в зависимости от заданных заказчиком правил или правил специализированной организации.

9.5.9. Подход Общих Критериев

Уверенность в безопасности ПИТ может быть получена только после выполнения действий, предпринимаемых в процессе разработки, оценки и эксплуатации.

9.5.9.1. Разработка

ОК не определяют какую-либо специальную методологию разработки или модель жизненного цикла. ***Рис. 3*** показывает основные взаимоотношения между требованиями безопасности и ООц. Рисунок используется как исходные положения для обсуждения и не должен толковаться как поддержка предпочтения для одной методологии (например «водопад» – нисходящего проектирования) над другой (например, использованием прототипов). Если в начале процесса разработки не установлены необходимые требования, то конечный продукт, хотя и хорошо выполненный с инженерной точки зрения, может не отвечать целям его потребителей.

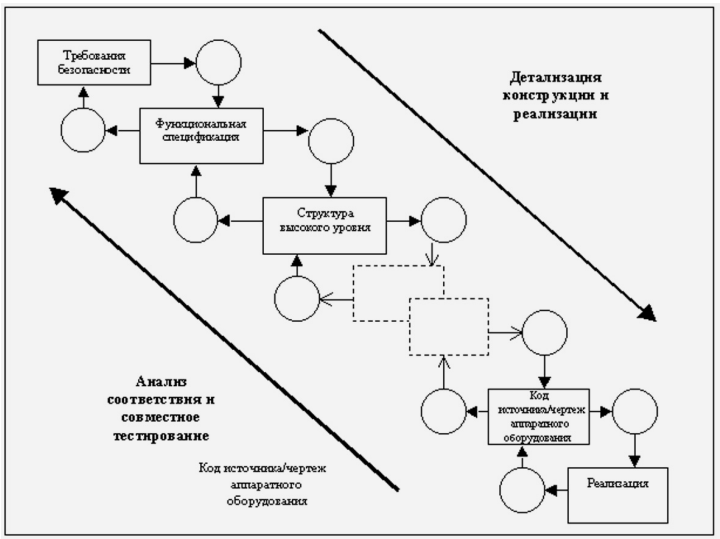


Рис. 3. Модель построения объекта оценки (ООц)

Процесс разработки основан на переработке требований безопасности во множество спецификаций ООц, выраженных в Задании по Безопасности (ЗБ). Каждый более низкий уровень представляет декомпозицию структуры с дополнительными деталями структуры. Последний уровень представления есть сама реализация ООц.

ОК не определяет специального множества представлений структуры. Требование ОК состоит в том, чтобы было достаточно представлений структуры на достаточном уровне подробности, чтобы при необходимости можно было показать выполнение следующих условий:

a) каждый уровень уточнения полностью представляет высшие уровни (т.е., все функции безопасности ООц, свойства и поведение, определенные на высших уровнях абстракции, должны явно присутствовать в низшем уровне);

b) каждый уровень уточнения есть точное представление высших уровней (т.е., нет функций безопасности ООц, свойств или поведений,

определенных на низших уровнях абстракции, которые не требуются на высшем уровне).

В ОК требования гарантии определяют 4 уровня: абстрактной структуры функциональной спецификации, структуры высокого уровня, структуры низкого уровня и реализации. В зависимости от уровня гарантии разработчики должны показать, как предложенная методология разработки соответствует требованиям гарантий ОК.

9.5.9.2. Оценка ООц

Процесс оценки ООц, как это показано на ***рис. 4***, может вестись параллельно с разработкой или следовать за ней. Для оценки требуется (минимально):

a) описание применения ООц, которое включает оцениваемое ЗБ как основу для оценки ООц;

b) ООц, для которого требуется оценка;

c) критерии оценки, методология и схема оценки (обычно реализуется в виде программы и методики испытаний).

Ожидаемый результат процесса оценки есть документальное подтверждение, что ООц удовлетворяет требованиям его безопасности, как это заявлено в ЗБ. Эти документы будут полезны для настоящих и будущих потребителей ООц, а также для разработчиков.

Степень уверенности (в безопасности), полученная благодаря оценке, зависит от требований по гарантиям (например, от уровня гарантии оценки).

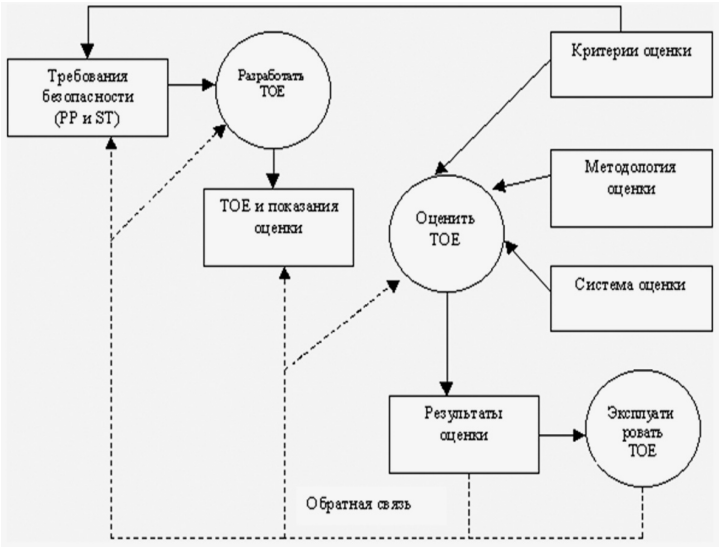


Рис. 4. Процесс оценки ООц

Оценка предназначена для определения ошибок или уязвимостей в ООц, которые разработчик может исправить, таким образом, уменьшая

вероятность нарушений безопасности в последующей работе. Кроме того, для получения высокой оценки разработчик может проявить больше внимания при конструировании и разработке ООц. Следовательно, процесс оценки может косвенно оказать положительное воздействие на начальные требования, процесс разработки, конечный продукт и операционную среду.

9.5.9.3. Эксплуатация

Потребитель может выбрать оцененный ООц для своей среды приложений. В эксплуатации неизвестные ранее ошибки и недостатки могут выявиться или может оказаться, что среда использования не является подходящей для продукта. В результате эксплуатации от разработчика может потребоваться корректировка ООц или переопределение требований безопасности или допущений относительно среды безопасности. Такие изменения могут потребовать переоценки ООц или повышения безопасности среды. В некоторых случаях это может потребовать оценки только необходимых обновлений, чтобы снова получить уверенность в ООц. Хотя ОК содержат критерии, охватывающие поддержку гарантий, но детальные процедуры по переоценке, включая повторное использование результатов оценки, не рассматриваются в ОК.

9.5.10. Концепции безопасности

Критерии оценки наиболее полезны в контексте процессов проектирования и структур отладки, которые являются поддерживающими при разработке и оценке безопасности ООц. Этот раздел необходим только для целей иллюстрации и руководства и не предназначен ограничивать процессы анализа, подходы к разработке или системы оценки, внутри которых могут быть использованы ОК.

ОК применимы, когда используется ПИТ и когда это использование обеспечивает возможность элемента ПИТ защищать имущество. Чтобы показать, что информационные ресурсы находятся в безопасности, вопросы безопасности должны рассматриваться на всех уровнях от наиболее абстрактного до уровней реализации. Эти уровни представления, как описано в последующих подразделах, позволяют характеризовать и обсуждать проблемы и результаты оценки безопасности, но сами по себе не гарантируют, что окончательная реализация ПИТ действительно обладает требуемым уровнем безопасности и, следовательно, обеспечивает доверие.

ОК требуют, чтобы определенные уровни представления содержали разумное объяснение для представления ООц на этом уровне. То есть, такой уровень должен содержать обдуманные и убедительные аргументы, которые показывают, что он находится в согласии с более высоким уровнем, и является самозавершенным, правильным и внутренне согласованным. Аргументы, демонстрирующие согласие с примыкающим более высоким уровнем представления, способствуют правильности реализации ООц. Аргументы, прямо демонстрирующие согласованность с целями безопасности, поддерживают уверенность в том, что ООц эффективно противостоит угрозам и реализует организационную политику безопасности.

ОК определяет различные уровни представления, как показано на ***рис.5***, который иллюстрирует средства, посредством которых требова-

ния и спецификации безопасности могут быть получены при разработке ПЗ или ЗБ. Все требования безопасности ООц, в конечном счете, возникают из рассмотрения целей и контекста ООц. Эта глава не предназначена ограничивать средства, которыми разрабатываются ПЗ или ЗБ, но иллюстрирует, как результаты некоторых аналитических подходов относятся к содержанию ПЗ и ЗБ.

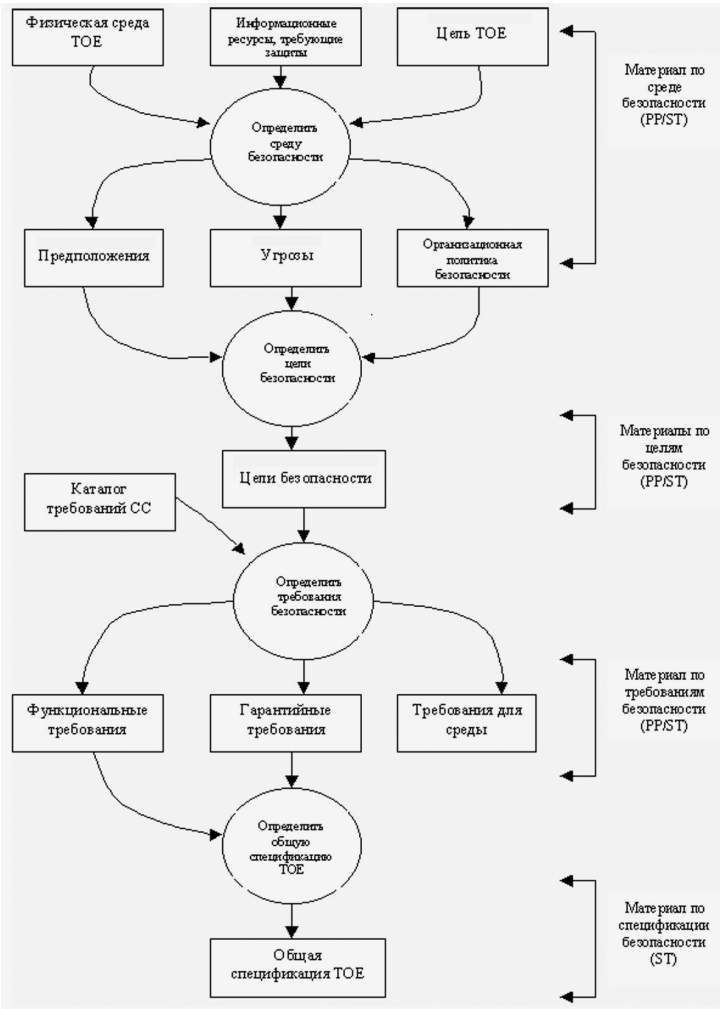


Рис. 5. Вывод требований и спецификаций

9.5.10.1. Среда безопасности

Среда безопасности включает все законодательные акты, политики безопасности, опыт и знания, которые определены в качестве действующих. Таким образом, определен контекст, в котором предназначено использовать ООц. Среда безопасности также включает угрозы безопасности, которые существуют или могут присутствовать в среде.

Чтобы определить среду безопасности, разработчик ПЗ или ЗБ должен принимать в расчет следующее:

a) физическую среду ООц, которая определяет все аспекты операционной среды ООц, относящиеся к безопасности ООц, включая известные физические и персональные меры безопасности;

b) информационные ресурсы, требующие защиты элементом ООц, к которому должны применяться требования или политики безопасности; это ресурсы, требующие прямой защиты (файлы, базы данных) или косвенной (сертификаты авторизации и сама реализация ПИТ);

c) цель ООц, которая должна определять тип продукта и условия использования ООц.

Исследование политик безопасности, угроз и риска должно позволить сформулировать относительно ООц следующие утверждения, касающиеся безопасности:

a) Утверждения о **предположениях**, которые должны быть удовлетворены средой ООц, чтобы ООц мог считать безопасным. Эти утверждения могут быть приняты как аксиомы для оценки ООц.

b) Утверждения об **угрозах** безопасности информационных ресурсов должны определять все угрозы, установленные анализом безопасности, как относящиеся к ООц. ОК характеризуют угрозу в терминах агента угрозы, предполагаемого способа атаки, любых недостатков (уязвимостей), которые являются основой для атаки, и в терминах информационных ресурсов, которые подвергаются атаке. Утверждение о риске безопасности должно квалифицировать каждую угрозу с помощью оценки правдоподобия развития такой угрозы в действительную атаку, оценки правдоподобия успеха такой атаки и последствия любого ущерба, который может быть в результате этого.

c) Утверждения о применяемых **политиках безопасности** должны определять соответствующие политики и правила. Для системы ПИТ такие политики могут быть явно обозначены, тогда как для продукта ПИТ общего применения или класса продуктов может оказаться необходимым сделать рабочие предположения о политике безопасности.

9.5.10.2. Цели безопасности

Результаты анализа среды безопасности должны затем использоваться для установления целей безопасности, которые должны противодействовать определенным угрозам и соответствовать определенным организационным политикам безопасности и предположениям. Цели безопасности должны быть согласованы с установленным операционным намерением или назначением продукта(ов) ООц и любыми знаниями относительно его физической среды.

Определение целей безопасности состоит в том, чтобы рассмотреть все проблемы безопасности и определить, какие аспекты безопасности должны адресоваться либо непосредственно ООц, либо его среде. Эта

категоризация основана на процессе, включающем инженерное решение, политику безопасности, экономические факторы и принятые решения о риске.

Цели безопасности для среды могут быть реализованы как средствами ПИТ, так и нетехническими или процедурными (организационно-техничесими) средствами.

Требования безопасности ПИТ адресуются только к целям безопасности для ООц и его среды ПИТ.

9.5.10.3. Требования безопасности ПИТ

Требования безопасности ПИТ являются уточнением (детализацией) целей безопасности в виде множества требований безопасности для ООц и требований безопасности для среды, которые, если будут удовлетворены, будут гарантировать, что ООц может отвечать своим целям безопасности.

ОК представляют требования безопасности под определенные категории функциональных требований и требований гарантии.

Функциональные требования накладываются на те функции ООц, которые существуют специально для поддержки безопасности ПИТ и определяют желаемое поведение в части безопасности. Часть 2 определяет функциональные требования ОК. Примеры функциональных требований включают требования по идентификации, аутентификации, аудита безопасности и безотказности источника информации.

Когда ООц содержит функции безопасности, которые реализуются вероятностными механизмами (например, пароль или хэш- функция), то требования по гарантиям могут определять тот минимальный уровень стойкости, согласованный с целями безопасности, который должен быть заявлен. В этом случае уровень должен быть одним из следующих: номинальная, средняя или высокая степень защиты (SOF). Каждая такая функция будет нужна, чтобы удовлетворить этот минимальный уровень или, по меньшей мере, опционально определенный специальный показатель.

Уровень гарантий может меняться для данного множества функциональных требований; этот уровень гарантий обычно выражается в терминах возрастающих уровней строгости, создаваемой компонентами гарантий. Часть 3 определяет гарантийные требования ОК и масштаб уровней гарантии оценки (EAL), сконструированные с использованием этих компонентов. Требования по гарантиям основаны на действиях разработчика, на полученных показаниях и на действиях оценивающего эксперта. Примеры требований по гарантиям включают ограничения на строгость процесса разработки и требования к поиску и анализу воздействия потенциальных уязвимостей безопасности.

Гарантии того, что цели безопасности достигаются выбранными функциями безопасности, складываются из следующих двух факторов:

a) уверенности в точности реализации функций безопасности, т.е., оценки того, правильно ли они реализованы;

b) уверенности в эффективности функций безопасности, т.е., оценки того, действительно ли они удовлетворяют заявленным целям безопасности.

Требования безопасности обычно включают как требования желаемого поведения, так и требования по отсутствию нежелаемого пове-

дения. Обычно можно продемонстрировать присутствие желаемого поведения функции безопасности путем ее использования или тестирования. Не всегда, однако, возможно выполнить убедительную демонстрацию отсутствия нежелаемого поведения. Тестирование, просмотр разработки и просмотр реализации функции безопасности дают значительный вклад в уменьшение риска того, что такое нежелательное поведение присутствует. Логически обоснованные утверждения обеспечивают дальнейшую поддержку заявлению, что такое нежелательное поведение отсутствует.

9.5.10.4. Общая спецификация ООц

Общая спецификация ООц, описываемая в ЗБ, определяет выбор требований безопасности для ООц. Она обеспечивает общие определения функций безопасности, необходимых, для удовлетворения функциональных требований, и меры по гарантиям, необходимые для удовлетворения требований по гарантиям.

9.5.10.5. Реализация ООц

Реализация ООц основывается на требованиях его функциональной безопасности и на общей спецификации ООц, содержащихся в ЗБ. Реализация ООц выполняется, используя процесс применения мастерства, опыта и знаний проектирования безопасности ПИТ. ООц будет удовлетворять целям безопасности, если он эффективно выполняет все требования безопасности, содержащиеся в ЗБ.

9.5.11. Описательный материал ОК

ОК представляют структуру, в которой может производиться оценка безопасности. Путем представления требований для доказательства (показания) и анализа могут быть достигнуты более объективные и, следовательно, более полезные результаты оценки. ОК объединяет общее множество конструкций и язык, в котором надо выразить и связать соответствующие аспекты безопасности ПИТ, и позволяет тем, кто отвечает за безопасность ПИТ, получить пользу от прежнего опыта и опыта других.

9.5.11.1. Выражение требований безопасности

ОК определяет множество конструкций, комбинируемых в типовые комплекты требований безопасности, которые могут быть использованы при установлении требований безопасности для будущих продуктов и систем.

Организация требований безопасности в иерархию класс-семейство-компонент предназначена для того, чтобы помочь потребителям определить конкретные требования безопасности.

ОК представляют требования для функциональных аспектов и аспектов гарантий в одном и том же общем стиле и используют для каждого аспекта ту же самую организацию и терминологию.

9.5.11.1.1. Класс

Термин «класс» используется для наиболее общей группы требований безопасности. Все члены класса имеют общее множество целей

(имеют общее назначение), в то же время, различаясь в охвате целей безопасности.

Члены класса называются семействами.

9.5.11.1.2. Семейство

Семейство является группой множеств требований безопасности, которые имеют общие цели безопасности, но могут отличаться в акценте или строгости.

Члены семейства называются компонентами.

9.5.11.1.3. Компонент

Компонент описывает специальное множество требований безопасности и является наименьшим выделяемым множеством требований безопасности для включения в структуры, определенные в ОК. Множество компонентов внутри семейства может быть упорядочено. Они также могут быть частично упорядочены, чтобы представить связанные неиерархические множества. В некоторых случаях в семействе имеется только один компонент, так что упорядочение не применяется.

Компоненты строятся из отдельных элементов. Элемент является самым низким уровнем выражения требований безопасности и есть неделимое требование безопасности, которое может быть проверено оценкой.

Зависимости между компонентами

Между компонентами могут существовать зависимости. Зависимости возникают, когда компонент не самодостаточен и зависит от присутствия другого компонента. Зависимости могут существовать между функциональными компонентами, между компонентами гарантий и между функциональными компонентами и компонентами гарантий.

Описания зависимости компонентов являются частью определений компонентов ОК. Чтобы гарантировать полноту требований ООц, зависимости должны быть удовлетворены при включении компонентов в ПЗ и ЗБ там, где это необходимо.

Операции, разрешенные на компонентах

Компоненты ОК могут использоваться точно, как это определено в ОК, или они могут быть преобразованы (приспособлены) через использование разрешенных операций, чтобы удовлетворить определенную политику безопасности или противодействовать определенной угрозе. Каждый компонент ОК определяет разрешенные операции итерации, уточнения, назначения и выбора, а также обстоятельства, при которых эти операции могут применяться к компоненту, и результаты применения операций. Операции итерации и уточнения могут быть выполнены для любого компонента. Эти четыре операции описаны в следующем виде:

a) **итерация**, которая позволяет использовать компонент более одного раза различными операциями;

b) **назначение**, которое позволяет задавать значение параметра, при котором компонент используется;

c) **выбор**, который позволяет выбирать элементы спецификации из перечня, данного в компоненте;

d) **уточнение**, которое позволяет при использовании компонента добавлять детали.

Некоторые требуемые операции могут быть выполнены (в целом или частично) в ПЗ или могут быть оставлены для выполнения в ЗБ. Тем не менее, все операции должны быть завершены в ЗБ.

9.5.11.2. Использование требований безопасности

ОК определяют три типа конструкций требований: пакет, профиль (ПЗ) и задание (ЗБ). Далее ОК определяют множество критериев безопасности ПИТ, которые учитывают интересы многих сообществ (лиц) и, таким образом, служат главным экспертным материалом при построении этих конструкций. ОК были разработаны на основе использования, где возможно, компонентов требований безопасности, определенных в ОК, которые представляют хорошо известную и понятную область. ***Рис.6*** показывает отношения между этими различными конструкциями.

9.5.11.2.1. Пакет

Промежуточная (вспомогательная) комбинация компонентов называется пакетом. Пакет позволяет выразить множество функциональных требований или требований по гарантиям, которые отвечают определенному подмножеству целей безопасности. Пакет предназначен для многократного использования и определяет требования, которые считаются полезными и эффективными при удовлетворении определенных целей. Пакет может быть использован при построении более объемных пакетов, ПЗ и ЗБ.

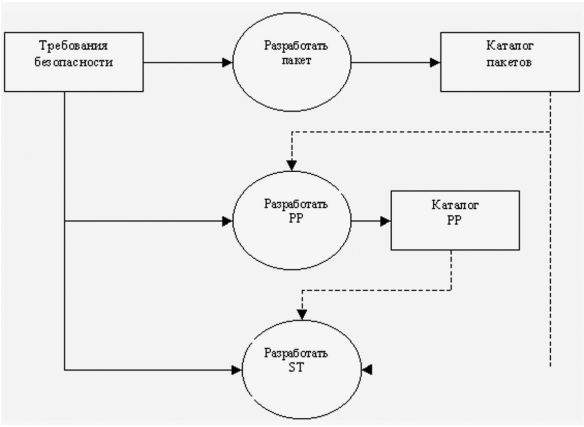


Рис. 6. Использование требований безопасности

Уровни гарантии оценки (EAL) предопределяются пакетами по гарантиям, содержащимися в Части 3. Уровень гарантий является основным требованием по гарантиям для оценки. Каждый EAL определяет точное множество требований по гарантиям. Вместе Уровни гарантий образуют упорядоченное множество, которое предопределяет общую шкалу гарантий Общих критериев.

9.5.11.2.2. Профиль Защиты (PP)

ПЗ содержит множество требований безопасности либо из ОК, либо явно сформулированных, которые включают Уровень гарантий (возможно, повышенный дополнительными компонентами по гарантиям). ПЗ позволяет выразить требования безопасности независимо от реализации для множества ООц, полностью согласованных со множеством целей безопасности. ПЗ предназначен для многократного использования и для определения требований ООц, которые считаются полезными и эффективными для удовлетворения определенных целей, как для функций, так и для гарантий. ПЗ также содержит обоснование для целей безопасности и требований безопасности.

ПЗ может быть разработан сообществами пользователей, разработчиками продуктов ПИТ или другими сторонами, заинтересованными в определении такого общего множества требований. ПЗ дает потребителям средства для выбора определенного множества потребностей в безопасности и облегчает будущую оценку удовлетворения этих потребностей.

9.5.11.2.3. Задание по Безопасности (ST)

ЗБ содержит множество требований безопасности в форме ссылок к ПЗ или прямых ссылок к функциональным компонентам и компонентам по гарантиям ОК, или сформулированных в явной форме. ЗБ позволяет выразить требования безопасности для определенного ООц, которые, как показано оценкой, полезны и эффективны при удовлетворении определенных целей.

ЗБ содержит общую спецификацию ООц вместе с требованиями безопасности, целями и обоснованием выбранных целей и требований. ЗБ является основой для соглашения между всеми сторонами относительно уровня безопасности, который предполагается в ООц.

9.5.11.3. Источники требований безопасности

Требования безопасности ООц могут быть построены с использованием следующих источников:

a) Существующие Профили Защиты (PP).

Требования безопасности ООц в ЗБ могут быть адекватно выражены или согласованы с ранее существующими требованиями, содержащимися в существующем ПЗ.

Существующие ПЗ могут использоваться как основа для новых ПЗ.

b) Существующие пакеты.

Часть требований безопасности ООц в ПЗ или ЗБ могут быть уже выражены в пакете, который может использоваться.

Множества определенных пакетов гарантийных требований составляют Уровни гарантий, определенные в Части 3. Требования по гарантиям ООц в ПЗ или ЗБ должны включать EAL из Части 3.

c) Существующие функциональные или гарантийные компоненты требований.

Функциональные и гарантийные требования в ПЗ или ЗБ могут быть выражены прямо, используя компоненты в Части 2 или 3.

d) Расширенные требования.

Дополнительные функциональные требования, не содержащиеся в Части 2 и/или дополнительные требования по гарантиям, не содержащиеся в Части 3, могут быть использованы в ПЗ или ЗБ.

9.5.12. Типы оценок

9.5.12.1. Оценка ПЗ

Оценка Профиля Защиты (PP) проводится по критериям оценки для ПЗ, содержащимся в Части 3. Целью этой оценки является подтверждение того, что ПЗ полон, состоятелен, технически правилен и подходит для использования как перечень требований для ООц, подлежащего оценке.

9.5.12.2. Оценка Задания по безопасности

Оценка Задания по Безопасности (ST) для ООц проводится по критериям оценки для ЗБ, также содержащимся в Части 3. Цель такой оценки является двойной: во-первых, продемонстрировать, что ЗБ является полным, состоятельным, технически правильным и, следовательно, пригодным для использования в качестве базиса для оценки соответствующего ООц; во-вторых, в случае, когда ЗБ заявляет о соответствии некоторому ПЗ, продемонстрировать, что ЗБ удовлетворяет требованиям этого ПЗ.

9.5.12.3. Оценка ООц

Оценка ООц проводится по критериям оценки, содержащимся в Части 3, принимая за базис оценки, полученные для ЗБ. Целью этой оценки является подтверждение того, что ООц удовлетворяет требованиям безопасности, содержащимся в ЗБ.

9.5.13. Требования Общих Критериев и результаты оценки

9.5.13.1. Введение

Данная часть ОК представляет ожидаемые результаты оценки ПЗ и ООц. Оценки ПЗ и ООц ведут соответственно к каталогам оцененных ПЗ или ООц. Оценка ЗБ ведет к промежуточным результатам, которые используются в рамках оценки ООц.

Оценка должна давать объективные и воспроизводимые результаты, на которые можно ссылаться как на показание, даже если нет полностью объективной шкалы для представления результатов оценки безопасности ПИТ. Наличие множества критериев оценки является необходимым предварительным условием для того, чтобы оценка привела к значимому результату, и обеспечивает технический базис для взаимного признания результатов оценки между организациями, занимающимися оценкой. Но применение критериев содержит как объективные, так и субъективные элементы, что говорит о том, что точные и всеобщие оценки безопасности ПИТ невозможны.

Оценка, сделанная относительно ОК, представляет собой выводы специального типа исследования свойств безопасности ООц. Такая оценка не гарантирует пригодность Ооц для использования в любой среде.

9.5.13.2. Требования в ПЗ и ЗБ

ОК определяют множество критериев безопасности ПИТ, которые учитывают потребности пользователей. Общие критерии были разработаны с учетом понимания того, что для объективной оценки Ооц необходимо использование функциональных компонентов безопасности, содержащихся в Части 2, и компонентов гарантий и EAL, содержащихся в Части 3.

ОК признают возможность того, что могут потребоваться дополнительные функциональные и гарантийные требования, не включенные в каталоги, чтобы представить полное множество требований безопасности ПИТ. Следующие условия должны выполняться при включении этих дополнительных функциональных и гарантийных требований:

a) Любые дополнительные функциональные и гарантийные требования, включенные в ПЗ или ЗБ, должны быть явно и однозначно выражены.

b) Результаты оценки, которые получены при использовании дополнительных функциональных и гарантийных требований, должны учитываться отдельно.

c) Включение дополнительных функциональных и гарантийных требований в ПЗ или ЗБ должно удовлетворять классам APE или ASE Части 3, соответственно.

9.5.13.2.1. Результаты оценки ПЗ

ОК содержат критерии оценки, которые позволяют оценивающему получить подтверждение того, что ПЗ полон, состоятелен, технически правилен и, следовательно, подходит для использования как перечень требований для оцениваемого ООц.

В результате оценки получается утверждение типа «да/нет». ПЗ, для которого оценка получает одобрение («да»), является годным для включения в каталог ПЗ.

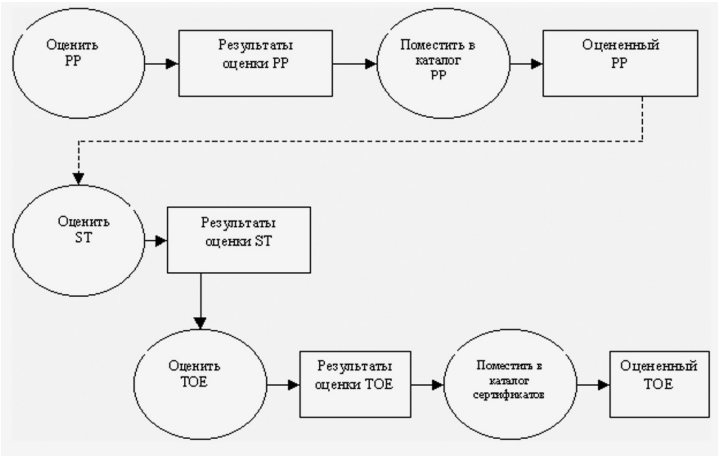


Рис. 7. Результаты оценки

9.5.13.3. Требования в ООц

ОК содержат критерии оценки, которые позволяют оценивающему получить подтверждение того, что ООц удовлетворяет требованиям, выраженным в ЗБ. Используя ОК при оценке ООц, оценивающий будет способен делать утверждения относительно следующих положений:

a) удовлетворяют ли определенные функции безопасности ООц функциональным требованиям и эффективны ли они для достижения целей безопасности ООц;

b) правильно ли реализованы определенные функции безопасности ООц.

Требования безопасности, выраженные в ОК, определяют область применимости критериев оценки безопасности ПИТ. Только такой ООц, для которого требования безопасности выражены в терминах функциональных и гарантийных требований, извлеченных из ОК, может быть оценен по ОК. Использование гарантий, которые не содержатся в EAL, должно быть обосновано.

Однако, может потребоваться, чтобы ООц удовлетворял дополнительным требованиям безопасности, прямо не выраженным в ОК. Поскольку дополнительные требования лежат вне известной области применимости ОК, то результаты такой оценки должны рассматриваться отдельно.

Результаты оценки включают утверждение о соответствии ОК. Описание безопасности ООц в терминах ОК позволит сравнивать характеристики безопасности различных ООц.

9.5.13.3.1. Результаты оценки ООц

Результат оценки ООц должен быть утверждением, которое описывает степень, до которой можно доверять ООц в части соответствия требованиям.

Оценка ООц должна выливаться в утверждение типа «да/нет». ООц, для которого оценка получает ответ «да», является годным для включения в каталог сертификатов ООц.

9.5.13.4. Классификация результатов оценки

Положительный результат оценки должен быть утверждением, которое описывает до какой степени можно доверять ООц или ПЗ относительно соответствия требованиям. Результаты должны формулироваться отдельно для Части 2 (функциональные требования), для Части 3 (гарантийные требования) или прямо для Профиля Защиты, как указано ниже.

a) **Соответствие Части 2** – ПЗ или ООц соответствуют Части 2, если функциональные требования основаны только на функциональных компонентах из Части 2.

b) **Соответствие расширенной Части 2** – ПЗ или ООц соответствуют расширенной Части 2, если функциональные требования включают функциональные компоненты не из Части 2.

c) **Соответствие Части 3** – ПЗ или ООц соответствуют Части 3, если гарантийные требования находятся в форме **EAL** или **пакета гарантий** и основаны только на компонентах гарантий из Части 3.

d) **Соответствие повышенной Части 3** – ПЗ или ООц соответствуют повышенной Части 3, если гарантийные требования находятся в форме **EAL** или **пакета гарантий** плюс другие компоненты гарантий из Части 3.

e) **Соответствие расширенной Части 3** – ПЗ или ООц соответствуют расширенной Части 3, если гарантийные требования находятся в форме **EAL** вместе с дополнительными гарантийными требованиями не из Части 3 или в форме **пакета гарантий**, который включает (или полностью составлен из) гарантийные требования не из Части 3.

f) **Соответствие ПЗ** – ООц соответствует ПЗ, если только он (ООц) соответствует всем частям ПЗ.

9.5.13.9.6. Использование результатов оценки ООц

Продукты и системы ПИТ отличаются использованием результатов оценки. На ***рис. 8*** представлен процесс использования результатов оценки.

ООц разрабатывается по требованиям, которые принимают во внимание свойства безопасности любых оцененных продуктов и ПЗ. Последующая оценка ООц завершается множеством результатов оценки, оформляемых в виде документа.

После оценки продукта ПИТ, предназначенного для широкого использования, суммарные результаты оценки могут быть также помещены в каталог оцененных продуктов, чтобы стать доступными более широкому кругу пользователей, заинтересованных в безопасных продуктах ПИТ.

Если ООц содержится или включается в существующую систему ПИТ, то она подвергается оценке аккредитора. Результаты оценки должны быть доступны аккредитору системы. Результаты оценки ОК могут затем рассматриваться аккредитором с точки зрения критериев, принятых в организации аккредитации. Результаты оценки ОК используются в процессе аккредитации, который завершается принятием решения о допустимости риска при функционировании системы.

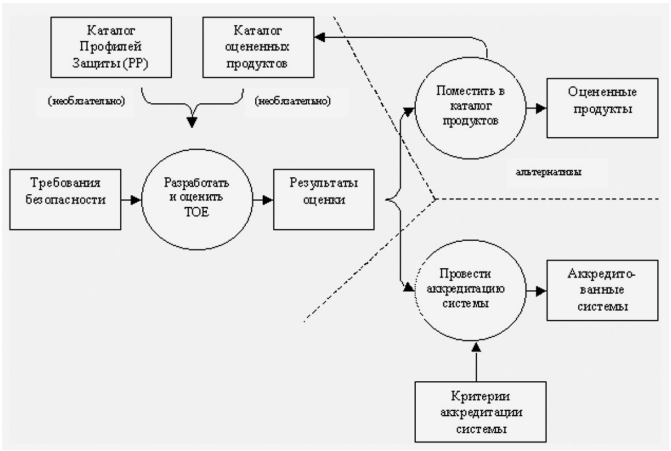


Рис. 8. Использование результатов оценки ООц

9.5.14. Спецификация Профилей Защиты

9.5.14.1. Обзор

ПЗ определяет независимое от реализации множество требований безопасности ПИТ для некоторой категории ООц. Такие ООц предназначаются для удовлетворения общих требований потребителей к безопасности ПИТ. Потребители могут выбирать ПЗ для выражения своих требований к безопасности ПИТ, не ссылаясь на некоторый специальный ООц.

Это приложение содержит требования для ПЗ в описательной форме. Класс гарантий APE, находящийся в главе 3 Части 3, содержит эти требования в форме гарантийных компонентов, которые должны использоваться для оценки ПЗ.

9.5.14.2. Содержание Профиля Защиты

9.5.14.2.1. Содержание и представление

ПЗ должны соответствовать требованиям, описанным в этом приложении. ПЗ должны быть представлены как ориентированный на пользователя документ, который минимизирует ссылки к другим материалам, которые прямо не доступны пользователю. Обоснования могут предоставляться отдельно при необходимости.

Содержание ПЗ представлено на ***рис. 9***, который можно использовать как руководство по структуре документа ПЗ.

9.5.14.2.2. Введение ПЗ

Введение ПЗ должно содержать информацию о содержании и пользовании документом, необходимую для работы с реестром ПЗ, в следующем виде:

a) **Наименование ПЗ** должно обеспечивать указатели и описательную информацию, необходимую для определения идентификатора, каталога, реестра и ссылок ПЗ.

b) **Обзор ПЗ** должен резюмировать ПЗ в повествовательной форме. Обзор должен быть достаточно подробным, чтобы позволить потенциальному пользователю ПЗ определить, представляет ли для него интерес данный ПЗ. Обзор ПЗ может также использоваться как отдельная аннотация для использования в каталогах и реестрах ПЗ.

9.5.14.2.3. Описание ООц

Эта часть ПЗ будет описывать ООц для понимания требований безопасности и содержать тип продукта, применение и общие свойства ПИТ в ООц.

Описание ООц обеспечивает контекст для оценки. Информация, представленная в описании ООц, должна использоваться в процессе оценки для обнаружения несовместимости заданным целям. Поскольку ПЗ обычно не учитывает конкретную реализацию, то описание свойств ООц может рассматриваться как предположения. Если ООц является продуктом или системой, чья главная функция есть безопасность, то эта

часть ПЗ может использоваться для описания более широкого контекста применения, в котором такой ООц будет подходящим.

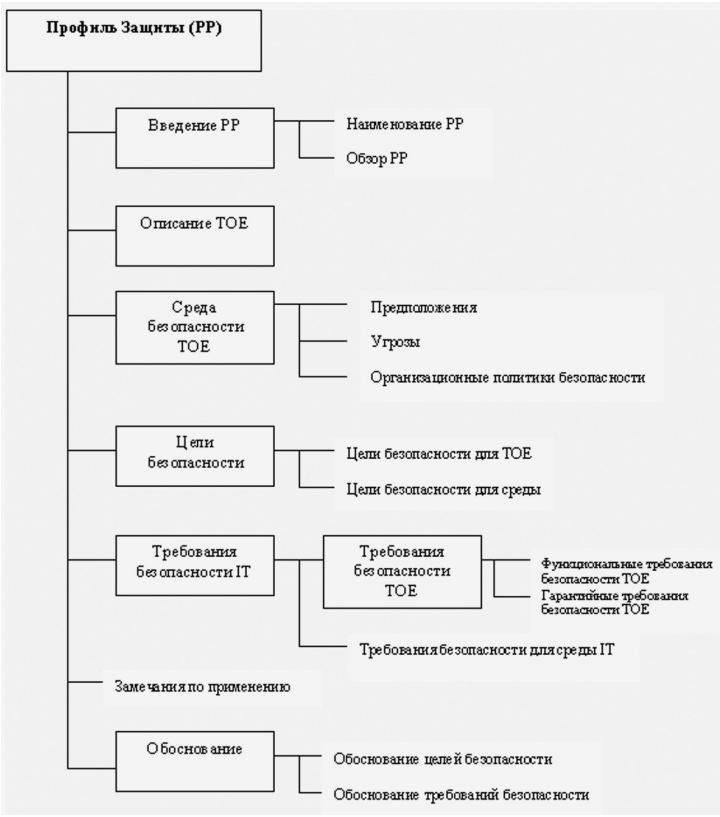


Рис. 9. Содержание Профиля Защиты (PP)

9.5.14.2.4. Среда безопасности ООц

Описание среды безопасности ООц должно содержать аспекты безопасности той среды, в которой предполагается использовать ООц, и способ, каким ожидается использовать ООц. Это описание должно содержать:

а) Описание **предположений** должно содержать аспекты безопасности той среды, в которой предполагается использовать ООц. Оно должно включать следующее:

• информацию о предполагаемом использовании ООц, включая такие аспекты, как предполагаемое использование, потенциальная ценность информационных ресурсов и возможные ограничения в использовании;

• информацию о среде использования ООц, включая физические, персональные и связные аспекты.

b) Описание **угроз** должно включать все угрозы информационным ресурсам, против которых требуется специальная защита внутри ООц или его среды. Заметим, что нет необходимости перечислять все угрозы, которые могут встретиться в среде, а только те, которые реальны для безопасной работы ООц.

Описание угроз должно содержать возможности агента угрозы, атаки и цели атаки. Для агента угрозы должны быть указаны его опыт, используемые ресурсы и мотивация. Описание атаки должно содержать метод атаки, используемые уязвимости и условия.

Если цели безопасности ООц могут быть выведены только из политик безопасности и предположений, то описание угроз может быть опущено.

c) Описание **политик безопасности** должно содержать перечисление и, если необходимо, объяснение всех правил и политик безопасности, которым ООц должен соответствовать. Объяснение и интерпретация необходимы для представления каждой политики в форме, позволяющей точно установить цели безопасности.

Если цели безопасности можно вывести только из угроз и предположений, то описание политик безопасности может быть опущено.

Если ООц физически распределен, то может быть необходимо обсуждать аспекты безопасности среды (предположения, угрозы, организационные политики безопасности) отдельно для каждой области среды ООц.

9.5.14.2.5. Цели безопасности

Описание **целей безопасности** должно определять цели безопасности для ООц и его среды. Цели безопасности должны рассматривать все известные аспекты безопасности среды. Цели безопасности должны отражать описанные намерения и должны быть удобны для противодействия всем определенным угрозам, а также охватывать все определенные организационные политики безопасности и предположения. Должны быть определены следующие категории целей. Заметим: когда угроза или организационная политика безопасности должна охватываться частично ООц, а частично его средой, тогда требуемая цель должна повторяться в каждой категории.

a) **Цели безопасности для ООц** должны быть точно сформулированы и соответствовать определенным угрозам и/или организационным политикам безопасности ООц.

b) **Цели безопасности для среды** должны быть точно сформулированы и соответствовать определенным угрозам, которым ООц не полностью противодействует, и/или организационным политикам безопасности или предположениям, которым не полностью отвечает ООц.

Заметим, что цели безопасности для среды могут быть повторным описанием, в целом или частично, части предположений описания среды безопасности ООц.

9.5.14.2.6. Требования безопасности ПИТ

В данной части ПЗ определяются подробные требования безопасности ПИТ, которые должны быть удовлетворены ООц или его средой. Требования безопасности ПИТ должны быть описаны в следующем виде:

a) Описание **требований безопасности ООц** должно определять функциональные и гарантийные требования безопасности, которым должны удовлетворять ООц и поддерживающее показание (доказательство) для его оценки, чтобы отвечать целям безопасности для ООц. Требования безопасности ООц должны быть описаны в следующем виде:

1) **Функциональные требования безопасности ООц** должны определять функциональные требования для ООц как функциональные компоненты из Части 2, где это возможно.

Там, где необходимо охватить различные аспекты тех же самых требований (например, идентификация более одного типа пользователей), можно повторно использовать (т.е., применять операцию итерации) тот же самый компонент Части 2, чтобы охватить каждый аспект.

Как часть оценки стойкости функций безопасности ООц (AVA.SOF.1), может быть проведена оценка того, удовлетворяет ли ООц требованиям по стойкости, сделанным для отдельных функций безопасности ООц, и общему минимальному уровню стойкости.

2) **Гарантийные требования безопасности ООц** должны определять требования по гарантиям как один из уровней EAL из Части 3. ПЗ может также расширить EAL посредством точного формулирования дополнительных гарантийных требований, не взятых из Части 3.

b) Опциональные **требования безопасности для среды ООц** должны определять требования безопасности ПИТ, которые должны быть удовлетворены средой ПИТ ООц. Если ООц не подтвердил зависимости от среды ПИТ, то эта часть ПЗ может быть опущена.

Заметим, что **требования безопасности для объектов, не являющихся ПИТ**, часто полезные для практики, могут не быть формальной частью ПЗ, так как они не касаются непосредственно реализации ООц.

c) Следующие **общие условия** должны применяться равно к выражению функциональных и гарантийных требований безопасности для ООц и его среды ПИТ:

1) Все требования безопасности ПИТ должны быть сформулированы посредством ссылки к компонентам требований безопасности, взятым из Части 2 или Части 3, где применимо. В случае отсутствия компонентов требований из Части 2 или Части 3, которые легко применимы ко всем или части требований безопасности, ПЗ может сформулировать эти требования определенно без ссылки к ОК.

2) Всякая точная формулировка функциональных и гарантийных требований безопасности ООц должна быть выражена явно и однозначно так, чтобы были выполнимы оценка и подтверждение соответствия. Уровень подробности и способ выражения существующих функциональных и гарантийных требований ОК должен использоваться как образец.

3) При выборе компонентов требований, которые определяют разрешенные операции (назначение или выбор), ПЗ должен использовать эти операции для повышения требований до уровня подробности, необходимого для подтверждения того, что цели безопасности удовлетворены. Любые требуемые операции, которые не выполняются внутри ПЗ, должны быть определены, как таковые.

4) Посредством использования операций на компонентах требований формулировки требований безопасности ООц могут по выбору предписывать или запрещать использование конкретных механизмов безопасности, где необходимо.

5) Все зависимости между требованиями безопасности ПИТ должны отражаться. Зависимости могут быть удовлетворены включением соответствующего требования внутри требований безопасности ООц или как требования на среду.

9.5.14.2.7. Замечания по применению

Эта часть ПЗ может содержать дополнительную поддерживающую информацию, которая имеет отношение или полезна для построения, оценки или использования ООц.

9.5.14.2.8. Обоснование

Эта часть ПЗ представляет доказательство (показание), используемое в оценке ПЗ. Это доказательство поддерживает заявления о том, что ПЗ представляет собой полное и согласованное множество требований, а соответствующий ООц будет обеспечивать эффективное множество контрмер безопасности ПИТ в среде безопасности. Обоснование будет включать следующее:

a) **Обоснование целей безопасности** должно показать, что заявленные цели безопасности прослеживаются по всем аспектам, определенным в среде безопасности ООц, и пригодны для их охвата.

b) **Обоснование требований безопасности** должно показать, что множество требований безопасности (ООц и среды) удобны для удовлетворения целям безопасности и для их отслеживания. Должно быть показано следующее:

1) что комбинация отдельных компонентов функциональных и гарантийных требований для ООц и его среды ПИТ удовлетворяет установленным целям безопасности;

2) что множество требований безопасности образует взаимно поддерживающее и внутренне состоятельное целое;

3) что выбор требований безопасности обоснован. Должно специально обосновываться любое из следующих условий:

- выбор требований, не содержащихся в Части 2 или 3;
- выбор гарантийных требований, не включенных в EAL;
- неудовлетворение зависимостей;

4) что выбранный уровень стойкости функции для ПЗ, вместе с любым точным заявлением стойкости функции, согласованы с целями безопасности для ООц.

Этот громоздкий материал может распространяться отдельно, поскольку он может не представлять интереса для всех пользователей ПЗ.

9.6. Спецификация Задания по Безопасности

9.6.1. Обзор

Задание по Безопасности (ST) содержит требования безопасности ПИТ определенного ООц и определяет функциональные и гарантийные меры безопасности ООц, удовлетворяющие установленным требованиям.

ЗБ для ООц является базисом для согласия между разработчиками, экспертами и потребителями относительно свойств безопасности ООц и области оценки. Аудитория пользователей ЗБ не ограничивается упомянутыми выше, ответственными за создание ООц и его оценку, но может

также включать ответственных за управление, маркетинг, продажу, инсталляцию, конфигурирование, эксплуатацию и использование ООц.

ЗБ может включать требования или заявлять соответствие одному или нескольким ПЗ. Влияние такого заявления о соответствии ПЗ не рассматривается, если сначала определяется требуемое содержание ЗБ в разделе 9.8.2. Подраздел 9.8.2.8 рассматривает влияние заявления о соответствии ПЗ на требуемое содержание ЗБ.

Это приложение содержит требования для ЗБ в описательной форме. Класс гарантий ASE, содержащийся в главе 5 ISO/IEC 15408-3, содержит эти требования в форме компонентов гарантий, которые должны использоваться для оценки ЗБ.

9.6.2. Содержание Задания по Безопасности

9.6.2.1 Содержание и представление

ЗБ должно соответствовать требованиям, описанным в этом приложении. ЗБ должно быть представлено как ориентированный на пользователя документ, который сводит к минимуму ссылки на другой материал, который может отсутствовать у пользователя ЗБ. Обоснование может поставляться отдельно, если это требуется.

Содержание ЗБ изображается на ***рис.10***, который должен быть использован при построении структурного описания ЗБ.

9.6.2.2. Введение ЗБ

Введение ЗБ должно содержать следующие документы по управлению и обзорную информацию.

a) **Наименование ЗБ** должно обеспечивать наименование и описательную информацию, необходимую для контроля и идентификации ЗБ и ООц, к которым она относится.

b) **Обзор ЗБ** должен резюмировать ЗБ в повествовательной форме. Обзор должен быть достаточно детальным, чтобы потенциальный потребитель ООц мог определить, представляет ли он интерес для него. Обзор должен быть отдельным документом, используемым для включения в список оцененных продуктов.

c) Заявление о **соответствии ОК** должно содержать все оцениваемые заявления о соответствии ОК для ООц, как определено в разделе 5.4 этой Части 1.

9.6.2.3. Описание ООц

Эта часть ЗБ должна описывать ООц для понимания его требований безопасности и должна содержать тип продукта или системы. Область и границы ООц должны быть описаны в общих терминах как в физическом плане (компоненты/модули аппаратной части и/или программного обеспечения), так и в логическом плане (ПИТ и свойства безопасности, предлагаемые ООц).

Описание ООц представляет контекст для оценки. Информация, представляемая в описании ООц, может использоваться в процессе оценки для обнаружения несостоятельности. Если ООц есть продукт или система, главная функция которой есть безопасность, то данная часть ЗБ может использоваться для описания контекста более широких приложений такого ООц.

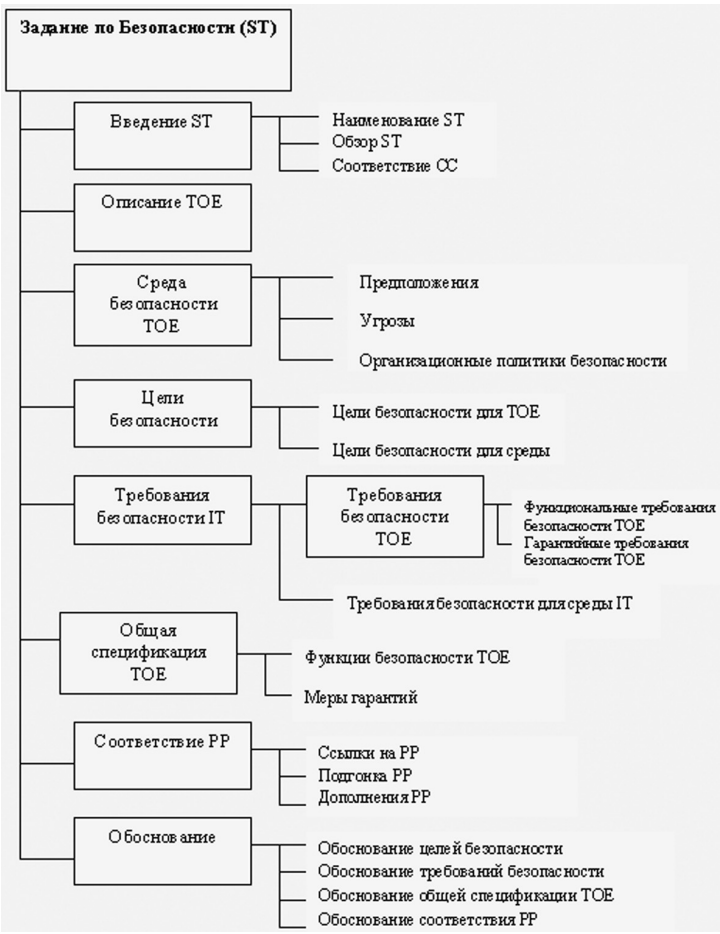


Рис. 10. Содержание Задания по Безопасности (ЗБ)

9.6.2.4. Среда безопасности ООц

Раздел **среды безопасности ООц** должен описывать аспекты безопасности среды, в которой предназначено использовать ООц, и способ, каким, как ожидается, будет использован ООц. Это описание должно включать следующее:

а) Описание **предположений** должно описывать аспекты безопасности среды, в которой ООц будет использоваться или предназначен использоваться. Оно должно включать следующее:

• информацию о намеченном использовании ООц, включая такие аспекты, как намеченное применение, потенциальная стоимость информационных ресурсов и возможные ограничения по использованию;

• информацию о среде использования ООц, включая физические, персональные и связные аспекты.

b) Описание **угроз** должно включать все угрозы информационным ресурсам, против которых требуется специальная защита внутри ООц или его среды. Заметим, что не все возможные угрозы, которые могут встретиться в среде, необходимо перечислять, а только те, которые имеют отношение к безопасной работе ООц.

Каждая угроза должна описываться в терминах агента угрозы, атаки и объекта, который является предметом атаки. Агенты угрозы должны описываться в терминах уровня профессионализма, имеющихся в их распоряжении ресурсов и мотивации. Описание атак включает указание метода атаки, используемые дефекты системы и условия проведения атаки.

Если цели безопасности выводимы только из политик безопасности и предположений, то описание угроз может быть опущено.

c) Описание **политик безопасности** должно определять и, если необходимо, объяснять любые требования и правила организационной политики безопасности, которым должен удовлетворять ООц. Объяснение и интерпретация могут быть необходимы для представления любой (каждой) политики способом, который позволяет точно определить цели безопасности.

Если цели безопасности выводятся только из угроз и предположений, тогда описание политик безопасности может быть опущено.

Если ООц физически распределен, то может быть необходимо обсудить аспекты безопасности среды (предположения, угрозы, организационные политики безопасности) отдельно для различных областей среды ООц.

9.6.2.5. Цели безопасности

Изложение **целей безопасности** должно определить цели безопасности для ООц и его среды. Цели безопасности должны учитывать все известные аспекты среды безопасности. Цели безопасности должны отражать установленное намерение и быть удобными для противодействия всем определенным угрозам и охватывать все определенные организационные политики безопасности и предположения. Должны быть определены следующие категории целей безопасности.

Примечание: когда угроза или организационная политика безопасности должны покрываться частично ООц, и частично его средой, тогда требуемая цель должна повторяться в каждой категории.

a) **Цели безопасности для ООц** должны быть точно сформулированы и сопоставлены с определенными угрозами, которым надо противодействовать, и/или политикам, которые должны быть удовлетворены ООц.

b) **Цели безопасности для среды** должны быть точно сформулированы и сопоставлены с аспектами определенных угроз, которым ООц не полностью противодействует, и/или политик безопасности или предположений, не полностью удовлетворяемых ООц.

Заметим, что цели безопасности для среды можно заново формулировать в целом или частично из предположений при изложении среды безопасности ООц.

9.6.2.6. Требования безопасности ПИТ

Эта часть ЗБ определяет детализированные требования безопасности ПИТ, которые должны быть удовлетворены ООц или его средой. Требования безопасности ПИТ должны определяться следующим образом:

a) Определение **требований безопасности ООц** должно определять функциональные и гарантийные требования безопасности, которые должны удовлетворяться ООц и поддерживающим доказательством (показанием) для его оценки, чтобы соответствовать целям безопасности для ООц. Требования безопасности ООц определяются следующим образом:

1) Определение **функциональных требований безопасности ООц** должно определять функциональные требования для ООц как функциональные компоненты, взятые из Части 2, в зависимости от применения.

Там, где необходимо охватить различные аспекты одного и того же требования (например, идентификация более одного типа пользователя), возможно повторное использование (т.е., применение операции итерации) того же самого компонента Части 2, чтобы охватить каждый аспект.

Как часть оценки стойкости функций безопасности ООц компонент (AVA_SOF.1), может использоваться для оценки стойкости отдельных функций безопасности ООц и общего минимального уровня стойкости, обеспечиваемого ООц.

2) Определение **гарантийных требований безопасности ООц** должно содержать требования гарантий как один из уровней EAL Части 3 ЗБ может расширить Уровни гарантий, точно устанавливая дополнительные требования гарантий, не принадлежащие из Части 3.

b) Опциональное определение **требований безопасности для среды ПИТ** должно определять требования безопасности ПИТ, которые должны быть удовлетворены средой ПИТ ООц. Эта часть ЗБ может быть опущена, если ООц не заявил зависимости от среды ПИТ.

Заметим, что **требования безопасности для среды, не связанной с ПИТ**, часто полезные в практике, не требуются в качестве формальной части ЗБ, так как они не имеют прямого отношения к реализации ООц.

c) Следующие **общие условия** должны равно применяться к выражению функциональных и гарантийных требований безопасности для ООц и среды ПИТ:

1) Все требования безопасности ПИТ должны определяться ссылкой к компонентам требований безопасности, взятым из Части 2 или Части 3, в зависимости от применения. В случае если нет компонентов требований Части 2 или Части 3, точно применимых ко всем или к части требований безопасности, ЗБ может определить точно эти требования без ссылок к ОК.

2) Любое явное определение функциональных и гарантийных требований безопасности ООц должно быть ясно и однозначно выражено так, чтобы были возможны оценка и подтверждение соответствия. Уровень деталировки и способ выражения существующих функциональных и гарантийных требований ОК должен использоваться как модель.

3) Чтобы усилить требования к уровню деталей, необходимых для подтверждения того, что цели безопасности удовлетворены, должны ис-

пользоваться любые требуемые операции.

4) Все зависимости между требованиями безопасности ПИТ должны быть удовлетворены. Зависимости могут быть удовлетворены включением соответствующего требования внутри требований безопасности или как требования на среду.

9.6.2.7. Общая спецификация ООц

Общая спецификация ООц должна определять реализацию требований безопасности для ООц. Эта спецификация должна обеспечивать описание функций безопасности и мер по гарантиям ООц, которые удовлетворяют требованиям безопасности ООц. Заметим, что функциональная информация, обеспечиваемая как часть общей спецификации ООц, может быть идентичной в некоторых случаях информации, обеспечиваемой для ООц как часть требований ADV_FSF.

Общая спецификация ООц содержит следующее:

a) Описание **функций безопасности ООц** должно охватывать функции безопасности ПИТ и должно определять, как эти функции удовлетворяют функциональным требованиям безопасности ООц (соответствуют функциональным требованиям безопасности ООц). Это описание должно включать двунаправленное соответствие между функциями и требованиями, которое ясно показывает, какие требования удовлетворяются какими функциями и что все требования удовлетворяются. Каждая функция безопасности должна удовлетворять, по крайней мере, одно требование безопасности.

1) Функции безопасности ПИТ должны быть определены неформально до уровня деталей, необходимых для понимания их назначения.

2) Все ссылки на механизмы безопасности, включенные в ЗБ, должны связываться с соответствующей функцией безопасности, чтобы было видно, какой механизм используется при реализации каждой функции.

3) Когда в гарантийные требования ООц включается AVA_SOF.1, должны быть определены все функции безопасности ПИТ, которые реализуются вероятностным механизмом (например, пароль или хэш-функция). Вероятность нарушения механизмов таких функций путем преднамеренной или случайной атаки является важной для безопасности ООц. Для всех этих функций должен обеспечиваться анализ стойкости функции безопасности ООц. Стойкость каждой определенной функции должна быть определена и заявлена или как SOF – номинальная, SOF – средняя, SOF – высокая, или как опционально определенное значение по специальной метрике.

b) Описание **мер гарантии** определяет меры гарантий ООц, которые удовлетворяют установленным требованиям гарантий. Меры гарантий должны быть связаны с требованиями гарантий так, чтобы было видно, какие меры удовлетворяют каким требованиям.

При необходимости, определение мер гарантий может быть сделано путем ссылок на соответствующие планы качества, планы жизненного цикла или планы управления.

9.6.2.8. Соответствие ПЗ

ЗБ могут заявлять, что ООц удовлетворяет требованиям одного или, возможно, нескольких ПЗ. Для любых сделанных заявлений о соответс-

твии ПЗ, ЗБ должно включать заявление о **соответствии ПЗ**, которое содержит объяснение, обоснование и другой материал, необходимый для подтверждения этого заявления.

Содержание и представление требований и целей ООц в ЗБ может зависеть от заявлений о **соответствии ПЗ**, сделанных для ООц. Это влияние на ЗБ может быть суммировано рассмотрением следующих случаев для каждого заявленного ПЗ:

a) Если в ЗБ никакого заявления о соответствии некоторому ПЗ не сделано, тогда полное представление требований и целей ООц должно быть сделано так, как описано в этом приложении. Никакие заявления о соответствии ПЗ не включаются.

b) Если ЗБ заявляет только соответствие с требованиями некоторого ПЗ без необходимости дальнейшей квалификации, тогда достаточна ссылка на этот ПЗ, чтобы определить и обосновать цели и требования ООц. В повторном описании содержания ПЗ нет необходимости.

c) Если ЗБ заявляет о соответствии с требованиями некоторого ПЗ и этот ПЗ требует дальнейшей квалификации, тогда ЗБ должно показать, что требования этого ПЗ по квалификации удовлетворены. Такая ситуация обычно возникает там, где ПЗ содержит неполные операции. В такой ситуации ЗБ могут сослаться на специальные требования, но дополнить операции внутри ЗБ. В некоторых случаях, где требования по дополнению операций существенны, может оказаться предпочтительным переписать содержание ПЗ внутри ЗБ для ясности.

d) Если ЗБ заявляет о соответствии с требованиями некоторого ПЗ, но расширяет этот ПЗ добавлением каких-то целей и требований, тогда ЗБ должно определить добавления, где ссылка на ПЗ может быть достаточной для определения целей и требований ПЗ. В некоторых случаях, где добавления существенны, может оказаться предпочтительным переписать содержание ПЗ внутри ЗБ для ясности.

e) Случай, когда ЗБ заявляет о частичном соответствии некоторому ПЗ, не приемлем для оценки по ОК.

ОК не являются предписывающими относительно выбора переписывания или ссылки на цели и требования ПЗ. Фундаментальное требование состоит в том, чтобы содержание ЗБ было полным, ясным и недвусмысленным, позволяющим провести оценку ЗБ, чтобы ЗБ был приемлемым базисом для оценки ООц и чтобы была ясной связь с любым заявленным соответствием ПЗ.

Если сделано какое-либо заявление о соответствии ПЗ, то оно должно содержать следующий материал для каждого заявленного ПЗ.

a) Описание **ссылки на ПЗ** должно определять ПЗ, для которого заявляется соответствие, плюс любое расширение, которое может быть необходимым к этому заявлению. Правильное заявление включает в себя то, что ООц удовлетворяет все требования ПЗ.

b) Описание **подгонки ПЗ** должно определить описания требований безопасности ПИТ, которым удовлетворяют разрешенные операции ПЗ или, в противном случае, дальше квалифицировать требования ПЗ.

c) Описание **дополнений ПЗ** должно содержать описания целей и требований ООц, которые являются дополнительными к целям и требованиям ПЗ.

9.6.2.9. Обоснование

Эта часть ЗБ представляет показания, используемые для его оценки. Эти показания поддерживают заявления о том, что ЗБ является полным и совместимым множеством требований, при выполнении которых соответствующий ООц должен обеспечивать эффективное множество контрмер безопасности ПИТ в данной среде безопасности, а общая спецификация ООц удовлетворяет этим требованиям. Обоснование также показывает, что все заявления о соответствии ПЗ удовлетворяются. Обоснование должно включать следующее:

a) **Обоснование целей безопасности**, показывающее, что установленные цели безопасности связаны со всеми аспектами, определенными в среде безопасности ООц, и пригодны для их охвата.

b) **Обоснование требований безопасности**, показывающее, что множество требований безопасности (для ООц и среды) является достаточным для удовлетворения и охвата целей безопасности. Должно быть показано:

1) что комбинация отдельных компонентов функциональных и гарантийных требований для ООц и его среды ПИТ удовлетворяет установленным целям безопасности;

2) что множество требований безопасности образует взаимно поддерживающее и внутренне состоятельное целое;

3) что выбор требований безопасности обоснован. Любое из следующих условий должно быть специально обосновано:

• выбор требований, не содержащихся в Части 2 или Части 3;

• выбор гарантийных требований, не включенных в EAL;

• отсутствие удовлетворения зависимостей;

4) что выбранный уровень стойкости функции для ЗБ, вместе с любыми точными заявлениями о стойкости функции, согласуется с целями безопасности для ООц.

c) **Обоснование общей спецификации ООц**, показывающее, что функции безопасности ООц и меры по гарантиям достаточны для удовлетворения требований безопасности ООц. Должно быть показано следующее:

1) что комбинация определенных функций безопасности ПИТ ООц работает так, что удовлетворяются функциональные требования безопасности ООц;

2) что сделанные заявления о стойкости функции ООц правильны или что утверждения о том, что такие заявления не нужны, правильны;

3) что заявление обосновано в том, что установленные меры по гарантиям согласуются с требованиями по гарантиям.

Описание обоснования должно быть представлено на уровне деталей, который согласуется с уровнем деталей определения функций безопасности.

d) Описание **обоснования соответствия ПЗ** должно объяснять любую разницу между целями и требованиями безопасности ЗБ и целями и требованиями безопасности любого ПЗ, соответствие которому заявлено. Эта часть ЗБ может быть опущена, если не сделано никаких заявлений о соответствии ПЗ или цели и требования безопасности ЗБ идентичны целям и требованиям безопасности любого заявленного ПЗ.

Этот объемный материал может распределяться по частям, так как он может не соответствовать или не быть полезным для всех пользователей ЗБ.

9.7. Современные нормативы обеспечения информационной безопасности в финансовой сфере

9.7.1. Введение

В данном разделе рассмотрим вторую редакция Стандарта СТО БР ИББС-1.0-2006 (принята и введена в действие Распоряжением Банка России от 26 января 2006 г. № Р-27, текст Стандарта опубликован в «Вестнике Банка России» № 6 от 03.02.06). Положения стандарта, в соответствии с разделом «Область применения», носят рекомендательный характер и применяются кредитными организациями на добровольной основе. Однако по ряду положений стандарта обязательность исполнения установлена действующим законодательством Российской Федерации и нормативными правовыми актами Банка России. В частности:

• статья 857 Гражданского кодекса Российской Федерации и статья 26 Федерального закона «О банках и банковской деятельности» обязывают банки гарантировать тайну банковского счета и банковского вклада, операций по счету и сведений о клиенте;

• статья 139 Гражданского кодекса Российской Федерации вводит понятие служебная и коммерческая тайна, а Федеральный закон «О коммерческой тайне» обязывает владельца информации, установившего в ее отношении режим «коммерческой тайны», принимать меры по охране конфиденциальности такой информации. Принятые меры признаются разумно достаточными, если они исключабт доступ к защищаемой информации любых лиц без согласия владельца (обладателя) и обеспечивают возможность использования и передачи информации без нарушения режима коммерческой тайны. В приложении 1 приведен минимально необходимый перечень мер, в Стандарте предлагается сделать отсылки к тексту статьи или процитировать ее.

• Федеральный закон от 10.01.2002 г. № 1-ФЗ «Об электронной цифровой подписи» регулирует условия использования ЭЦП в электронных документах, при соблюдении которых электронная подпись признается равнозначной собственноручной подписи в документе на бумажном носителе.

• Использование программного обеспечения и баз данных в части обеспечения авторских и смежных прав регламентируется законом от 23.09.1992 г. № 3523-1 «О правовой охране программ для электронных вычислительных машин и баз данных» и законом «Об авторском праве и смежных правах».

• Создание условий предоставления телекоммуникационных услуг банкам и их клиентам обеспечивают нормы Федерального закона «О связи», предусматривающие, в том числе, обязательное лицензирование деятельности, направленной на коммерческое предоставление услуг связи.

• Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и защите информации» регулирует отношения, возникающие при формировании и использовании информационных ресурсов, технологий и средств их обеспечения, а также защите

информации. Последний нормативный документ, в частности, содержит исходные нормы, регулирующие отношения сторон при совместной обработке информации и при ее обработке в порядке предоставления услуг.

Помимо законов федерального уровня действует целый ряд нормативных; документов, выпущенных Банком России. В первую очередь, это документ, регламентирующий правила построения кредитными организациями собственных систем выявления, измерения и определения приемлемого уровня рисков, присущих банковской деятельности, и организации внутреннего контроля, а именно – Положение Банка России № 242-П от 16.12.2003 г. «Об организации внутреннего контроля в кредитных организациях и банковских группах», а также письма Банка России № 70-Т от 23.06.2004 «О типичных банковских рисках», № 76-Т от 24.05.2005 г. «Об организации управления операционным риском в кредитных организациях» и № 92-Т от 30.06.2005 г. «Об организации управления правовым риском и риском потери деловой репутации».

Стандарт информационной безопасности Банка России в целом коррелирован с законодательством Российской Федерации в области информационных технологий и безопасности, банковским и гражданским законодательством, наработками Банка России по управлению банковскими рисками, в первую очередь операционными.

Настоящий стандарт распространяется на организации банковской системы Российской Федерации (далее по тексту – организации БС РФ) и устанавливает положения (политики, требования и т.п.) по обеспечению информационной безопасности в организациях БС РФ.

Настоящий стандарт рекомендован для применения путем включения ссылок на него и(или) прямого использования устанавливаемых в нем положений во внутренних нормативных и методических документах организаций БС РФ, а также в договорах.

Положения настоящего стандарта применяются на добровольной основе, если только в отношении конкретных положений обязательность не установлена действующим законодательством Российской Федерации, нормативным правовым актом Банка России или условиями договора.

Настоящий стандарт может быть введен в действие организаций БС РФ в качестве обязательного к исполнению в случае, если такая необходимость существует.

Документ использует следующие термины и определения:

Банковская система Российской Федерации – Банк России и кредитные организации, а также филиалы и представительства иностранных банков[1].

Активы организации банковской системы Российской Федерации – все, что имеет ценность для организации банковской системы Российской Федерации и находится в ее распоряжении.

К активам организации БС РФ могут относиться:

---

1. Федеральный закон «О банках и банковской деятельности» от 01.12.1990 № 395-1 в редакции ФЗ от 03.02.1996 № 17-ФЗ, от 31.07.1998 № 151-ФЗ, от 05.07.1999 № 126-ФЗ, от 08.07.1999 № 136-ФЗ, от 19.06.2001 № 82-ФЗ, от 07.08.2001 № 121-ФЗ, от 21.03.2002 № 31-ФЗ с изменениями, внесенными постановлением Конституционного Суда Российской Федерации от 23.02.1999 № 4-П.

• банковские ресурсы (финансовые, людские, вычислительные, телекоммуникационные и пр.);

• информационные активы, в т.ч. различные виды банковской информации (платежной, финансово-аналитической, служебной, управляющей и пр.) на следующих фазах их жизненного цикла: генерация (создание), обработка, хранение, передача, уничтожение;

• банковские процессы (банковские платежные технологические процессы, банковские информационные технологические процессы, процессы жизненного цикла автоматизированных банковских систем и др.);

• банковские продукты и услуги, предоставляемые клиентам.

Автоматизированная банковская система – автоматизированная система, реализующая банковский технологический процесс или его часть.

Роль в организации банковской системы Российской Федерации: заранее определенная совокупность правил, устанавливающих допустимое взаимодействие между субъектом и объектом в организации БС РФ.

1. К субъектам относятся лица из числа руководителей организации БС РФ, ее персонала, клиентов или инициируемые от их имени процессы по выполнению действий над объектами.

2. Объектами могут быть аппаратное средство, программное средство, программно-аппаратное средство, информационный ресурс, услуга, процесс, система, над которыми выполняются действия.

Банковский технологический процесс – технологический процесс, содержащий операции по изменению и(или) определению состояния банковской информации, используемой при функционировании или необходимой для реализации банковских услуг.

1. Операции над банковской информацией могут выполняться вручную или быть автоматизированными, например, с помощью комплексов средств автоматизации автоматизированных банковских систем.

2. Операции над банковской информацией требуют указания ролей их участников (исполнителей и лиц, принимающих решения или имеющих полномочия по изменению технологических процессов, в том числе персонала автоматизированных банковских систем).

3. В зависимости от вида деятельности выделяют: банковский информационный технологический процесс, банковский платежный технологический процесс и др.

Информационная безопасность организации банковской системы Российской Федерации – состояние защищенности интересов (целей) организации банковской системы Российской Федерации в условиях угроз в информационной сфере.

1. Защищенность достигается обеспечением совокупности свойств информационной безопасности – конфиденциальностью, целостностью, доступностью информационных активов и инфраструктуры. Приоритетность свойств информационной безопасности определяется значимостью информационных активов для интересов (целей) организации.

2. Информационная сфера представляет собой совокупность информации, информационной инфраструктуры, субъектов, осуществляющих сбор, формирование, распространение, хранение и использование информации, а также системы регулирования возникающих при этом отношений.

Менеджмент – скоординированная деятельность по руководству и управлению.

Термин «management» иногда относится к людям, т.е. к лицу или группе работников, наделенных полномочиями и ответственностью для руководства и управления организацией. Когда «management» используется в этом смысле, его следует всегда применять с определяющими словами с целью избежания путаницы с понятием «management», определенным выше. Например, не одобряется выражение «руководство должно...», в то время как «высшее руководство должно…» – приемлемо.

Система менеджмента информационной безопасности организации банковской системы Российской Федерации; СМИБ: часть общей системы менеджмента организации банковской системы Российской Федерации, основывающаяся на подходе бизнес-риска, предназначенная для создания, реализации, эксплуатации, мониторинга, анализа, поддержки и повышения информационной безопасности организации банковской системы Российской Федерации.

Система менеджмента включает структуру, политики, деятельности по планированию, обязанности, практики, процедуры, процессы и ресурсы организации.

Осознание информационной безопасности: понимание организацией необходимости самостоятельно на основе принятых в ней ценностей и накопленных знаний формировать и учитывать в рамках основной деятельности (бизнеса) прогноз результатов от деятельности по обеспечению информационной безопасности, а также поддерживать эту деятельность адекватно прогнозу.

Осознание информационной безопасности является внутренним побудительным мотивом организации инициировать и поддерживать деятельность по менеджменту информационной безопасности, в отличие от побуждения или принуждения, когда решение об инициировании и поддержке деятельности по менеджменту информационной безопасности определяется соответственно либо возникшими проблемами организации, такими, как инцидент информационной безопасности, либо внешними факторами, например, требованиями законов.

Политика информационной безопасности организации банковской системы Российской Федерации – одно или несколько правил, процедур, практических приемов и руководящих принципов в области информационной безопасности, которыми руководствуется организация банковской системы Российской Федерации в своей деятельности.

Инцидент информационной безопасности организации банковской системы Российской Федерации – событие, вызывающее действительное, предпринимаемое или вероятное нарушение информационной безопасности организации банковской системы Российской Федерации.

Нарушение может вызываться либо ошибкой людей, либо неправильным функционированием технических средств, либо природными факторами (например, пожар или наводнение), либо преднамеренными злоумышленными действиями, приводящими к нарушению конфиденциальности, целостности, доступности, учетности или неотказуемости.

Риск – неопределенность, предполагающая возможность потерь (ущерба).

Менеджмент риска – скоординированные действия по руководству и управлению в отношении риска с целью его минимизации.

Обычно менеджмент риска включает в себя оценку риска, обработку риска, принятие риска.

Риск нарушения информационной безопасности организации банковской системы Российской Федерации: неопределенность, предполагающая возможность ущерба состояния защищенности интересов (целей) организации банковской системы Российской Федерации в условиях угроз в информационной сфере.

Мониторинг информационной безопасности организации банковской системы Российской Федерации (мониторинг ИБ) – постоянное наблюдение за событиями мониторинга ИБ, сбор, анализ и обобщение результатов наблюдения.

Аудит информационной безопасности организации банковской системы Российской Федерации: периодический, независимый и документированный процесс получения свидетельств аудита и объективной их оценки с целью установления степени выполнения в организациях БС РФ установленных требований по обеспечению информационной безопасности.

Внутренние аудиты («аудиты первой стороной») проводятся самой организацией или от ее имени для анализа менеджмента или других внутренних целей и могут служить основанием для самодеклараций организации о соответствии требованиям по ИБ.

Внешние аудиты включают «аудиты второй стороной» и «аудиты третьей стороной». Аудиты второй стороной проводятся сторонами, заинтересованными в деятельности организации, например, потребителями или другими лицами от их имени. Аудиты третьей стороной проводятся внешними независимыми организациями.

Независимость при аудите предполагает полную свободу аудитора (самостоятельность) в отборе и анализе свидетельства аудита (изложение фактов или другой информации, связанной с критериями аудита) в отношении объекта аудита.

Оценка соответствия информационной безопасности организации банковской системы Российской Федерации установленным требованиям: любая деятельность, связанная с прямым или косвенным определением того, что выполняются или не выполняются соответствующие требования информационной безопасности в организации банковской системы Российской Федерации.

9.7.2. Исходная концептуальная схема (парадигма) обеспечения информационной безопасности организаций БС РФ

Любая целенаправленная деятельность (бизнес) порождает риски, сущность которых – естественная неопределенность будущего. Это – объективная реальность, и понизить эти риски можно лишь до уровня неопределенности сущностей, характеризующих природу бизнеса. Оставшаяся часть риска, определяемого факторами среды деятельности организации БС РФ, на которые организация не в силах влиять, должна быть неизбежно принята. При этом степень необходимой защищенности информационной сферы организации определяется анализом и оцен-

кой рисков ИБ, которые должны быть согласованы с рисками основной (бизнес) деятельности организации БС РФ.

Деятельность организации БС РФ осуществляется через реализацию трех групп высокоуровневых процессов: основные процессы (процессы основной деятельности), вспомогательные процессы (процессы по видам обеспечения), процессы менеджмента (управления) организацией. Процессы по обеспечению ИБ организации БС РФ составляют один из видов вспомогательных процессов, реализующих поддержку (обеспечение) процессов основной деятельности организации в целях достижения ею максимально возможного результата.

В основе исходной концептуальной схемы информационной безопасности организаций БС РФ лежит противоборство собственника[2] и злоумышленника[3] за контроль над информационными активами. Однако другие, незлоумышленные, действия также лежат в сфере рассмотрения данного стандарта.

В случае если злоумышленник устанавливает контроль над информационными активами, как самой организации БС РФ, так и клиентам, которые доверили ей свои собственные активы, может быть нанесен ущерб.

Наибольшими возможностями для нанесения ущерба организации БС РФ обладает ее собственный персонал. В этом случае содержанием деятельности злоумышленника является нецелевое использование предоставленного контроля над информационными активами, а также сокрытие следов своей деятельности. Внешний злоумышленник скорее да, чем нет, может иметь сообщника(ов) внутри организации.

Собственник практически никогда не знает о готовящемся нападении, оно всегда бывает неожиданным. Нападения, как правило, носят локальный и конкретный по месту, цели и времени характер.

Злоумышленник изучает объект нападения, как правило, не только теоретически, никак не проявляя себя, но и практически, путем эксперимента, подбора «отмычек» к системе менеджмента ИБ (СМИБ) организации. Таким образом, он отрабатывает наиболее эффективный метод нападения. Поэтому собственник должен постоянно стремиться к выявлению следов такой активности. В том числе и для этой цели собственник создает уполномоченный орган – свою службу ИБ (подразделения (лица) в организации, ответственные за обеспечение ИБ).

Сложно и ресурсоемко, а значит, малоэффективно искать следы такой активности и по факту настраивать СМИБ. Поэтому главный инструмент собственника – основанный на опыте прогноз (составление модели угроз и модели нарушителя)[3], а также работа с персоналом организации по повышению его бдительности в возможных критических

2. Под собственником здесь понимается субъект хозяйственной деятельности, имеющий права владения, распоряжения или пользования активами, который заинтересован или обязан (согласно требованиям законов или иных законодательных или нормативно-правовых актов) обеспечивать защиту активов от угроз, которые могут снизить их ценность или нанести ущерб собственнику.

3. Под злоумышленником здесь понимается лицо, которое совершает или совершило заранее обдуманное действие с осознанием его опасных последствий или не предвидело, но должно было и могло предвидеть возможность наступления этих последствий (адаптировано из ст. 27 УК РФ). Далее по тексту данные лица именуются злоумышленниками (нарушителями).

условиях, готовности и способности к адекватным действиям в условиях потенциальной злоумышленной активности.

Чем точнее сделан прогноз (составлены модель угроз и модель нарушителя), тем ниже риски нарушения ИБ в организации БС РФ при минимальных ресурсных затратах.

Наиболее правильный и эффективный способ добиться минимизации рисков нарушения ИБ для собственника – разработать на основе точного прогноза, базирующегося в том числе и на анализе и оценке рисков ИБ, политику ИБ организации и в соответствии с ней реализовать, эксплуатировать и совершенствовать СМИБ организации.

Политика ИБ организаций БС РФ разрабатывается на основе принципов обеспечения ИБ организаций БС РФ, моделей угроз и нарушителей, идентификации активов, подлежащих защите, оценки рисков с учетом особенностей бизнеса и технологий, а также интересов конкретного собственника.

Собственник должен знать, что он должен защищать. Собственник должен знать и уметь выделять (идентифицировать) наиболее важный для его бизнеса информационный актив (ресурс).

При этом собственник принимает решение относительно принятия конкретного риска или же внедрения мер контроля и процедур по обработке существующих рисков.

При принятии решений о внедрении защитных мер (мер контроля) для противодействия идентифицированным угрозам (рискам) собственник должен учитывать, что тем самым он увеличивает сложность своей системы управления ИБ, а повышение сложности управления ИБ порождает новые уязвимости. Поэтому при выборе решения о внедрении защитных мер для обработки существующих рисков, а не принятия или переноса рисков должны учитываться вопросы эксплуатации защитных мер и их влияния на структуру рисков организации.

Далеко не каждый собственник располагает потенциалом для составления точного прогноза (модели угроз и модели нарушителя). Такой прогноз может и должен составляться с учетом опыта ведущих специалистов банковской системы, а также с учетом международного опыта в этой сфере. Аналогично должны разрабатываться и основные требования ИБ организаций БС РФ.

При разработке моделей угроз и моделей нарушителя необходимо учитывать, что по сложившейся уже практике существующая сложность современных банковских технологий приводит к их меньшей привлекательности для злоумышленника, чем персонал и система управления безопасностью организации. Поэтому все точки в банковских технологических процессах, где осуществляется взаимодействие персонала со средствами и системами автоматизации, должны тщательно контролироваться.

Соблюдение политики ИБ в значительной степени является элементом корпоративной этики, поэтому на уровень ИБ в организации серьезное влияние оказывают отношения, как в коллективе, так и между коллективом и собственником или менеджментом организации, представляющим интересы собственника. Поэтому этими отношениями необходимо управлять. Понимая, что наиболее критичным элементом безопасности организации является ее персонал, собственник должен всемерно поощрять решение проблемы ИБ.

Любые защитные меры в силу ряда объективных причин со временем имеют тенденцию к ослаблению своей эффективности, в результате чего общий уровень ИБ может снижаться. Это неминуемо ведет к возрастанию рисков нарушения ИБ.

Для того чтобы этого не допустить, необходимо определить процессы, обеспечивающие контроль (мониторинг и аудит ИБ организаций БС РФ), а также оценку эффективности СМИБ организаций БС РФ (так называемый «процессный подход»), что должно стать основой для дальнейшего планирования ИБ. Указанные процессы должны реализовываться в рамках циклической модели менеджмента ИБ: «планирование – реализация – проверка – совершенствование – планирование», отвечающей принципам и моделям корпоративного менеджмента в организациях, включая менеджмент в банковском деле.

При этом эффективность и результативность обеспечения ИБ, включая соответствующие процессы ИБ, должны оцениваться с позиции содействия (пользы) в достижении целей деятельности организации.

Обеспечение ИБ организаций БС РФ основывается на «процессном подходе» для установления, реализации, эксплуатации, мониторинга, обслуживания и повышения эффективности СМИБ.

Любое действие, использующее ресурсы и управляемое для обеспечения преобразования неких входных ресурсов, информации и иных сущностей в выходы, определяется как «процесс». Выход одного процесса может быть входом для другого процесса. Представление деятельности по обеспечению ИБ в виде системы процессов в пределах организации вместе с идентификацией, взаимодействиями и их координацией и управлением определяется как «процессный подход».

«Процессный подход» к обеспечению ИБ организаций БС РФ требует, чтобы персонал организации, клиенты, пользователи, контрагенты и иные заинтересованные стороны придавали особое значение:

а) пониманию требований информационной безопасности бизнеса и потребности устанавливать политику и цели для информационной безопасности;

б) реализации и надлежащей эксплуатации средств управления ИБ (защитных мер) в контексте управления общим риском деятельности (бизнеса) организации;

в) мониторингу и анализу работы и эффективности СМИБ;

г) непрерывному усовершенствованию СМИБ на основе объективного измерения.

Рис. 11 иллюстрирует модель непрерывного циклического процесса менеджмента ИБ организации (модель Деминга), определенную требованиями раздела 4 международного стандарта ISO/IEC IS 27001.

На стадии планирования устанавливают политики информационной безопасности, цели, задачи, процессы и процедуры, адекватные потребностям в менеджменте риска ИБ и совершенствованию СМИБ, для достижения результатов в соответствии с политиками и целями организации.

На стадии реализации осуществляются внедрение и поддержка политики информационной безопасности организации, средств управления (защитных мер), регламентов, процессов и процедур СМИБ организации.

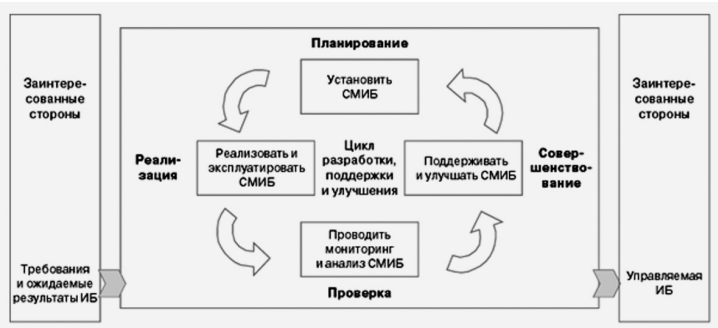


Рис. 11. Элементы процесса менеджмента ИБ

На стадии проверки осуществляются оценка и, если необходимо, измерение эффективности процессов менеджмента ИБ организации на соответствие требованиям политики информационной безопасности, целям и установленным практикам, обеспечивается отчетность высшему руководству о результатах для проведения соответствующего анализа.

На стадии совершенствования осуществляются выработка и принятие корректирующих и превентивных действий, основанных на результатах анализа, для достижения непрерывного усовершенствования СМИБ организации.

Использование для обеспечения ИБ организаций БС РФ «процессного подхода» на базе циклической модели Деминга, который является основой модели менеджмента стандартов качества ГОСТ Р КСО 9001 и ГОСТ Р КСО 14001, позволит обеспечить поддержку и интеграцию требований к различным системам менеджмента в рамках общего корпоративного менеджмента в организациях БС РФ. Требования настоящего стандарта к СМИБ для организаций БС РФ имеют прикладную практическую направленность, определяющую условия, цели и задачи применения в организациях БС РФ высокоуровневых международных стандартов для СМИБ организаций. Подобным прикладным стандартом является Рекомендация Международного союза электросвязи X.1051, обеспечивающая практическую основу по применению положений международного стандарта ISO/IEC IS 27001 в организациях, чей бизнес лежит в области телекоммуникаций.

Обеспечение ИБ организации включает реализацию и поддержку процессов осознания ИБ и процессов менеджмента ИБ.

Процессы осознания ИБ организации имеют отношение к руководству организации и определяют его ответственность в части реализации принципов обеспечения информационной безопасности организаций БС РФ, определенных положениями настоящего стандарта, а также требованиями раздела 5 «Ответственность высшего руководства организации» международного стандарта ISO/IEC IS 27001.

Процессы осознания ИБ должны охватывать всю организацию, а процессами менеджмента ИБ может быть охвачена ее часть или части. Обоснованием тому может быть ограниченность в ресурсах или време-

ни. Необходимо стремиться к тому, чтобы процессы менеджмента ИБ организации распространялись на всю ее деятельность.

Стратегия обеспечения ИБ организаций БС РФ, таким образом, заключается в развертывании, эксплуатации и совершенствовании СМИБ организации, включающей деятельность (процессы) менеджмента ИБ и стимулируемой и управляемой процессами осознания ИБ. Деятельность (процессы) СМИБ организации должна обеспечивать достижение целей деятельности организации в условиях:

- штатного функционирования;
- возникновения локальных инцидентов и проблем ИБ;
- возникновения широкомасштабных катастроф и аварий различной природы, последствия которых имеют или могут иметь отношение к ИБ организации БС РФ.

При этом менеджмент ИБ есть часть общего корпоративного менеджмента организации БС РФ, которая ориентирована на содействие достижению целей деятельности организации через обеспечение защищенности ее информационной сферы. Менеджмент ИБ не должен рассматриваться как самостоятельный вид деятельности в организации. Осознание ИБ обеспечивает основу эффективного функционирования СМИБ организации, где под эффективностью понимается соотношение между достигнутым результатом и использованными ресурсами.

9.7.3. Основные принципы обеспечения информационной безопасности организаций БС РФ

Своевременность обнаружения проблем. Организация должна своевременно обнаруживать проблемы, потенциально способные повлиять на ее бизнес-цели.

Прогнозируемость развития проблем. Организация должна выявлять причинно-следственную связь возможных проблем и строить на этой основе точный прогноз их развития.

Оценка влияния проблем на бизнес-цели. Организация должна адекватно оценивать степень влияния выявленных проблем на ее бизнес-цели.

Адекватность защитных мер. Организация должна выбирать защитные меры, адекватные моделям угроз и нарушителей, с учетом затрат на реализацию таких мер и объема возможных потерь от выполнения угроз.

Эффективность защитных мер. Организация должна эффективно реализовывать принятые защитные меры.

Использование опыта при принятии и реализации решений. Организация должна накапливать, обобщать и использовать как свой опыт, так и опыт других организаций на всех уровнях принятия решений и их исполнения.

Непрерывность принципов безопасного функционирования. Организация должна обеспечивать непрерывность реализации принципов безопасного функционирования.

Контролируемость защитных мер. Организация должна применять только те защитные меры, правильность работы которых может быть проверена, при этом организация должна регулярно оценивать

адекватность защитных мер и эффективность их реализации с учетом влияния защитных мер на бизнес-цели организации.

9.7.4. Специальные принципы обеспечения информационной безопасности организации

Реализация специальных принципов обеспечения ИБ направлена на повышение уровня зрелости процессов управления ИБ в организации.

Определенность целей. Функциональные цели и цели ИБ организации должны быть явно определены во внутрибанковском документе. Неопределенность приводит к «расплывчатости» организационной структуры, ролей персонала, политик ИБ и невозможности оценки адекватности принятых защитных мер.

Знание своих клиентов и служащих. Организация должна обладать информацией о своих клиентах, тщательно подбирать персонал (служащих), вырабатывать и поддерживать корпоративную этику, что создает благоприятную доверительную среду для деятельности организации по управлению активами.

Персонификация и адекватное разделение ролей и ответственности. Ответственность должностных лиц организации за решения, связанные с ее активами, должна персонифицироваться и осуществляться преимущественно в форме поручительства. Она должна быть адекватной степени влияния на цели организации, фиксироваться в политиках, контролироваться и совершенствоваться.

Адекватность ролей функциям и процедурам и их сопоставимость с критериями и системой оценки. Роли должны адекватно отражать исполняемые функции и процедуры их реализации, принятые в организации. При назначении взаимосвязанных ролей должна учитываться необходимая последовательность их выполнения. Роль должна быть согласована с критериями оценки эффективности ее выполнения. Основное содержание и качество исполняемой роли реально определяются применяемой к ней системой оценки.

Доступность услуг и сервисов. Организация должна обеспечить доступность для своих клиентов и контрагентов услуг и сервисов в установленные сроки, определенные соответствующими договорами (соглашениями) и/или иными документами.

Наблюдаемость и оцениваемость обеспечения ИБ. Любые предлагаемые защитные меры должны быть устроены так, чтобы результат их применения был явно наблюдаем (прозрачен) и мог быть оценен подразделением организации, имеющим соответствующие полномочия.

9.7.5. Модели угроз и нарушителей информационной безопасности организаций БС РФ

Модели угроз и нарушителей (прогноз ИБ) должны быть основным инструментом менеджмента организации при развертывании, поддержании и совершенствовании системы обеспечения ИБ организации.

Деятельность организации БС РФ поддерживается входящей в ее состав информационной инфраструктурой, которая обеспечивает реализацию банковских технологий и может быть представлена в виде иерархии следующих основных уровней:

• физического (линии связи, аппаратные средства и пр.);

• сетевого (сетевые аппаратные средства: маршрутизаторы, коммутаторы, концентраторы и пр.);

• сетевых приложений и сервисов;

• операционных систем (ОС);

• систем управления базами данных (СУБД);

• банковских технологических процессов и приложений;

• бизнес-процессов организации.

На каждом из уровней угрозы и их источники (в т.ч. злоумышленники), методы и средства защиты и подходы к оценке эффективности являются различными.

Главной целью злоумышленника является получение контроля над активами на уровне бизнес-процессов. Прямое нападение на уровне бизнес-процессов, например, путем раскрытия конфиденциальной банковской аналитической информации, более эффективно для злоумышленника и опаснее для собственника, чем нападение, осуществляемое через нижние уровни, требующее специфических опыта, знаний и ресурсов (в т.ч. временных) и поэтому менее эффективное по соотношению «затраты/получаемый результат».

Организация должна определить конкретные объекты защиты на каждом из уровней информационной инфраструктуры.

Наиболее актуальные источники угроз на физическом, сетевом уровнях и уровне сетевых приложений:

• внешние источники угроз: лица, распространяющие вирусы и другие вредоносные программы, хакеры и иные лица, осуществляющие несанкционированный доступ (НСД);

• внутренние источники угроз, реализующие угрозы в рамках своих полномочий и за их пределами (персонал, имеющий права доступа к аппаратному оборудованию, в том числе сетевому, администраторы сетевых приложений и т.п.);

• комбинированные источники угроз: внешние и внутренние, действующие совместно и/или согласованно.

Наиболее актуальные источники угроз на уровнях операционных систем, систем управления базами данных, банковских технологических процессов:

• внутренние, реализующие угрозы в рамках своих полномочий и за их пределами (администраторы ОС, администраторы СУБД, пользователи банковских приложений и технологий, администраторы ИБ и т.д.);

• комбинированные источники угроз: внешние и внутренние, действующие в сговоре.

Наиболее актуальные источники угроз на уровне бизнес-процессов:

• внутренние источники, реализующие угрозы в рамках своих полномочий и за их пределами (авторизованные пользователи и операторы АБС, представители менеджмента организации и пр.);

• комбинированные источники угроз: внешние (например, конкуренты) и внутренние, действующие в сговоре.

Также необходимо учитывать угрозы, связанные с природными и техногенными катастрофами и террористической деятельностью.

Источники угроз для реализации угрозы используют уязвимости объектов и системы защиты.

Хорошей практикой является разработка моделей угроз и нарушителей ИБ для данной организации.

Модель угроз ИБ включает описание источников угрозы, уязвимостей, используемых угрозами, методов и объектов нападений, пригодных для реализации угрозы, типов возможной потери (например, конфиденциальности, целостности, доступности активов), масштабов потенциального ущерба.

Для источников угроз – людей – может быть разработана модель нарушителя ИБ, включающая описание их опыта, знаний, доступных ресурсов, необходимых для реализации угрозы, и возможной мотивации их действий.

Степень детализации параметров моделей угроз и нарушителей ИБ может быть различна и определяется реальными потребностями для каждой организации в отдельности.

При анализе угроз ИБ необходимо исходить из того, что эти угрозы непосредственно влияют на операционные риски деятельности организации. Операционные риски сказываются на бизнес-процессах организации.

Операционные риски порождаются следующими эксплуатационными факторами: технические неполадки, ошибочные (случайные) и/или преднамеренные злоумышленные действия персонала организации, ее клиентов при их непосредственном доступе к АБС организаций и другими факторами.

Наиболее эффективным способом минимизации рисков нарушения ИБ для собственника является разработка совокупности мероприятий, методов и средств, создаваемых и поддерживаемых для обеспечения требуемого уровня безопасности информационных активов в соответствии с политикой ИБ организации БС РФ, разрабатываемой на основе моделей угроз и нарушителей ИБ.

9.7.6. Состав и назначение политики информационной безопасности организации БС РФ

Собственник (и/или менеджмент) организации должен обеспечить разработку, принятие и внедрение политики ИБ организации БС РФ, включая выделение требуемых для реализации этой политики ресурсов.

Политика ИБ должна описывать цели и задачи СМИБ и определять совокупность правил, требований и руководящих принципов в области ИБ, которыми руководствуется организация в своей деятельности.

Должны быть назначены лица, ответственные за реализацию политики ИБ и поддержание ее в актуальном состоянии.

Требования ИБ должны быть взаимоувязаны в непрерывный по задачам, подсистемам, уровням и стадиям жизненного цикла комплекс.

Требования ИБ должны определять содержание и цели деятельности организации БС РФ в рамках процессов управления ИБ.

Эти требования должны быть сформулированы как минимум для следующих областей:

- назначения и распределения ролей и доверия к персоналу;
- стадий жизненного цикла АБС;

• защиты от НСД, управления доступом и регистрацией в АБС, в телекоммуникационном оборудовании и автоматических телефонных станциях и т.д.;

• антивирусной защиты;

• использования ресурсов Интернет;

• использования средств криптографической защиты информации;

• защиты банковских платежных и информационных технологических процессов.

Политика ИБ организации БС РФ может учитывать и другие области, такие, как обеспечение непрерывности, физическая защита и т.д., отвечающие ее бизнес-целям.

Роль – это заранее определенная совокупность правил, устанавливающих допустимое взаимодействие между субъектом, например, сотрудником организации, и неким объектом, например, программно-аппаратным средством.

Для эффективного выполнения целей организации и задач по управлению активами должны быть выделены и определены соответствующие роли персонала организации. Роли следует персонифицировать с установлением ответственности за их исполнение. Формирование ролей, как правило, должно осуществляться на основании бизнес-процессов. Ответственность должна быть зафиксирована в должностных инструкциях.

При определении ролей для сотрудников организации БС РФ необходимо учитывать цели организации, имеющиеся ресурсы, функциональные и процедурные требования, критерии оценки эффективности выполнения правил для данной роли.

Не рекомендуется, чтобы одна персональная роль целиком отражала цель, например, включала все правила, требуемые для реализации бизнес-процесса. Совокупность правил, составляющих роли, не должна быть критичной для организации с точки зрения последствий успешного нападения на ее исполнителя. Не следует совмещать в одном лице (в любой комбинации) роли разработки, сопровождения, исполнения, администрирования или контроля, например, исполнителя и администратора, администратора и контролера или других комбинаций.

Роль должна быть обеспечена ресурсами, необходимыми и достаточными для ее выполнения.

Роли должны группироваться и взаимодействовать так, чтобы организационная структура соответствовала целям организации. Роль одного из руководителей организации (уполномоченного менеджера, высшего менеджера и т.п.) должна включать задачу координации своевременности и качества выполнения ролей сотрудников для достижения целей организации.

Ненадлежащее выполнение правил назначения и распределения ролей создает уязвимости.

Для контроля за качеством выполнения требований ИБ в организации должны быть выделены и определены роли по обеспечению ИБ.

При приеме на работу должны быть проверены идентичность личности, заявляемая квалификация, точность и полнота биографических фактов, наличие рекомендаций.

Лиц, которых предполагается принять на работу, связанную с защищаемыми активами или операциями, следует подвергать проверке в

части профессиональных навыков и оценки профессиональной пригодности. Рекомендуется выполнять контрольные проверки уже работающих сотрудников регулярно, а также внепланово при выявлении фактов их нештатного поведения, или участия в инцидентах ИБ, или подозрений в таком поведении или участии.

Весь персонал организации БС РФ должен давать письменное обязательство о соблюдении конфиденциальности, приверженности правилам корпоративной этики, включая требования по недопущению конфликта интересов. При этом условие о соблюдении конфиденциальности должно распространяться на всю защищаемую информацию, доверенную сотруднику или ставшую ему известной в процессе выполнения им своих служебных обязанностей.

Для внешних организаций требования по ИБ регламентируются положениями, включаемыми в договоры (соглашения).

Персонал организации должен быть компетентным для выполнения своих функций в области обеспечения ИБ. Компетентность персонала следует обеспечивать с помощью процессов обучения в области ИБ, осведомленности персонала и периодической проверки уровня компетентности.

Обязанности персонала по выполнению требований ИБ в соответствии с положениями ISO TR 13569 и ISO/IEC IS 17799-2005 следует включать в трудовые контракты (соглашения, договоры).

ИБ АБС должна обеспечиваться на всех стадиях жизненного цикла (ЖЦ) АБС, автоматизирующих банковские технологические процессы, с учетом всех сторон, вовлеченных в процессы ЖЦ (разработчиков, заказчиков, поставщиков продуктов и услуг, эксплуатирующих и надзорных подразделений организации).

При заказе АБС модель ЖЦ (стадии ЖЦ, этапы работ и процессы ЖЦ, выполняемые на этих стадиях) рекомендуется определять в соответствии с ГОСТ 34.601-90 и документом ISO/IEC IS 15288-2002.

Разработка технических заданий, проектирование, создание и тестирование и приемка средств и систем защиты АБС должны осуществляться по согласованию с подразделениями (лицами) в организации БС РФ, ответственными за обеспечение ИБ.

Ввод в действие, эксплуатация, снятие с эксплуатации АБС в части вопросов ИБ должны осуществляться при участии подразделения (лиц) в организации, ответственного за обеспечение ИБ.

На стадиях, связанных с разработкой АБС (определение требований заинтересованных сторон, анализ требований, архитектурное проектирование, реализация, интеграция и верификация, поставка, ввод в действие), разработчиком должна быть обеспечена защита от угроз:

- неверной формулировки требований к АБС;
- выбора неадекватной модели ЖЦ АБС, в том числе неадекватного выбора процессов ЖЦ и вовлеченных в них участников;
- принятия неверных проектных решений;
- внесения разработчиком дефектов на уровне архитектурных решений;
- внесения разработчиком недокументированных возможностей в АБС;
- неадекватной (неполной, противоречивой, некорректной и пр.) реализации требований к АБС;

• разработки некачественной документации;

• сборки АБС разработчиком/производителем с нарушением требований, что приводит к появлению недокументированных возможностей в АБС либо к неадекватной реализации требований;

• неверного конфигурирования АБС;

• приемки АБС, не отвечающей требованиям заказчика;

• внесения недокументированных возможностей в АБС в процессе проведения приемочных испытаний посредством недокументированных возможностей функциональных тестов и тестов ИБ.

Привлекаемые для разработки и(или) производства средств и систем защиты АБС на договорной основе специализированные организации должны иметь лицензии на данный вид деятельности в соответствии с законодательством РФ.

При приобретении организациями БС РФ готовых АБС и их компонентов разработчиком должна быть предоставлена документация, содержащая, в том числе описание защитных мер, предпринятых разработчиком в отношении перечисленных выше для АБС угроз.

Разработчиком должна быть представлена документация, содержащая описание защитных мер, предпринятых разработчиком АБС, и их компонентов относительно безопасности разработки, безопасности поставки и эксплуатации, поддержки жизненного цикла, включая описание модели жизненного цикла, оценки уязвимости. Данная документация может быть представлена в рамках декларации о соответствии или быть результатом оценки соответствия изделия, проведенной в рамках соответствующей системы оценки.

В договор (контракт) о поставке АБС и их компонентов организациям БС РФ рекомендуется включать положения по сопровождению поставляемых изделий на весь срок их службы. В случае невозможности включения в договор (контракт) указанных требований к разработчику должна быть рассмотрена возможность приобретения полного комплекта рабочей конструкторской документации на изделие, обеспечивающего возможность сопровождения АБС и их компонентов без участия разработчика. Если оба указанных варианта неприемлемы, например, вследствие высокой стоимости, руководство организации БС РФ должно обеспечить анализ влияния угрозы невозможности сопровождения АБС и их компонентов на обеспечение непрерывности бизнеса.

На стадии эксплуатации в соответствии с документом ISO TR 13569 должна быть обеспечена защита от следующих угроз:

• умышленное несанкционированное раскрытие, модификация или уничтожение информации;

• неумышленная модификация или уничтожение информации;

• недоставка или ошибочная доставка информации;

• отказ в обслуживании или ухудшение обслуживания.

Кроме этого, актуальной является угроза отказа от авторства сообщения.

На стадии сопровождения должна быть обеспечена защита от угроз:

• внесения изменений в АБС, приводящих к нарушению ее функциональности либо к появлению недокументированных возможностей;

• невнесения разработчиком/поставщиком изменений, необходимых для поддержки правильного функционирования и правильного состояния АБС.

На стадии снятия с эксплуатации должно быть обеспечено удаление информации, несанкционированное использование которой может нанести ущерб бизнес-деятельности организации, и информации, используемой средствами обеспечения ИБ, из постоянной памяти АБС или с внешних носителей.

Требования ИБ должны включаться во все договоры и контракты на проведение работ или оказание услуг на всех стадиях ЖЦ АБС.

При распределении прав доступа персонала и клиентов к активам организации БС РФ следует руководствоваться специальным принципом «знание своих клиентов и служащих», выражаемым следующим образом:

- «знать своего клиента»;
- «знать своего служащего»;
- «необходимо знать»[4],

а также руководствоваться принципом «двойное управление»[5].

В составе АБС должны применяться встроенные механизмы защиты информации, а также могут использоваться сертифицированные или разрешенные к применению средства защиты информации от НСД.

В организации должны обеспечиваться: идентификация, аутентификация, авторизация; управление доступом; контроль целостности; регистрация, включая:

- функционирование системы парольной защиты электронных вычислительных машин (ЭВМ) и локальных вычислительных сетей (ЛВС). Рекомендуется организовать службу централизованной парольной защиты для генерации, распространения, смены, удаления паролей, разработки необходимых инструкций, контроля за действиями персонала по работе с паролями;
- непротиворечивая и прозрачная административно-техническая поддержка задач управления доступом к ресурсам ЭВМ и/или ЛВС. Назначение/лишение полномочий по доступу сотрудников к ресурсам ЭВМ и/или ЛВС санкционируется руководителем функционального подразделения организации, несущего персональную ответственность за обеспечение ИБ в данном подразделении;
- контроль доступа пользователей к ресурсам ЭВМ и/или ЛВС. Оперативный контроль доступа пользователей осуществляется подразделениями (лицами) в организации, ответственными за обеспечение ИБ;
- формирование уникальных идентификаторов сообщений и идентификаторов пользователей (виды идентификаторов определяются особенностями конкретного технологического процесса);
- регистрация действий персонала и пользователей в специальном электронном журнале. Данный электронный журнал должен быть доступным для чтения, просмотра, анализа, хранения и резервного копирования только администратору ИБ. При невозможности поддержки данного режима эксплуатирующимися в организации БС РФ аппаратно-про-

4. «Необходимо знать» (Need to Know): принцип безопасности, который ограничивает доступ к информации и ресурсам по обработке информации тем, кому требуется выполнять определенные обязанности [ISO TR 13569].

5. «Двойное управление» (Dual Control): принцип сохранения целостности процесса и борьбы с искажением функций системы, требующий того, чтобы два лица независимо предпринимали некое действие до завершения определенных транзакций [ISO TR 13569].

граммными средствами реализация данного требования должна быть обеспечена организационными и/или административными мерами.

В ЭВМ и АБС не допускается присутствие и использование программного обеспечения и данных, не связанных с выполнением конкретных функций в банковских технологических процессах организации.

Устанавливаемое или изменяемое программное обеспечение должно быть предварительно проверено на отсутствие вирусов. После установки или изменения программного обеспечения должна быть выполнена антивирусная проверка.

При обнаружении компьютерного вируса необходимо принять меры по устранению последствий вирусной атаки, проинформировать руководство и приостановить при необходимости работу (на период устранения последствий вирусной атаки).

Ресурсы сети Интернет в организации БС РФ могут использоваться для ведения дистанционного банковского обслуживания (например, Internet-banking), получения и распространения информации, связанной с банковской деятельностью (путем создания информационных web-сайтов), информационно-аналитической работы в интересах организации, обмена почтовыми сообщениями исключительно с внешними организациями, а также ведения собственной хозяйственной деятельности.

Иное использование ресурсов сети Интернет, решение о котором не принято руководством организации в установленном порядке, должно рассматриваться как нарушение ИБ.

При принятии руководством организации решений об использовании сети Интернет для производственной и/или собственной хозяйственной деятельности необходимо учитывать следующие положения:

• сеть Интернет не имеет единого органа управления (за исключением службы управления пространством имен и адресов) и не является юридическим лицом, с которым можно было бы заключить договор (соглашение). Провайдеры (посредники) сети Интернет могут обеспечить только те услуги, которые реализуются непосредственно ими;

• гарантии по обеспечению ИБ при использовании сети Интернет никаким органом не предоставляются.

В организациях БС РФ, осуществляющих дистанционное банковское обслуживание клиентов, в связи с повышенными рисками информационной безопасности при взаимодействии с сетью Интернет обязательно должны применяться соответствующие средства защиты информации (межсетевые экраны, антивирусные средства, средства криптографической защиты информации (СКЗИ) и пр.), обеспечивающие прием и передачу информации только в установленном формате и только для конкретной технологии. Хорошей практикой является выделение и неподключение к внутренним сетям ЭВМ, с помощью которых осуществляется взаимодействие с сетью Интернет.

Почтовый обмен через сеть Интернет должен осуществляться с использованием защитных мер.

Хорошей практикой является наличие в организации ограниченного количества точек почтового обмена с сетью Интернет, состоящих из внешнего (подключенного к сети Интернет) и внутреннего (подключенного к внутренним сетям организации) почтовых серверов с безопасной системой репликации почтовых сообщений между ними (интернет-киоски).

Электронная почта должна архивироваться. Архив должен быть доступен только подразделению (лицу) в организации, ответственному за обеспечение ИБ. Изменения в архиве не допускаются. Доступ к информации архива должен быть ограничен.

В организациях БС РФ наличие банковской информации на ЭВМ, с помощью которых осуществляется взаимодействие с сетью Интернет в режиме on-line, определяется бизнес-целями организации. При этом необходимо учитывать высокую вероятность несанкционированного доступа, потери и искажения данной информации. Хорошей практикой является практика, когда ЭВМ, с помощью которых осуществляется взаимодействие с сетью Интернет в режиме on-line, не содержат никакой банковской информации (в т.ч. открытой).

8.2.6.7. Порядок подключения и использования ресурсов сети Интернет в организации БС РФ должен контролироваться подразделениями (лицами) в организации, ответственными за обеспечение ИБ. Любое подключение и использование сети Интернет должно быть санкционировано руководством функционального подразделения организации.

Система обеспечения информационной безопасности банковского платежного технологического процесса должна соответствовать требованиям настоящего стандарта и иных нормативных документов по вопросам информационной безопасности, действие которых распространяется на банковскую систему Российской Федерации.

В качестве объектов защиты должны рассматриваться:

- банковский платежный технологический процесс;
- платежная информация;
- технологический процесс по управлению ролями и полномочиями сотрудников организации БС РФ, задействованных в обеспечении банковского платежного технологического процесса.

Банковский платежный технологический процесс должен быть однозначно определен (отражен) в нормативно-методических документах организации БС РФ.

Порядок обмена платежной информацией должен быть зафиксирован в договорах между участниками, осуществляющими обмен платежной информацией. В роли участников могут выступать организации БС РФ, юридические и физические лица.

Сотрудники организации БС РФ, в том числе администраторы автоматизированных систем и средств защиты информации, не должны обладать всей полнотой полномочий для бесконтрольного создания, авторизации, уничтожения и изменения платежной информации, а также проведения операций по изменению состояния банковских счетов.

Результаты технологических операций по обработке платежной информации должны быть контролируемы (проверены) и удостоверены лицами/автоматизированными процессами. Лица/автоматизированные процессы, осуществляющие обработку платежной информации и контроль (проверку) результатов обработки, должны быть независимы друг от друга.

При работе с платежной информацией необходимо проводить авторизацию и контроль целостности данной информации.

Лучшей практикой при автоматизированной обработке платежной информации является оснащение средств вычислительной техники

(на которых осуществляются операции над платежной информацией) сертифицированными или разрешенными руководителем организации БС РФ к применению средствами защиты от НСД и средствами криптографической защиты информации.

Подготовленная клиентами организации БС РФ платежная информация, на основании которой совершаются расчетные, учетные и кассовые операции, предназначена для внутреннего использования в организации БС РФ и может быть передана иным организациям только в соответствии с действующим законодательством Российской Федерации.

Указанная информация относится к категории строгой отчетности. Ограничительные пометки (грифы) «Для служебного пользования», «Конфиденциально» или «Банковская тайна» на документы, содержащие данную информацию, не проставляются.

Безопасность информации, отнесенной к банковской тайне, обеспечивается в соответствии со статьей 26 Федерального закона «О банках и банковской деятельности».

Обязанности по администрированию средств защиты платежной информации для каждого технологического участка ее прохождения возлагаются приказом по организации БС РФ на сотрудников (сотрудника), задействованных на данном технологическом участке (администраторов информационной безопасности), с отражением этих функций в его должностных обязанностях.

Администратор информационной безопасности должен действовать на основании соответствующего нормативного документа, разработанного в организации БС РФ и утвержденного руководством организации БС РФ.

Хорошей практикой является назначение денежной надбавки администратору информационной безопасности к его должностному окладу.

Комплекс мер по обеспечению информационной безопасности банковского платежного технологического процесса должен предусматривать:

• защиту платежной информации от искажения, фальсификации, переадресации, несанкционированного уничтожения, ложной авторизации платежных документов;

• минимально необходимый, гарантированный доступ сотрудника организации БС РФ только к тем ресурсам банковского платежного технологического процесса, которые необходимы ему для исполнения служебных обязанностей или реализации прав, предусмотренных технологией обработки платежной информации;

• контроль (мониторинг) исполнения установленной технологии подготовки, обработки, передачи и хранения платежной информации;

• аутентификацию обрабатываемой платежной информации;

• двустороннюю аутентификацию автоматизированных рабочих мест, участников обмена платежной информацией;

• восстановление платежной информации в случае ее умышленного (случайного) разрушения (искажения) или выхода из строя средств вычислительной техники;

• авторизованный ввод платежной информации в автоматизированные банковские системы двумя сотрудниками с последующей программной сверкой результатов ввода на совпадение (Dual Control, ISO TR 13569);

• сверку выходных платежных сообщений с соответствующими поступившими платежными сообщениями;

• гарантированную доставку платежных сообщений участникам обмена.

Организации БС РФ – члены международных платежных систем с использованием банковских карт должны обеспечивать выполнение требований данных систем по информационной безопасности.

8.2.9.1. Система обеспечения информационной безопасности банковского информационного технологического процесса должна соответствовать требованиям настоящего стандарта и иных нормативных документов по вопросам информационной безопасности, действие которых распространяется на БС РФ.

В организации БС РФ неплатежная информация классифицируется как:

• открытая информация, предназначенная для официальной передачи во внешние организации и средства массовой информации;

• внутренняя банковская информация, предназначенная для использования исключительно сотрудниками организации БС РФ при выполнении ими своих служебных обязанностей;

• информация, содержащая сведения ограниченного распространения в соответствии с утвержденным организацией БС РФ Перечнем, подлежащая защите в соответствии с законодательством РФ, например, банковская тайна, персональные данные;

• информация, полученная из федеральных органов исполнительной власти и содержащая сведения ограниченного распространения;

• информация, содержащая сведения, составляющие государственную тайну.

Каждому виду информации соответствует свой необходимый уровень защиты (свой набор требований по защите).

Так как требования по защите двух последних видов информации определяются государственными нормативно-методическими документами, то вопросы обеспечения защиты информации, содержащей указанные сведения, в настоящем стандарте не рассматриваются. Автоматизированные системы организации БС РФ, обрабатывающие, хранящие и/или передающие такую информацию, должны быть физически изолированы от прочих автоматизированных систем данной организации.

В качестве объектов защиты должны рассматриваться:

• информационные ресурсы;

• управляющая информация АБС;

• банковский информационный технологический процесс.

Организация БС РФ несет ответственность за:

• достоверность информации, официально предоставляемой внешним организациям и гражданам;

• достоверность и выполнение регламента предоставления внешним организациям и гражданам информации, обязательность и порядок предоставления которой определены законодательством Российской Федерации и/или нормативными документами Банка России;

• обеспечение соответствующего законодательству Российской Федерации уровня защиты как собственной информации, так и информации, официально полученной из внешних организаций и от граждан.

Если в АБС обрабатывается информация, требующая по решению руководства защиты, то соответствующим распоряжением должен быть назначен администратор информационной безопасности. Допускаются назначение одного администратора информационной безопасности на несколько АБС, а также совмещение выполнения указанных функций с другими обязанностями.

При этом совмещение в одном лице функций администратора АБС и администратора информационной безопасности АБС не допускается.

Администратор АБС не должен иметь служебных полномочий (а при возможности и технических средств) по настройке параметров системы, влияющих на полномочия пользователей по доступу к информации. Однако он должен иметь право добавить в систему нового пользователя без всяких полномочий по доступу к информации, а также удалить из системы такого пользователя.

Администратор информационной безопасности АБС должен иметь служебные полномочия и технические возможности по контролю действий соответствующих администраторов АБС (без вмешательства в их действия) и пользователей, а также полномочия (а при возможности и технические средства) по настройке для каждого пользователя только тех параметров системы, которые определяют права доступа к информации. Устанавливаемые права доступа к информации должны назначаться подразделением организации БС РФ, ответственным за эту информацию (владельцем информационного актива).

Администратор информационной безопасности не должен иметь права добавить нового пользователя в АБС, а также удалить из нее существующего пользователя.

В случае отсутствия у администратора информационной безопасности технических возможностей по настройке параметров АБС, влияющих на полномочия пользователей по доступу к информации, эти настройки выполняются администратором АБС, но с обязательным предварительным согласованием устанавливаемых прав доступа пользователей к информации с администратором информационной безопасности.

Для каждой АБС должен быть определен порядок контроля ее функционирования со стороны лиц, отвечающих за ИБ.

Процессы подготовки, ввода, обработки и хранения информации, а также порядок установки, настройки, эксплуатации и восстановления необходимых технических и программных средств должны быть регламентированы и обеспечены инструктивными и методическими материалами, согласованными со службой информационной безопасности.

Должна осуществляться и быть регламентирована процедура периодического тестирования всех реализованных программно-техническими средствами функций (требований) по обеспечению ИБ. Регламентирующие документы должны быть согласованы со службой ИБ.

Должна осуществляться и быть регламентирована процедура восстановления всех реализованных программно-техническими средствами функций по обеспечению ИБ. Регламентирующие документы должны быть согласованы со службой ИБ.

9.7.7. Система менеджмента информационной безопасности организации БС РФ

Для успешного функционирования СМИБ организации БС РФ следует реализовать следующие процессы:

а) определение/уточнение области действия СМИБ и выбор подхода к оценке рисков ИБ. Определение/уточнение области действия СМИБ должно осуществляться на основе результатов оценки операционных рисков, а также оценки репутационных и правовых рисков деятельности организации БС РФ;

б) анализ и оценка рисков ИБ, варианты обработки рисков ИБ для наиболее критичных информационных активов и бизнес-процессов организации. При анализе и оценке рисков ИБ должны использоваться положения раздела 7 настоящего стандарта;

в) определение/уточнение политики для СМИБ организации;

г) выбор/уточнение целей ИБ и защитных мер и их обоснование для минимизации рисков ИБ. Цели ИБ и защитные меры могут быть выбраны на основе раздела 8 настоящего стандарта, а дополнительно на основе:

1) международного стандарта ISO/IEC IS 27001-2005 или положений международного стандарта ISO/IEC IS 17799-2005, обеспечивающего большую детализацию;

2) стандартов ISO TR 13569, COBIT, BSI PAS 56 и других руководств по обеспечению информационной безопасности;

3) ГОСТ Р КСО/МЭК 15408-1ч3-2002 в части требований к продуктам информационных технологий;

4) стандартов ISO/IEC TR 18028, ISO/IEC TR 18043, ISO/IEC TR 18044 и других стандартов для отдельных областей обеспечения ИБ.

Обоснование по обработке рисков с учетом применения защитных мер должно быть подготовлено в виде отдельного документа (аналогичного документу «Statement of applicability» по ISO/IEC IS 27001), являющегося основой для разработки плана обработки рисков ИБ;

д) принятие менеджментом организации БС РФ остаточных рисков и решения о реализации и эксплуатации/совершенствовании СМИБ. Остаточные риски ИБ должны быть соотнесены с рисками банковской деятельности и оценено их влияние на достижение целей деятельности организации БС РФ.

Организации БС РФ следует реализовать следующие процессы:

а) разработка плана обработки рисков ИБ;

б) реализация плана обработки рисков ИБ и реализация защитных мер, управление работами и ресурсами, связанными с реализацией СМИБ;

в) реализация программ по обучению и осведомленности ИБ. Реализация процесса по обучению и осведомленности ИБ должна обеспечиваться с учетом требований раздела 8 международного стандарта ISO/IEC IS 17799-2005;

г) обнаружение и реагирование на инциденты безопасности. Реализация процесса обнаружения и реагирования на инциденты безопасности должна обеспечиваться с учетом требований раздела 13 международного стандарта ISO/IEC IS 17799-2005 и технического отчета ISO/IEC TR 18044;

д) обеспечение непрерывности бизнеса и восстановления после прерываний. Обеспечение непрерывности бизнеса и восстановления после прерываний должны обеспечиваться с учетом требований раздела 14 международного стандарта ISO/IEC IS 17799-2005 и положений BSI PAS-56.

Процессы мониторинга и анализа СМИБ организации должны быть интегрированы в систему внутреннего контроля организаций БС РФ.

Организации следует реализовать следующие процессы мониторинга и анализа СМИБ:

а) мониторинг и контроль защитных мер, включая регистрацию действий и событий, связанных со СМИБ;

б) анализ эффективности СМИБ, включая анализ уровней остаточного и приемлемого рисков ИБ;

в) внутренний аудит СМИБ;

г) анализ СМИБ со стороны высшего руководства;

д) проведение периодического внешнего аудита СМИБ.

Организации следует реализовать следующие процессы:

а) реализация тактических улучшений в СМИБ, осуществляемых в рамках полномочий служб (ответственных) ИБ организации;

б) реализация стратегических улучшений СМИБ, требующих принятия решений на уровне руководства организации и инициирования процессов планирования СМИБ. Использование опыта;

в) информирование об изменениях и их согласование с заинтересованными сторонами;

г) оценка достижения поставленных целей и потребностей в развитии СМИБ.

Организации следует использовать систему менеджмента документации для СМИБ. Документы в данной системе необходимо соответствующим образом защищать и контролировать. Данная система должна также включать любые записи, которые создаются или поддерживаются для обеспечения свидетельств эффективной работы СМИБ.

Организации следует разработать и внедрить план обеспечения непрерывности бизнеса (деятельности) и восстановления после прерываний. Данный план и соответствующие процессы восстановления должны пересматриваться на регулярной основе и своевременно обновляться (например, при существенных изменениях в операционной деятельности, организационной структуре, бизнес-процессах и автоматизированных банковских системах). Эффективность документированных процедур восстановления необходимо периодически проверять и тестировать (как минимум на полугодовой основе). С данным планом должны быть ознакомлены все сотрудники, отвечающие за его выполнение и вовлеченные в процессы восстановления.

В качестве методологической основы при разработке плана могут быть использованы общепринятые международные стандарты, регулирующие вопросы менеджмента непрерывности бизнеса (например, BSI PAS-56).

Для реализации задач развертывания и эксплуатации СМИБ организации рекомендуется иметь в своем составе (самостоятельную или в составе службы безопасности) службу ИБ (уполномоченное лицо). Службу ИБ (уполномоченное лицо) рекомендуется наделить следующими полномочиями:

• управлять всеми планами по обеспечению ИБ организации;

• разрабатывать и вносить предложения по изменению политики ИБ организации;

• изменять существующие и принимать новые нормативно-методические документы по обеспечению ИБ организации;

• выбирать средства управления и обеспечения ИБ организации;

• контролировать пользователей, в первую очередь пользователей, имеющих максимальные полномочия;

• контролировать активность, связанную с доступом и использованием средств антивирусной защиты, а также связанную с применением других средств обеспечения ИБ;

• осуществлять мониторинг событий, связанных с ИБ;

• расследовать события, связанные с нарушениями ИБ, и в случае необходимости выходить с предложениями по применению санкций в отношении лиц, осуществивших противоправные действия, например, нарушивших требования инструкций, руководств по обеспечению ИБ организации;

• участвовать в действиях по восстановлению работоспособности АБС после сбоев и аварий;

• создавать, поддерживать и совершенствовать систему управления ИБ организации.

Хорошей практикой является создание службы ИБ и выделение ей своего собственного бюджета.

Хорошей практикой является, когда служба ИБ организации имеет собственного куратора на уровне Первого лица в руководстве организации БС РФ (Председателя или заместителя Председателя правления и т.п.). При этом служба ИБ и служба информатизации (автоматизации) не должны иметь общего куратора.

Организационная основа менеджмента ИБ в организациях должна определяться целями бизнеса организации на финансовом рынке, размерами организации, наличием сети филиалов и другими факторами.

Организациям, имеющим сеть филиалов или региональных представительств, рекомендуется выделить соответствующие подразделения ИБ на местах, обеспечив их соответствующими ресурсами и нормативной базой.

9.7.8. Проверка и оценка информационной безопасности организации БС РФ

Проверка и оценка ИБ организации может быть произведена с помощью аудита, самооценки и мониторинга ИБ.

Аудит ИБ организации БС РФ может быть внутренним или внешним. Цель, порядок и периодичность проведения аудитов ИБ организации в целом (или ее отдельных структурных подразделений) или АБС определяется руководством организации на основе потребностей в такой деятельности и фиксируется в программе аудита ИБ.

Цель аудита ИБ организации состоит в проверке и оценке соответствия ИБ требованиям настоящего стандарта. Заключение по результатам проведения аудита ИБ организации должно показывать:

• текущий уровень ИБ организации;

• уровень зрелости процессов менеджмента ИБ организации;

• уровень осознания ИБ организации.

При проведении аудита ИБ организации должны использоваться стандартные процедуры документальной проверки, опрос и интервью с руководством и персоналом организации. При необходимости уточнения результатов документальной проверки, опросов и интервью в рамках внутреннего аудита ИБ в качестве дополнительного способа может применяться «проверка на месте», которая проводится для обеспечения уверенности в том, что конкретные защитные меры реализуются, правильно используются и проверяются с помощью тестирования. Обстоятельства, при которых требуется дополнительный способ в рамках внутреннего аудита ИБ, должны быть определены и согласованы в плане проведения аудита ИБ в организации.

При проведении внутреннего аудита ИБ могут использоваться журналы регистрации инцидентов ИБ, ведущиеся службами безопасности организации и формируемые на основе данных мониторинга ИБ.

При проведении внешнего аудита ИБ руководство организации должно обеспечить документальное и, если это необходимо, техническое подтверждение того, что:

• политика ИБ отражает требования бизнеса и цели организации;

• организационная структура управления ИБ создана;

• процессы выполнения требований ИБ исполняются и удовлетворяют поставленным целям;

• защитные меры (например, межсетевые экраны, средства управления физическим доступом) настроены и используются правильно;

• остаточные риски оценены и остаются приемлемыми для организации;

• система управления ИБ соответствует определенному уровню зрелости управления ИБ;

• рекомендации предшествующих аудитов ИБ реализованы.

Аудиторский отчет должен храниться в организации в течение установленного времени. Доступ к аудиторскому отчету должен быть разрешен только руководству организации и руководителям подразделения (лицам), ответственным за ИБ в организации.

Хорошей практикой подготовки к аудиту ИБ и проверки уровня ИБ организации БС РФ является проведение самооценки ИБ. Самооценка ИБ проводится собственными силами и по инициативе руководства организации. При проведении самооценки ИБ должны использоваться журналы регистрации инцидентов ИБ, ведущиеся службами безопасности организации и формируемые на основе данных мониторинга ИБ, проверяться эффективность реализованных защитных мер путем тестовых проверок (могут быть проверки на проникновение).

Мониторинг ИБ должен проводиться персоналом организации, ответственным за ИБ, с целью обнаружения и регистрации отклонений функционирования защитных мер от требований ИБ и оценки полноты реализации положений политики ИБ, инструкций и руководств обеспечения ИБ в организации.

Основными целями мониторинга ИБ в организации являются оперативное и постоянное наблюдение, сбор, анализ и обработка данных под заданные руководством цели. Такими целями анализа могут быть:

• контроль за реализацией положений нормативных актов по обеспечению ИБ в организации;

• выявление нештатных (или злоумышленных) действий в АБС организации;

• выявление инцидентов ИБ.

Для целей оперативного и постоянного наблюдения объектов мониторинга могут использоваться как специализированные (например, программные) средства, так и штатные (входящие в коммерческие продукты и системы) средства регистрации действий пользователей и процессов.

ЛИТЕРАТУРА

1. Зегжда Д.П., Ивашко А.М. Как построить защищенную информационную систему? Под научной ред. Зегжды Д.П. и Платонова В.В. – Спб: Мир и семья-95, 1997 – 312 стр., с илл. ISBN-88857-010X.

2. Гостехкомиссия России. Руководящий документ. Концепция защиты средств вычислительной техники от несанкционированного доступа к информации. М.: Военное издательство, 1992.

3. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. М.: Военное издательство, 1992.

4. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. М.: Военное издательство, 1992.

5. Гостехкомиссия России. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники. М.: Военное издательство, 1992.

6. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. М.: Военное издательство, 1992.

7. Trusted Computer System Evaluation Criteria. US Department of Defense. CSC-STD-00l-83, Aug. 1983.

8. Trusted Network Interpretation. National Computer Security Center. NCSC-TG-005 Version 1, July 1987.

9. Trusted Database Management System Interpretation. National Computer Security Center. NCSC-TG-021 Version 1, April 1991.

10. A guide to understanding discretionary access control in trusted systems. National Computer Security Center. NCSC-TG-003 Version 1, September 1987.

11. Password management guideline. US Department of Defense. CSC-STD-002-85, April 1985.

12. Guidance for applying the Department of Defense Trusted Computer System Evaluation Criteria in specific environment. US Department of Defense. CSC-STD-003-85, June 1985.

13. A Guide to Understanding Audit in Trusted Systems. National Computer Security Center. NCSC-TG-001, July 1987.

14. Guide to understanding configuration management in trusted systems. National Computer Security Center. NCSC-TG-006-88, March 1988.

15. The Interpreted Trusted Computer System Evaluation Criteria Requirements. National Computer Security Center. NCSC-TG-005-95, Jan. 1995.

16. Information Technology Security Evaluation Criteria. Harmonized Criteria Of France-Germany-Netherlands-United K,ngdom. – Department of Trade and Industry. London, 1991.

17. [B&L] Bell, D. E. and LaPadula, L. J., Secure Computer Systems: Unified Exposition and MULTICS Interpretation, Revision 1, US Air Force ESD-TR-75-306, MITRE Corporation MTR-2997, Bedford MA, March 1976. (Безопасные компьютерные системы: объединенное представление и интерпретация (MULTICS).

18. [Biba] Biba, K. J., Integrity Considerations for Secure Computer Systems, ESD-TR-372, ESD/AFSC, Hanscom AFB, Bedford MA., April 1977.(Исследования целостности для безопасных компьютерных систем).

19. [CTCPEC] Canadian Trusted Computer Product Evaluation Criteria, Version 3.0, Canadian System Security Centre, Communications Security Establishment, Government of Canada, January 1993. (Канадские критерии оценки доверенных компьютерных продуктов).

20. [FC] Federal Criteria for Information Technology Security, Draft Version 1.0, (Volumes I and II),jointly published by the National Institute of Standards and Technology and the National Security Agency, US Government, January 1993. (Федеральные критерии оценки для безопасности информационных технологий).

21. [Gogu1] Goguen, J. A. and Meseguer, J., «Security Policies and Security Models,» 1982 Symposium on Security and Privacy, pp.11-20, IEEE, April 1982. (Политики и модели безопасности).

22. [Gogu2] Goguen, J. A. and Meseguer, J., «Unwinding and Inference Control,» 1984 Symposium on Security and Privacy, pp.75-85, IEEE, May 1984. (Развертывание и логический контроль).

23. [ITSEC] Information Technology Security Evaluation Criteria, Version 1.2, Office for Official Publications of the European Communities, June 1991. (Критерии оценки безопасности информационных технологий).

24. [ISO/IEC 7498-2:1989] Information processing systems – Open Systems Interconnection – Basic Reference Model, Part 2: Security Architecture. (Системы обработки информации-Взаимодействие открытых систем-Основная модель монитора обращений, часть 2: Архитектура безопасности).

25. [TCSEC] Trusted Computer Systems Evaluation Criteria, US DoD 5200.28-STD, December 1985. (Критерии оценки доверенных компьютерных систем).

СПИСОК СОКРАЩЕНИЙ

АБС – автоматизированная банковская система;
АО – ассоциированный объект;
БС – банковская система;
ЖЦ – жизненный цикл;
ИБ – информационная безопасность;
КА – код аутентификации;
КС – компьютерная система;
КЦ – контроль целостности;
ЛВС – локальная вычислительная сеть;
ЛС – локальный сегмент;
ЛСУ – локальный субъект управления;
МБО – монитор безопасности объектов;
МБС – монитор безопасности субъектов;
МО – монитор обращений;
МРЗФ – модуль реализации защитных функций;
МЭ – межсетевой экран;
НСД – несанкционированный доступ;
ОНСД – опосредованный несанкционированный доступ;
ОМ – менеджер объектов, подсистема МБО;
ОС – операционная среда;
ОУ – объект управления;
ПБ – политика безопасности;
ПИТ – продукт информационных технологий;
ПО – программное обеспечение;
ПРД – правила разграничения доступа;
ПФ – пакетный фильтр;
РПВ – разрушающее программное воздействие;
СКЗИ – средство криптографической защиты информации;
СМИБ – система менеджмента информационной безопасности;
ТПО – телекоммуникационное программное обеспечение;
УС – управляющий субъект;
УСУ – удаленный субъект управления;
УЦ – удостоверяющий центр;
ЭЦП – электронная цифровая подпись;
ФПУ – фильтр прикладного уровня;
ЭВМ – электронная вычислительная машина;
ЭЦП – электронная цифровая подпись.