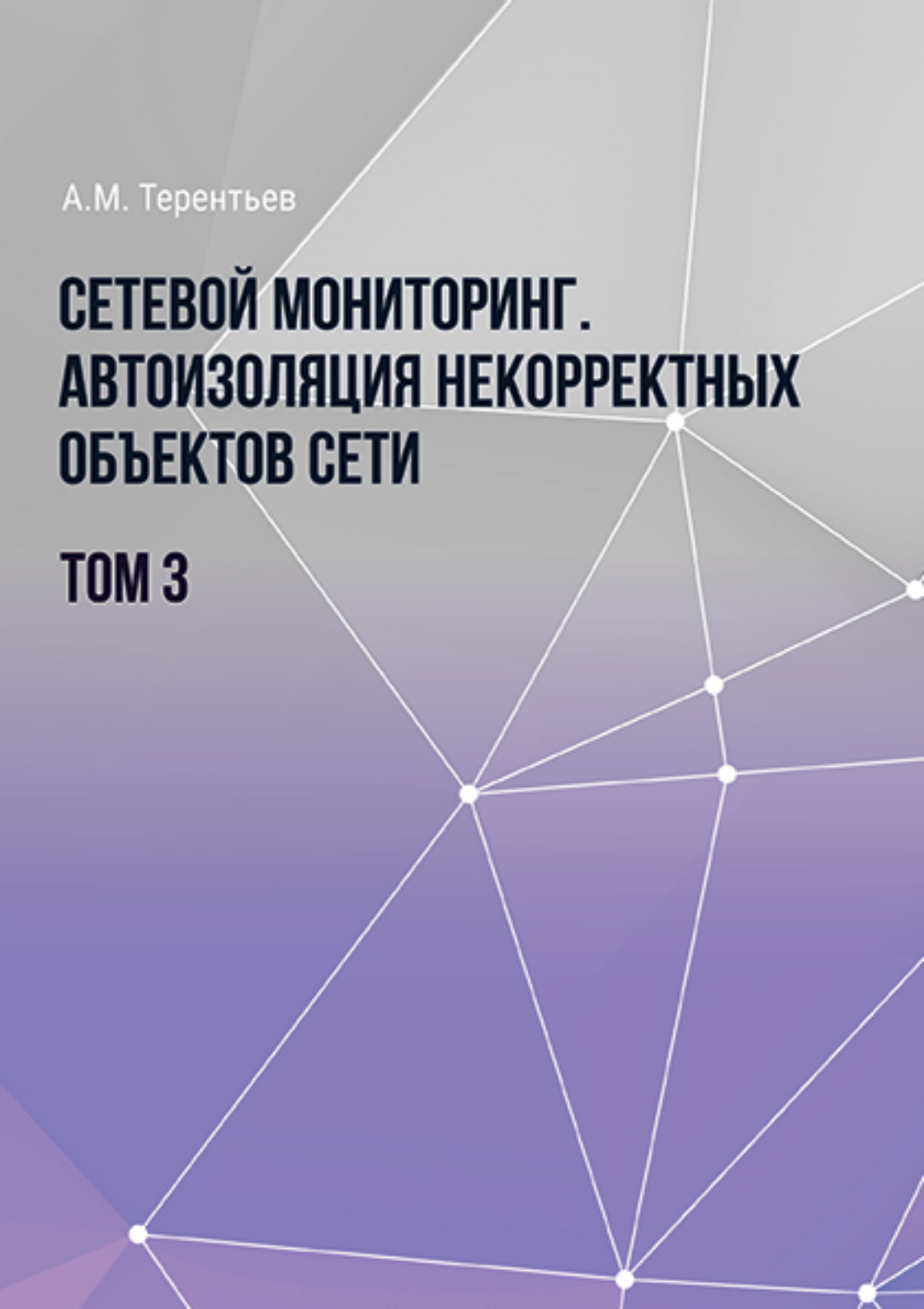
А.М. Терентьев

СЕТЕВОЙ МОНИТОРИНГ. АВТОИЗОЛЯЦИЯ НЕКОРРЕКТНЫХ ОБЪЕКТОВ СЕТИ

ТОМ 3



Федеральное государственное бюджетное учреждение науки
«Центральный экономико-математический институт РАН»

А.М. ТЕРЕНТЬЕВ

Сетевой мониторинг. Автоизоляция некорректных объектов сети. Том 3

Монография

Чебоксары
Издательский дом «Среда»
2021

УДК 004.42 + 004.49
ББК 32.973.5
Т35

Рецензенты:

Хрусталёв Евгений Юрьевич, д-р экон. наук,
г.н.с. ФГБУН «Центральный экономико-математический институт РАН»

Тельнов Юрий Филиппович, д-р экон. наук,
зав. кафедрой прикладной информатики и информационной безопасности ФГБОУ «Российский экономический университет им. Г.В. Плеханова»

Терентьев А.М.

Т35 **Сетевой мониторинг. Автоизоляция некорректных объектов сети. Т. 3** : монография / А.М. Терентьев. – Чебоксары: ИД «Среда», 2021. – 104 с.

ISBN 978-5-907411-18-0

В данной работе описана оригинальная технология круглосуточного отслеживания сетевых пакетов, циркулирующих в локальной сети.

Технология базируется на выделенной рабочей станции, работающей в MS-DOS и принимающей все доступные пакеты. Агрегированные данные передаются на соседний Windows-компьютер через serial-соединение. Мониторная программа на этом Windows-ПК способна отследить заражённые ПК и выполнить действия по изоляции их от локальной сети. Эта функция исполняется с помощью коммутатора Cisco.

В данном томе описываются научно-практические исследования, позволившие создать среду автоматического отключения заражённых сетевыми вирусами ПК от локальной сети. Описана также интегрирующая система, выводящая оперативные данные о загрузке сети для наблюдения из Интернета.

За проведённые исследования по данной тематике в 2003 г. автор удостоен учёного звания “Doctor of Philosophy” Европейской Академии информатизации (Брюссель).

Монография рекомендована к печати Учёным советом Федерального государственного бюджетного учреждения науки «ЦЭМИ РАН».

ISBN 978-5-907411-18-0
DOI 10.31483/a-10275

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Использование персональных компьютеров (ПК) практически во всех научных и производственных единицах немыслимо без задействования специальных средств их объединения в единую локальную вычислительную сеть (ЛВС) предприятия. Общение через ЛВС каждого ПК, сервера или иного сетевого устройства требует наличия в нём сетевого адаптера – специальной платы с характеристиками, соответствующими типу сетевых соединений. Сетевое устройство с сетевым адаптером (ПК, сервер, сетевой принтер, каждый сетевой выход коммутатора и т.п.) будем называть нода[1].

Современные средства организации ЛВС могут также включать различные сетевые устройства, организующие ЛВС и выход в Интернет – коммутаторы, маршрутизаторы, хабы, свитчи и др. Эти устройства делают для каждой ноды возможность иметь доступ к другим.

Обмен информацией по ЛВС подробно рассмотрен в томе 1 данной работы [1]. Показано, что по умолчанию сетевые адаптеры настроены так, чтобы принимать либо сетевые пакеты, адресованные «всем» (broadcasting), либо данной ноде-устройству. Для этого, каждый сетевой адаптер имеет свой, уникальный в мире, технологический MAC-адрес (Media Access Control Address), используемый на нижних уровнях модели ISO-OSI. Для идентификации ноды на верхних уровнях ISO-OSI используется IP-адрес (Internet Protocol Address), уникальный внутри ЛВС или во всём мире.

Особенности организации локальных сетей также рассмотрены в [1]. Там же определён термин «корпоративная сеть учреждения» (КВС), введённый автором ранее [2]. Далее было показано, что принципиально возможно техническое решение, при котором некоторый ПК, называемый Наблюдающей станцией (НС), принимает *все* пакеты, циркулирующие в сети. На базе этого решения создан прототип сетевого мониторинга – программы, наблюдающей проходящие пакеты, с возможностью дампировать все или избранные из них на HDD для последующего подробного анализа.

---

[1] Этот термин, как и множество других (E-Mail, смайлик и пр.) пришёл в Интернет из самодеятельной международной сети FIDOnet (1985 г). Вместо IP-адреса там был общемировой аналог, например, автор данной работы известен, в частности, под адресом **2:5020/614.13**.

Физический обмен данными осуществляется с высокой скоростью, от 10Мбит/с до 1Гбит/с. В обычной работе, конечный пользователь не имеет возможности просмотреть конкретное содержимое пакетов во время обмена и проанализировать все уровни соглашений модели ISO-OSI. Однако, созданная разработка дала такую принципиальную возможность.

Программа сетевого мониторинга является профессиональным полнофункциональным пакетом [3], созданным по гранту РФФИ 04-07-90260в (с 2004 по 2006 гг.) и впоследствии неоднократно пополненным.

С течением времени, схема подключения компьютеров и структура локальной сети ЦЭМИ РАН несколько раз изменялась. Средства сетевого мониторинга обязаны были быть адаптированными к актуальным изменениям. Изменения в структуре ЛВС 2004-2006 гг позволили дополнить пакет средствами автоматизированного исключения некорректно работающих ПК из ЛВС.

Данный том работы посвящен созданию средств сетевого мониторинга, позволяющих автоматически исключать некорректно настроенные ПК из ЛВС и осуществлять таким образом автоматическую «самоочистку» ЛВС. Базовым средством таких исключений является коммутатор Cisco Catalyst.

Таким образом, **объектами данного исследования** являлись, с одной стороны, развивающаяся КВС института, и, с другой стороны, особенности настройки и эксплуатации коммутаторов Cisco в этих сетях.

Достигнутые результаты излагаются в последовательности их получения. В данном томе сосредоточены сведения о развитии структуры ЛВС, научные исследования по созданию и развитию автоматической связи с коммутаторами Cisco Catalyst технических средств сетевого мониторинга. **В результате исследований** были сформированы актуальные алгоритмы соединения с коммутаторами Cisco Catalyst, синтезирован язык описания моделей управления коммутаторами, созданы необходимые технические и программные средства и достигнута **цель работы** – создан полноценный комплекс программно-технических и организационных решений, обеспечивающий круглосуточное наблюдение и регистрацию некорректно настроенных компьютеров с автоматическим отсечением их от корпоративной сети. Дополнительно, агрегированные данные наблюдения доведены до демонстрации в Интернете.

Достигнута желаемая **степень внедрения** – результаты проведенных исследований нашли практическое применение в институте. **Итоги внедрения** – результаты исследований использовались в широком спектре, как для биллинговых статистических исследований, так и для реформирования корпоративной вычислительной сети института.

Возможная **область применения** работы – распространение созданного технического решения на другие организации – достигнута в ходе разработки.

Экономическую **эффективность** работы – не представляется возможным определить вследствие уникальности проведённых исследований и выполненных работ. Сама возможность оперативно, автоматизированно реагировать на заражённые вирусами ПК вследствие реально протекающих с большой скоростью процессов в локальной сети представляет качественно новый метод воздействия на сетевые потоки информации в корпоративных сетях.

Сфера практического применения весьма разнообразна, от выявления некорректно настроенных компьютеров с возможностью их автоматического отключения до фиксации сетевых атак и необходимости реорганизации локальной сети. Большинство применений сетевого мониторинга рассмотрено в [3], возможность автоматического отключения и передача агрегированной информации в Интернет будут рассмотрены здесь.

ГЛАВА 1. УПРАВЛЕНИЕ КОММУТАТОРОМ CISCO CATALYST

Выбор и нацеленность средств сетевого мониторинга заявлены в программной работе [13]. К таковым можно отнести следующие.

1) Оптимизация построения корпоративной сети

■ Постоянный замер общего трафика, выявление пиковых нагрузок.

■ Выявление наиболее загруженных участков сети, возможно, тормозящих общий трафик.

■ Выявление «зацикливаний» при передаче пакетов в сети.

2) Экономические (биллинговые) приложения

■ Учёт трафика интересующей группы ПК.

■ Суммарный учёт Интернет-трафика, возможно, по подразделениям организации (института).

■ Учёт трафика, специализированного по нужному критерию (SQL-сервер, ресурсные системы Интернета, локальная почта и т.д.).

3) Информационная безопасность

■ Выявление некорректно настроенных ПК и/или серверов.

■ Выявление некорректно работающих сетевых устройств.

■ Выявление вирусной активности в сети с определением источника.

■ Индикация неработоспособности («падения») серверов.

■ Отслеживание попыток взлома, нерегламентированных доступов, хакерских атак.

Описанные задачи представляются наиболее интересными и не описывают, разумеется, всех возможностей сетевого мониторинга. Значительную часть поставленных задач удалось решить в ходе разработки и внедрения этого программного средства.

Созданный прототип, как показано в [1], успешно решал задачи мониторинга при невысоких сетевых скоростях порядка Vcp^2 30-50 КБ/с. Профессиональный вариант [3] снял эти ограничения по скорости. Однако, развитие ЛВС с течением времени стало диктовать решение новых задач.

[2] Все обозначения введены в [1] и подробно описаны там же.

1.1. Коммутатор Cisco в локальной сети

Малые сети редко нуждаются в специальном проектировании. В средних сетях на начальных стадиях их развития основные проблемы проектирования сводятся к оптимизации длины соединяющих кабелей, сохранению не более трех уровней в схеме подключения концентраторов и обеспечению безопасности отдельных ПК и серверов. Так, в сентябре 2001 г. схема соединений ЛВС ЦЭМИ представляла собой единое некоммутируемое пространство. Как известно [1], в такой сети каждый пакет, испущенный любым ПК, через концентраторы виден всем остальным ПК этой сети (т.е. анализируется их сетевыми адаптерами). Мониторинг на этой сетевой топологии [2] показал, что при одновременной сетевой активности более 50 ПК резко нарастало число сетевых коллизий и понижалась эффективная пропускная способность сети в целом. К аналогичным последствиям вела и повышенная активность ряда ПК и серверов. Неисправность одной сетевой карты вызывала торможение всей сети. Появление сетевого вируса (напр., атаки сетевыми вирусами Code Red [4] и Nimda [5] в 2001–2002 гг. с поражением ряда ПК и др.) при невозможности оперативно локализовать и отключить пораженный ПК, являющийся источником распространения вируса по ЛВС и Интернету, вызывало длительный его поиск. К этим проблемам добавлялась неэффективность работы сети вследствие произвольности каскадного подключения концентраторов с разными скоростями (10 или 100 Мбит/с).

В то же время, несомненным достоинством указанной схемы являлось то, что включение разработанных в ЦЭМИ РАН средств мониторинга сети [2] в любую точку ЛВС давало аутентичные данные об общей загрузке сети и трафиках конкретных ПК, позволяя ставить и оперативно решать разные [1] задачи отслеживания нужных компонент потоков по нужным сетевым объектам (серверам, сетевым принтерам и отдельным ПК). Запланированная реконструкция сети, вместе с решением задачи улучшения характеристик ее пропускной способности, должна была сохранить возможность полного охвата мониторингом всех сетевых объектов.

Введение в строй коммутатора Cisco Catalyst в 2003 г (см. рис. 1) резко изменило пропускную способность сети, физически разбив ее на фрагменты по числу задействованных рабочих портов. Теперь испущенный любым ПК пакет, попадая на порт коммутатора,

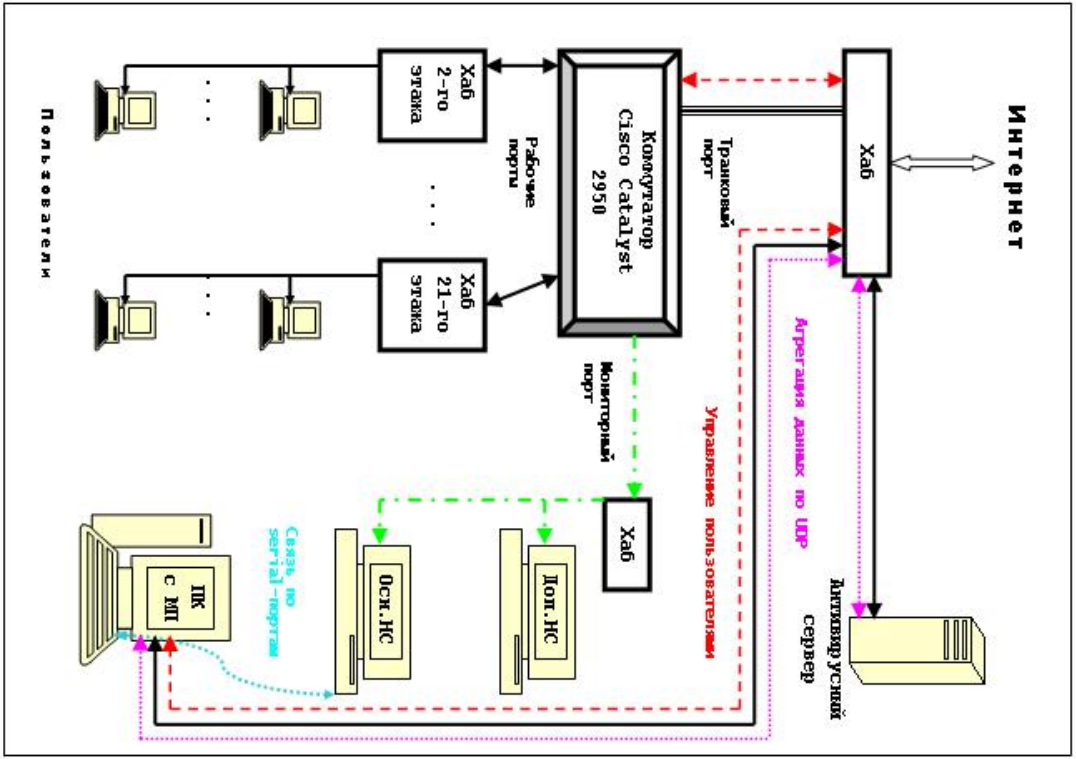


Рис. 1. Схема сетевого мониторинга с 2006 г.

к которому он подключен, передается только на порт коммутатора, соответствующий MAC-адресу устройства назначения. Если исходный и принимающий ПК находятся на одном и том же порту коммутатора, ходящие между ними пакеты вообще не пропускаются коммутатором на другие его порты. К примеру, пакеты между Антивирусным сервером и мониторной станцией (МП, рис. 1) достигают только ПК, подключенных к их общему концентратору (хабу), но не других ПК сети. Это и есть то основное свойство коммутаторов, которое дает основание его применения в сетях. Применительно к ЦЭМИ РАН, речь идет о модели Catalyst-2924, однако рассуждения статьи справедливы и для других моделей коммутаторов.

Помимо обеспечения развязывания трафика, коммутатор Catalyst-2924 решает и другие задачи. Важной из них является то, что его FastEthernet-порты независимо обеспечивают скорость 10 или 100 Мбит/сек, так что скорость порта всегда соответствует скорости подключенного к нему концентратора [6], и имевшиеся ранее проблемы каскадирования концентраторов с разными характеристиками с введением в строй коммутатора отступают.

Наконец, наиболее важной в смысле обсуждаемой тематики особенностью введенного в строй коммутатора является его возможность фильтрации пакетов конкретных ПК по их MAC-адресам. Режим **Secure**, который может быть введен на коммутаторе по каждому порту независимо от прочих, позволяет задать исчерпывающий список MAC-адресов сетевых устройств, пакеты от которых обслуживаются коммутатором и для которых разрешена коннективность к остальной сети. Это означает, что при введении режима Secure любой ПК, MAC-адрес которого не включен в специальные таблицы коммутатора, не сможет получить доступ к серверам ЛВС, серверам КВС и Интернету (за исключением компьютеров, подключенных к общему с ним концентратору, которым «видны» прочие ПК данного фрагмента сети). Реализация коммутатором этой важной функции позволяет реально взять под жесткий оперативный контроль допуск новых ПК к ЛВС ЦЭМИ.

Управление списками Secure-адресов по портам коммутатора и иными сервисами Catalyst-2924 осуществляется с ограниченного числа ПК, перечень которых определяется с помощью обычного метода ACL – списков управления доступом (подробнее см. [7]). Обращаясь снова к рис. 1, легко видеть, что пакеты управлением

коммутатора (красная пунктирная линия) поступают через транковый порт Catalyst-2924. Таким образом, управляющие пакеты не видны большинству пользователей ЛВС, что обеспечивает изоляцию управляющего трафика и повышает безопасность ЛВС.

Особый вид порта – **мониторный** – позволяет автоматически копировать на него пакеты, проходящие в/из прочих заданных портов. Именно наличие мониторного порта даёт возможность организовать наблюдение всех или проходящих в/на интересуемые порты коммутатора сетевых пакетов.

Проведенные нами эксперименты установили точный смысл режима **Secure** в IOS 12.0[3] (подробнее см. [7]).

> Находящийся в Secure-режиме порт коммутатора коммутирует только первый из нарушающих Secure-режим пакетов с образованием состояния Violation, сохраняемое на порту до истечения Aging Time. Представляется целесообразным соотнести устанавлимое значение Aging Time с периодом формирования отчетов основной наблюдающей станцией (15 минут).

Введение Secure-режима сразу же дало ряд положительных результатов: все вводимые в строй новые ПК вынуждены предварительно проверяться и настраиваться специалистами Отделения; блокированы попытки подключения к ЛВС ПК с некорректно настроенными сетевыми подключениями; обеспечена возможность оперативного отключения выборочных ПК за несоблюдение правил эксплуатации сети.

Конфигурация Catalyst-2924 при этом включает строки, указанные на рис. 2[4]. Отметим, что транковые порты не нуждаются в мониторинге: все проходящие через них пакеты мониторируются при прохождении рабочих портов.

В 2004 г. с помощью доработанного программного обеспечения НС было проведено исследование полноты мониторинга, обеспечиваемого Catalyst 2924 с IOS 12.0. В исследованиях посылались тестовые ARP-пакеты [14], исходящие от фиктивных MAC-

[3] Операционная система Cisco Catalyst-2924.

[4] Для обеспечения конфиденциальности конкретные номера портов заменены звездочками.

адресов, направленные на реальные MAC-адреса сетевых ПК. Исследованиями установлено, что:

– созданная схема мониторинга обеспечивает полноту протоколирования пакетов, циркулирующих по обычным (рабочим) портам;

– не регистрируются пакеты, не проходящие через коммутатор вследствие нарушения Secure-режима (кроме самого первого за время Aging Time);

– пакеты, проходящие между реальными ПК, подключенными к одному и тому же порту, **не регистрируются**, за исключением бродкастинга (ARP-запрос является бродкастингом и регистрируется всегда, ARP-ответ является направленным и регистрируется лишь в том случае, если пакет коммутируется).

Таким образом, вместе с центральным местом в ЛВС ЦЭМИ РАН, которое занимает коммутатор Cisco Catalyst-2924, определено новое понимание мониторинга ЛВС как учет и разбор пакетов, проходящих через этот коммутатор. Отметим, что для учета и исследования полного трафика серверов ЛВС желательно, чтобы они были подсоединены к отдельному порту коммутатора. Выбранная схема размещения серверов Интернет, к примеру, гарантирует мониторинг всех пакетов между ними и любым ПК ЛВС ЦЭМИ, однако не включает внешний трафик между этими серверами и Интернетом.

Введение в строй коммутатора Cisco Catalyst-2924 открывает принципиально новую возможность оперативного управления компьютерами ЛВС ЦЭМИ РАН. Как показано в [8], администрирование вручную не удовлетворяет по скорости современным требованиям к реакции на инциденты информационной безопасности. Поэтому, автором разработан метод автоматического отсечения ПК ЛВС ЦЭМИ РАН, нарушающих установленные правила поведения в ЛВС, с использованием информации наблюдающих станций и интерфейса с коммутатором.

В частности, нарушающими правила следует считать ПК, испускающие любые пакеты с какими-либо ip-адресами, кроме явно назначенного[5]. Дело в том, что кроме ПК с явно установленным неверным ip-адресом, источником посторонних ip-адресов могут служить ПК, зараженные сетевыми вирусами и использующие

[5] Технологические адреса типа 192.168.*.* и др. не рассматриваются и пропускаются.

фальсифицированный адрес источника (например, Linux.Sorso [9]; W32.Blaster.D.Worm [10]; W32.HLLW.Kazmor [11] и др.), имеющие возможности организации атак типа DoS и dDoS [12]. Для пресечения указанных нарушений сетевой безопасности также может быть применено автоматическое отсечение от сети. Программное обеспечение этой возможности начато реализацией в 2005 г.

```
   Using 13275 out of 32768 bytes
   ! Last configuration change at 18:36:03 UTC Sun Aug 8
2004
   ! NVRAM config last updated at 18:36:11 UTC Sun Aug 8
2004
   !
   version 12.0
   no service pad
   service timestamps debug datetime msec localtime
show-timezone
   service timestamps log datetime msec localtime show-
timezone
   service password-encryption
   !...
   interface FastEthernet0/1
    description trunk-1 (from 9-th rack)
    port group 1
    switchport trunk encapsulation dot1q
    switchport mode trunk
   !...
   interface FastEthernet0/**
    description NOC CEMI - Room 921
    port security max-mac-count 13
    switchport access vlan 194
   !...
   interface FastEthernet0/23
    description Network audit system
    duplex full
    speed 100
    port monitor FastEthernet0/**
   !...
    port monitor FastEthernet0/24
    switchport access vlan 194
   !...
   end
```

Рис. 2. Конфигурирование коммутатора для обеспечения мониторинга

К моменту создания этого метода, к сожалению, в ЛВС ЦЭМИ присутствовали ПК, грубо нарушавшие правило однозначности ip-адреса. На рис. 3 приведен пример только одного ПК, в те годы непосредственно администрируемого зав. Лабораторией локальных сетей[6] (полностью см. в [7]).

Приведенная на рис. 3 информация свидетельствует, что упомянутый ПК (ему назначен адрес 193.232.194.33) излучает сам или пропукает через себя пакеты с явно сфальсифицированными адресами[7]. Это однозначно свидетельствует о присутствии вирусов либо в данном ПК, либо в сети, маршрутизируемой[8] через него.

Несмотря на регулярную информацию автором руководства, означенная ситуация, как видно из рис. 3, **продолжалась почти год**, с 25.09.2003 по 02.09.2004. Более того, по жалобе нарушителя руководством ЦЭМИ РАН было выражено недовольство к автору данной работы, опубликовавшему эти данные в [7] (!).

Наличие сетевого мониторинга позволяет оперативно уточнять причину появления Secure-инцидентов. С начала 2003 г за 8 месяцев было зарегистрировано 3928 инцидентов (см. рис. 4). Большинство их составляют пакеты: новых ПК в процессе установки; долгое время неиспользованных ПК при настройке режима Secure; ПК, проходящих паспортизацию. В то же время, 223 инцидента (5,68% от всех) составили случаи, которые объяснить не удалось. В целом, их стоит рассматривать как неизвестные ПК, самовольно подключенные к ЛВС.

[6] С 2010 г. А.Ю.Смитиенко, к счастью, не является зав.лаб. Локальных сетей.

[7] К примеру, 64.12.200.89 принадлежит America Online, Inc, Manassas,VA-20109, US; 199.67.204.28 - Salomon Inc., Rutherford, NJ, US; 205.188.2.26 - America Online, Inc, Sterling, VA-20166,US; и т.д.

[8] Например, через личный DialUp, сконфигурированный на данном ПК; либо через вторую сетевую карту, подключенную к завирусованной сети.

```
03.009027370F50 193.232.194.033 43:Смитиенк/0917 25/09-3=19:00 0002 08/10-4=00:00 8738
07.009027370F50 207.046.107.080 43:Смитиенк/0917 01/01-4=23:00 0380 01/01-4=23:00 0380
07.009027370F50 207.046.106.053 43:Смитиенк/0917 02/01-4=18:00 0456 02/01-4=18:00 0456
07.009027370F50 207.046.106.054 43:Смитиенк/0917 03/01-4=01:00 0484 03/01-4=01:00 0484
07.009027370F50 207.046.106.190 43:Смитиенк/0917 03/01-4=11:00 0524 03/01-4=11:00 0524
07.009027370F50 169.254.116.201 43:Смитиенк/0917 03/01-4=22:00 0568 03/01-4=00:00 0576
07.009027370F50 169.254.030.192 43:Смитиенк/0917 09/01-4=15:00 1116 09/01-4=15:00 1116
07.009027370F50 207.046.106.151 43:Смитиенк/0917 09/01-4=16:00 1120 09/01-4=16:00 1120
07.009027370F50 064.012.024.023 43:Смитиенк/0917 09/01-4=16:00 1120 15/01-4=01:00 1638
07.009027370F50 205.188.179.233 43:Смитиенк/0917 09/01-4=20:00 1136 09/01-4=20:00 1136
07.009027370F50 064.012.024.021 43:Смитиенк/0917 09/01-4=20:00 1136 20/01-4=11:00 2158
07.009027370F50 064.012.024.253 43:Смитиенк/0917 12/01-4=23:00 1438 20/01-4=10:00 2154
07.009027370F50 064.012.024.020 43:Смитиенк/0917 12/01-4=23:00 1438 12/01-4=23:00 1438
07.009027370F50 064.012.026.117 43:Смитиенк/0917 13/01-4=21:00 1526 20/01-4=17:00 2182
07.009027370F50 064.012.024.254 43:Смитиенк/0917 15/01-4=01:00 1638 15/01-4=01:00 1638
07.009027370F50 151.001.209.073 43:Смитиенк/0917 17/01-4=23:00 1918 17/01-4=23:00 1918
07.009027370F50 193.045.014.142 43:Смитиенк/0917 17/01-4=23:00 1918 17/01-4=23:00 1918
07.009027370F50 064.012.026.118 43:Смитиенк/0917 16/01-4=16:00 1794 18/01-4=18:00 1994
07.009027370F50 207.046.104.020 43:Смитиенк/0917 20/01-4=10:00 2154 20/01-4=10:00 2154
07.009027370F50 207.046.106.181 43:Смитиенк/0917 20/01-4=11:00 2158 20/01-4=11:00 2158
07.009027370F50 169.254.003.111 43:Смитиенк/0917 14/02-4=09:00 4542 14/02-4=09:00 4542
.....
07.009027370F50 194.067.027.124 43:Смитиенк/0917 12/08-4=21:00 3251 17/08-4=10:00 3687
07.009027370F50 081.019.066.097 43:Смитиенк/0917 12/08-4=21:00 3251 14/08-4=02:00 3367
07.009027370F50 081.222.128.012 43:Смитиенк/0917 14/08-4=02:00 3367 14/08-4=02:00 3367
07.009027370F50 194.067.057.050 43:Смитиенк/0917 14/08-4=10:00 3399 14/08-4=10:00 3399
07.009027370F50 217.016.018.203 43:Смитиенк/0917 17/08-4=10:00 3687 17/08-4=10:00 3687
07.009027370F50 169.254.172.211 43:Смитиенк/0917 20/08-4=02:00 3943 20/08-4=02:00 3943
07.009027370F50 205.188.248.201 43:Смитиенк/0917 02/09-4=02:00 5191 02/09-4=02:00 5191
```

Рис. 3. Примеры постоянных нарушений эксплуатации ЛВС одним из ПК

Порт	MAC-адрес	Кол-во	Пользователь	Подраздел.
9	0000.c035.fcc2	7	Сопцов	201
9	0003.472c.8158	123	Поляк С.	401
15	00e0.4ceb.3255	658	Глазырин	109
15	000b.cd08.e06e	9	Терушкин	303
9	0090.f524.68dc	320	Полтерович	103
3	00c0.26a6.439d	38	?	?
3	00c0.262c.15f4	155	Носова	902
9	00c0.f021.7135	389	Селиверстов	801
9	0003.478e.0fb9	10	?	?
13	00c0.26a6.638b	22	?	ИСЭПН
13	5254.4c17.fa94	59	Овчарова	ИСЭПН
13	0002.440c.9dd1	393	Попова	ИСЭПН
13	5254.4c17.eb04	170	Токсанбаева	ИСЭПН
16	00ca.ca00.0000	6	Тестовая проверка	402
13	5254.4c17.fa43	49	Корхова	ИСЭПН
9	0010.22ff.6178	153	?	?
13	0000.1c3a.43a5	1367	Орлов	ИСЭПН
Всего инцидентов		**3928**		
В т.ч. необъясненных		**223**	5,68%	

Рис. 4. Сводка инцидентов нарушения Secure-режима за 8 месяцев 2003 г.

В качестве иллюстрации изменения режима в сети при замене центрального хаба коммутатором можно предложить рисунки 5 и 6.

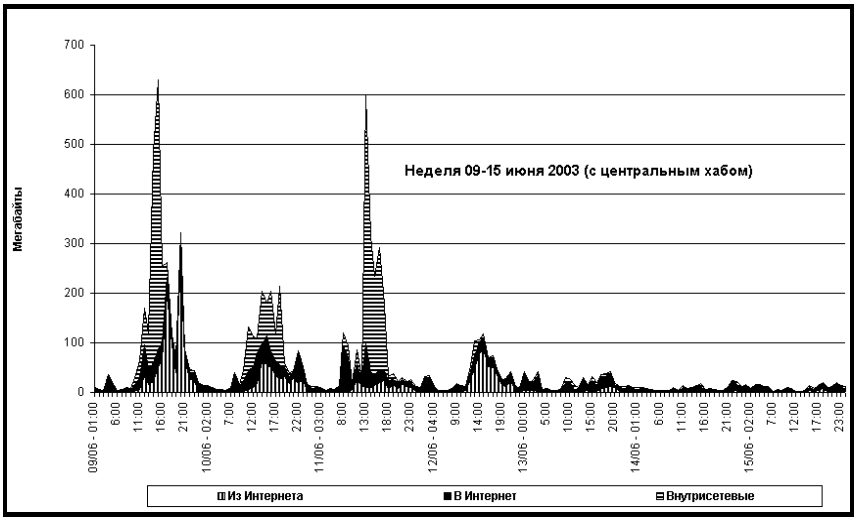


Рис. 5. Трафик в ЛВС за неделю до введения Catalyst-2924

Поскольку масштаб по оси ординат на этих графиках сильно изменен, зрительное восприятие этих диаграмм может быть неадекватным. На самом деле суммарные цифровые данные (см. рис. 7) по второй неделе более, чем вдвое выше. Разумеется, такая картина долго не задержалась: после завершения периода отпусков трафик сильно возрос.

Аналогичный характер имеют и данные средних скоростей: смысл внедрения коммутатора именно в «развязывании» трафика, т.е. в устранении общей очереди ожидания сетевых адаптеров и распараллеливании потоков по фрагментам ЛВС, подключенным к разным портам коммутатора.

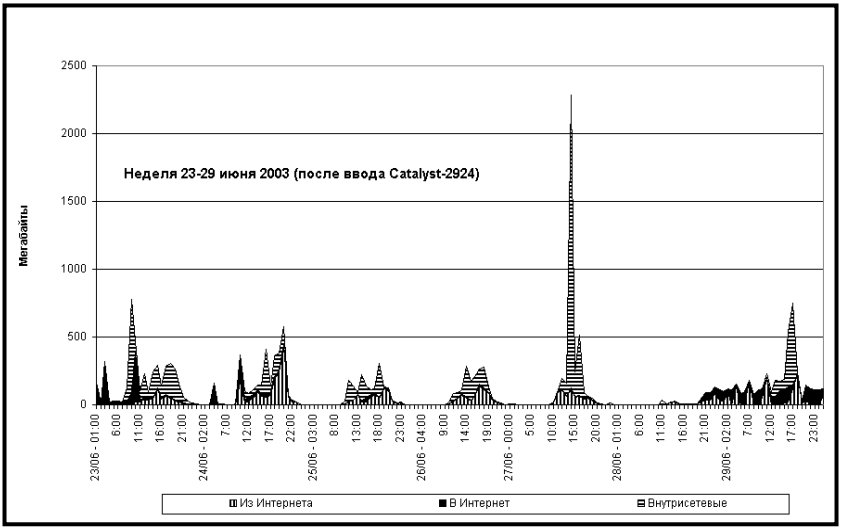


Рис. 6. Трафик в ЛВС через неделю после введения Catalyst-2924

	09÷15 июня	23-29 июня
Трафик из Интернета [Мб]	2118,99	5418,35
Трафик в Интернет [Мб]	2569,09	3826,09
Внутрисетевые пересылки [Мб]	3916,91	9956,66
Всего за неделю [Мб]	**8604,99**	**19201,10**

Рис. 7. Данные о недельном трафике до и после введения Catalyst-2924

Суммируя, из рассмотренного можно заключить, что введение коммутатора в ЛВС:

- не исключает возможность мониторинга корпоративной сети и не отменяет его необходимость;
- меняет смысл мониторинга, исключая из рассмотрения локальный трафик фрагментов сети;
- открывает возможность полноценного аудита сети в смысле оперативного автоматического администрирования;
- требует специальной конфигурации коммутатора для обеспечения мониторинга и полноценного аудита корпоративной сети.

1.2. Возможность полуавтоматического управления сетевыми коммутаторами Cisco Catalyst

Современные средние и тем более крупные локальные сети по разным причинам всё более нуждаются в наличии специализированных средств сетевого управления – маршрутизаторов и коммутаторов. Одной из особых причин этого является необходимость оперативного отключения от сети [8] ПК с некорректными настройками, завирусованных или пораженных иными вредоносными программами, мешающими работе сети.

В этих целях актуально наличие и развитие программных средств, допускающих автоматизированную работу по заранее определенным алгоритмам, либо полуавтоматическое управление сетевыми управляющими устройствами малоквалифицированным пользователем по опять-таки заранее предопределенным шаблонам.

Однако, разнообразные имеющиеся средства управления сетевыми коммутаторами (далее для единообразия будет идти речь именно о них, хотя те же рассуждения актуальны и для иных устройств сетевого управления), интенсивно развиваясь в методах их управления, были и остаются самостоятельным замкнутым классом устройств. Интенсивное развитие GUI-ориентированных методов, от «Web Console» и «Visual Switch Manager» до «Cluster Management Suite», и «Cisco Works», по-прежнему предполагает привлечение высококвалифицированных специалистов для любых операций.

Между тем, большинство конкретных действий по управлению коммутаторами, представляется, может быть сведено к нескольким типовым схемам, управление которыми может осуществляться специалистом средней квалификации по заранее разработанным шаблонам.

Таким образом, **актуальной становится проблема автоматизации управления сетевыми коммутаторами**. Для ее решения автором впервые предложено было создать специализированные средства полуавтоматического управления, действующие по заранее разработанным шаблонам.

Решение указанной проблемы должно базироваться на создании программных средств взаимодействия с сетевыми коммутаторами. Центральным вопросом создания этих средств является выбор

актуального сетевого протокола. В связи с универсальностью, исследовалась попытка создания подобных средств на базе сетевого протокола Telnet [15-57].

Выбор Telnet для создания средств полуавтоматического управления сетевыми коммутаторами является не случайным: он стал дефакто стандартным протоколом дистанционного управления сетевыми устройствами. Так, даже в современных продуктах серии Cisco, маршрутизаторах и коммутаторах различного назначения, несмотря на интенсивное развитие GUI-ориентированных программных средств, Telnet с его крайне отсталой командной строкой включен во все базовые операционные средства коммутаторов и маршрутизаторов фирмы Cisco Systems.

Казалось бы, при таком уверенном месте, занимаемым этим протоколом, рынок программных приложений должен изобиловать средствами, реализующими большинство его компонент на уровне DLL-библиотек. Давно назревшая необходимость интеграции процессов управления коммутаторами в общую технологичную среду управления локальными сетями должна была бы привести к достаточному выбору автоматизированных управляемых программных средств, предназначенных для включения в конструируемую среду управления сетью. Однако, по нашему мнению, этого не произошло. Рынок предлагает разнообразные реализации протокола, являющиеся законченным программным продуктом, не допускающим ни настройки на актуальный список управляемых сетевых устройств, ни автоматизации в наборе длинных и труднозапоминаемых команд управления этими устройствами. Даже речи нет о какой-либо автоматизации действий пользователя: не предлагается предварительно создаваемых шаблонов, выделенных групп команд, макросредств и т.д.

Такая отсталость в средствах автоматизации типового сетевого протокола управления тем более странна, что большинством потребителей Telnet являются системные сетевые администраторы – лица высокой квалификации, либо сами являющиеся программистами, либо имеющие тесные связи с таковыми.

Под полуавтоматическими программными средствами управления сетевыми коммутаторами автором понимается программный модуль, взаимодействующий в сеансах связи с управляемым устройством. После команды (пользователя или из шаблона), запущенное программное средство в очередном сеансе связи должно

выполнить заранее определенный специально разработанный сценарий, возможно, с некоторыми параметрами (MAC-адресом сетевого устройства, номером порта и др.). Предполагается обеспечить возможность автоматизированного контроля за ходом исполнения, хотя бы методом проверки ответных реплик. Выходом программы должно являться новое состояние коммутатора и бинарный индикатор успешности завершения заданного сценария. В связи с особой важностью для состояния сети процессов управления сетевым коммутатором, весь ход исполнения сценария должен быть сохранен в виде читабельного текстового протокола.

Несмотря на кажущуюся простоту, протоколу Telnet среди серии протоколов TCP/IP посвящено, пожалуй, едва ли не наибольшее число RFC-документов ([15-57], приведены только актуальные на момент создания документы). Это объясняется бурным развитием актуального протокола сетевого управления.

Протокол Telnet представляет собой имитацию «сетевого телетайпа», включающего устройства посимвольного ввода и построчного вывода. В процессе управления характерен посимвольный обмен между управляющим «сетевым телетайпом» и управляемым устройством. Вводимая команда завершается стандартным концом строки. Ответным действием управляемого устройства может быть высылка одной или более текстовых строк. В специальных случаях подаются дополнительные сигналы, например при превышении определенного числа выведенных строк далее выводится мерцающий текст “**-- More--**” и ожидается реакция управления в виде нажатия клавиши <Enter>.

В конкретных реализациях протокола обязательно присутствуют также служебные команды как последовательности спецсимволов, включенных в произвольное место основного текстового обмена данными. Эти последовательности принято называть опциями протокола. Одна из принципиальных трудностей реализации состоит в определении широты реализованных опций в конкретных версиях Telnet, «зашитых» в действующую IOS коммутатора. Разумеется, никакой информации по этому вопросу в обширной документации сопровождения сетевых коммутаторов не имеется. Неясна полнота реализации протокола, не указаны конкретные типы имитируемых терминалов, не описан список поддерживаемых опций. Иными словами, требовалось синтезировать

алгоритм исполнения Telnet для конкретного сетевого коммутатора, включая все детали.

Для исследования всех этих вопросов были проведены эксперименты на реальном современном сетевом коммутаторе Cisco Catalyst-2935 с IOS 12.1 с задействованием разработанных ранее средств сетевого мониторинга [12]. В сеансах связи стандартной Windows-реализации протокола Telnet с одного из рабочих мест с сетевым коммутатором, все сетевые пакеты связи были запротоколированы с помощью наблюдающей станции (НС), настроенной на отбор пакетов (как отсылаемых, так и получаемых) для дампирования по MAC-адресу коммутатора. Вывод дампированных пакетов получен утилитой ***tamview.exe*** из комплекта программ, сопровождающих разработку математического обеспечения НС. Схема этого эксперимента приведена на рис. 8.

Эксперимент проведен в условиях, исключающих иное стороннее наблюдение изнутри ЛВС, кроме рабочей НС: ip-связь с тестируемым устройством (выделено жирной линией) проходила, минуя компьютеры ЛВС.

В целях понимания представленных результатов эксперимента укажем, что тестовый сеанс связи проходил с ПК с MAC-адресом **004005411AAF**, ip-адресом **193.232.194.13** на коммутатор с MAC-адресом **000DBCBBC780** и ip-адресом **193.232.194.25**; состоял всего лишь в вводе логина, пароля и команды **EXIT**. На рис. 9 показаны первые 8 пакетов (их общее число – 70).

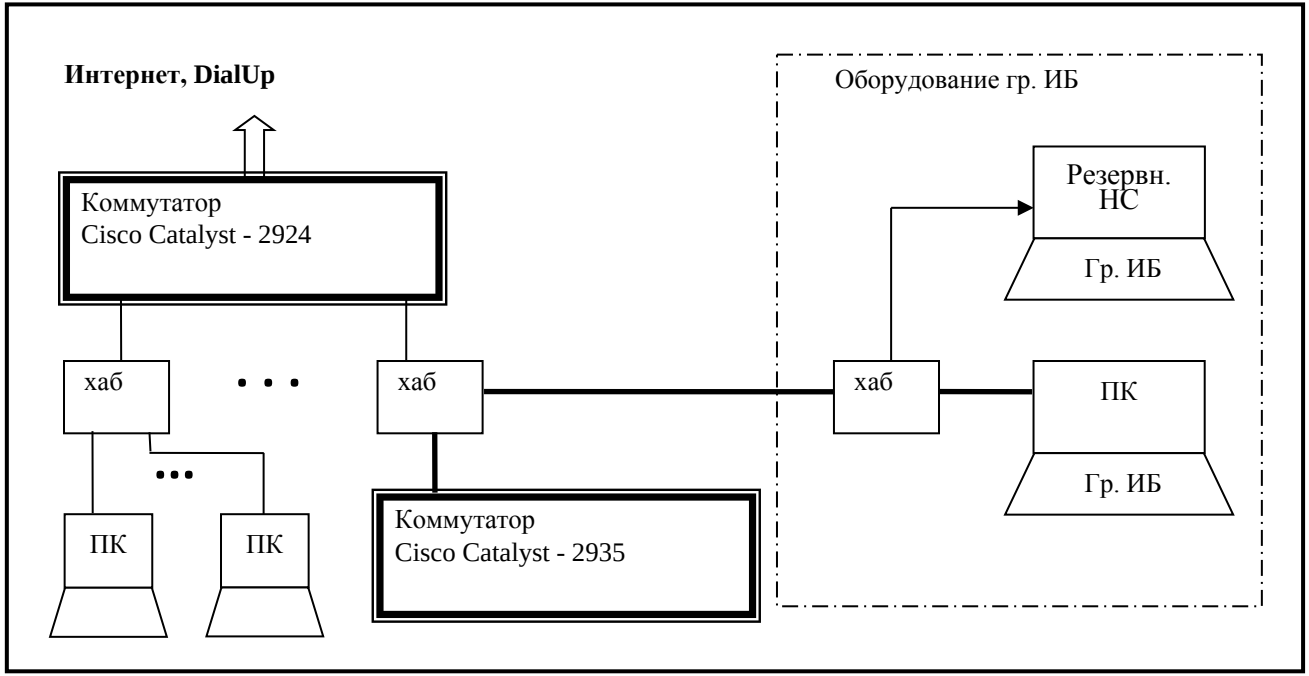


Рис. 8. Схема эксперимента связи с Cisco Catalyst-2935

```
   : 55AA 3C00   Offs= 0 (00000000h)   Pkt= 1   L= 60
     FFFFFFFFFFFF 004005411AAF 0806=E2-ARP
     00 01 08 00 06 04 00 01 00 40 05 41 1A AF C1 E8   .........@.A.ЇБи
     C2 0C 00 00 00 00 00 00 C1 E8 C2 19 00 00 00 00   В.......БиВ.....   00 00
00 00 00 00 00 00 00 00 00 00 00 00          ..............
   : 55AA 3C00   Offs= 64 (00000040h)   Pkt= 2   L= 60
     004005411AAF 000DBCBBC780 0806=E2-ARP
     00 01 08 00 06 04 00 02 00 0D BC BB C7 80 C1 E8   ..........j»ЗЂБи
     C2 19 00 40 05 41 1A AF C1 E8 C2 0C 00 00 00 00   В..@.A.ЇБиВ.....
     00 00 00 00 00 00 00 00 00 00 00 00 00 00         ..............
   : 55AA 3E00   Offs= 128 (00000080h)   Pkt= 3   L= 62
     000DBCBBC780 004005411AAF 0800=E2-IP 193.232.194.025<193.232.194.012:TCP
45 00 00 30 B4 5E 40 00 80 06 3E 72 C1 E8 C2 0C   E..0ґ^@.Ъ.>rБиВ.
     C1 E8 C2 19 0E 1F 00 17 08 E1 2D 20 00 00 00 00   БиВ......б- ....
     70 02 40 00 F6 F0 00 00 02 04 05 B4 01 01 04 02   р.@.цр.....ґ....
   : 55AA 3C00   Offs= 194 (000000C2h)   Pkt= 4   L= 60
     004005411AAF 000DBCBBC780 0800=E2-IP 193.232.194.012<193.232.194.025:TCP
     45 00 00 2C 00 00 00 00 FF 06 B3 D4 C1 E8 C2 19   E..,......iФБиВ.
     C1 E8 C2 0C 00 17 0E 1F 97 1D 77 B6 08 E1 2D 21   БиВ.....—.w¶.б-!
     60 12 10 20 2C F3 00 00 02 04 05 B4 00 00         `.. ,у.....ґ..
   : 55AA 3C00   Offs= 258 (00000102h)   Pkt= 5   L= 60
     000DBCBBC780 004005411AAF 0800=E2-IP 193.232.194.025<193.232.194.012:TCP
     45 00 00 28 B4 5F 40 00 80 06 3E 79 C1 E8 C2 0C   E..(ґ_@.Ъ.>уБиВ.
     C1 E8 C2 19 0E 1F 00 17 08 E1 2D 21 97 1D 77 B7   БиВ......б-!—.w·
```

```
     50 10 44 70 10 60 00 00 00 00 00 00 00 00         P.Dp.`........
   : 55AA 4200   Offs= 322 (00000142h)   Pkt= 6   L= 66
     004005411AAF 000DBCBBC780 0800=E2-IP 193.232.194.012<193.232.194.025:TCP
     45 C0 00 34 00 01 00 00 FF 06 B3 0B C1 E8 C2 19   EA.4......i.БиВ.
     C1 E8 C2 0C 00 17 0E 1F 97 1D 77 B7 08 E1 2D 21   БиВ.....—.w·.б-!
     50 18 10 20 31 80 00 00 FF FB 01 FF FB 03 FF FD   P.. 1Ъ...ы..ы..э
     18 FF FD 1F                                       ..э.
   : 55AA 4200   Offs= 392 (00000188h)   Pkt= 7   L= 66   000DBCBBC780
004005411AAF 0800=E2-IP 193.232.194.025<193.232.194.012:TCP
     45 00 00 34 B4 60 40 00 80 06 3E 6C C1 E8 C2 0C   E..4ґ`@.Ъ.>lБиВ.
     C1 E8 C2 19 0E 1F 00 17 08 E1 2D 21 97 1D 77 C3   БиВ......б-!—.wГ
     50 18 44 64 FC 2D 00 00 FF FE 01 FF FD 03 FF FC   P.Ddь-...ю..э..ь
     18 FF FC 1F                                       ..ь.
   : 55AA 6000   Offs= 462 (000001CEh)   Pkt= 8   L= 96
     004005411AAF 000DBCBBC780 0800=E2-IP 193.232.194.012<193.232.194.025:TCP
     45 C0 00 52 00 02 00 00 FF 06 B2 EC C1 E8 C2 19   EA.R......IмБиВ.
     C1 E8 C2 0C 00 17 0E 1F 97 1D 77 C3 08 E1 2D 2D   БиВ.....—.wГ.б--
     50 18 10 14 B1 ED 00 00 0D 0A 0D 0A 55 73 65 72   P...±н......User
     20 41 63 63 65 73 73 20 56 65 72 69 66 69 63 61    Access Verifica
     74 69 6F 6E 0D 0A 0D 0A 55 73 65 72 6E 61 6D 65   tion....Username
     3A 20                                             :
```

Рис. 9. Формат первых 8 пакетов связи с Cisco Catalyst

По каждому пакету была проанализирована информационная часть и составлен список опций протокола Telnet, реально используемых в сеансах связи с коммутатором Cisco Catalyst-2935. Так, например, на рис. 10 показан отдельно пакет №6 рисунка 9, реструктурированный в соответствии с правилами TCP/IP [3]. Подчеркнуты 4 группы Telnet-команд, в спецификациях протокола Telnet именуемые ***WILL Echo***, ***WILL SuppressGA***, ***DO Terminal-Type*** и ***DO Window-Size***.

```
: 55AA 4200   Offs= 322 (00000142h)   Pkt= 6   L= 66
004005411AAF 000DBCBBC780 0800  Ethernet-заголовок
 45 C0 00 34 00 01                        IP-заголовок
 00 00 FF 06 B3 0B C1 E8 C2 19 C1 E8
 C2 0C 00 17 0E 1F 97 1D 77 B7 08 E1   TCP-заголовок
 2D 21 50 18 10 20 31 80 00 00
 FF FB 01 FF FB 03 FF FD 18 FF FD 1F   TCP-тело
```

Рис. 10. Формат пакета 6 связи с Cisco Catalyst

Источник	16-вид	Команда
Сервер	**FF FB 01**	**WILL Echo**
Сервер	**FF FB 03**	**WILL Suppress GA**
Сервер	**FF FD 18**	**DO Terminal-Type**
Сервер	**FF FD 1F**	**DO Window-Size**
Клиент	**FF FD 01**	**DO Echo**
Клиент	**FF FD 03**	**DO Suppress GA**
Клиент	**FF FB 18**	**WILL Terminal-Type**
Клиент	**FF FC 1F**	**WON'T Window-Size**
Сервер	**FF FA 18 01 FF F0**	***Сообщить тип терминала***
Сервер	**FF FE 1F**	**DON'T Window-Size**
Клиент	**FF FA 18 00 41 4E 53 49 FF F0**	***Тип Терминала*** **= ANSI**
Клиент	**FF FC 1F**	**WON'T Windows-Size**

Рис. 11. Список используемых Terminal-опций в Telnet

Описанный эксперимент позволил установить начальный список опций, заведомо используемых в Telnet-протоколе связи с коммутатором Catalyst-2935 (рис. 11). Дальнейшие эксперименты показали, что этот список окончательный, хотя порядок опций в

отношении основного информационного потока может варьироваться в зависимости от сетевой коннективности.

Следует с сожалением отметить, что, как показали специальные эксперименты, реализация даже указанных команд в IOS Cisco неполна. Так, попытка применения ответной опции **DON'T Echo** на стороне клиента вовсе не подавляет последующие эхо-ответы сервера, как должно следовать из [27].

Реализующее программное обеспечение было создано в среде Windows на языке высокого уровня с использованием компилятора **PowerBASIC for Windows 7.04**. Использованы немодальные (modeless) диалоги, позволившие выполнить разработку программы в стиле реального времени (realtime). Общий вид окна программы приведен на рис. 12.

В этой, начальной версии программы предусмотрено выполнение всего одной содержательной команды Cisco Catalyst, которая должна быть параметром вызова программы. После ее исполнения следует автоматическое исполнение команды **EXIT** и связь с Cisco Catalyst разрывается. Однако, команда может быть выполнена несколько раз по инициативе оператора (нажатие кнопки **Start**).

Тем самым воспроизведены все необходимые условия создания настраиваемого программного обеспечения: конкретная исполняющая команда может быть задана параметром вызова программы. В дальнейшем предполагалось дать возможность вместо конкретной команды задать параметром каталог шаблонов возможных сценариев.

На рис. 13 показан журнал работы, сформированный программой при исполнении команды **DIR**. Символы пароля заменены звездочками. Каждая строка соответствует одной транзакции (посылке или приему блока информации).

На приведенном рисунке после даты и времени указано направление транзакций: стрелка влево показывает символы, передающиеся на управляемый объект, стрелка вправо показывает принимаемые символы. Видно, как формируются транзакции при работе протокола: из программы управляющие символы подаются по одному, исключая концы строк и опциональные блоки.

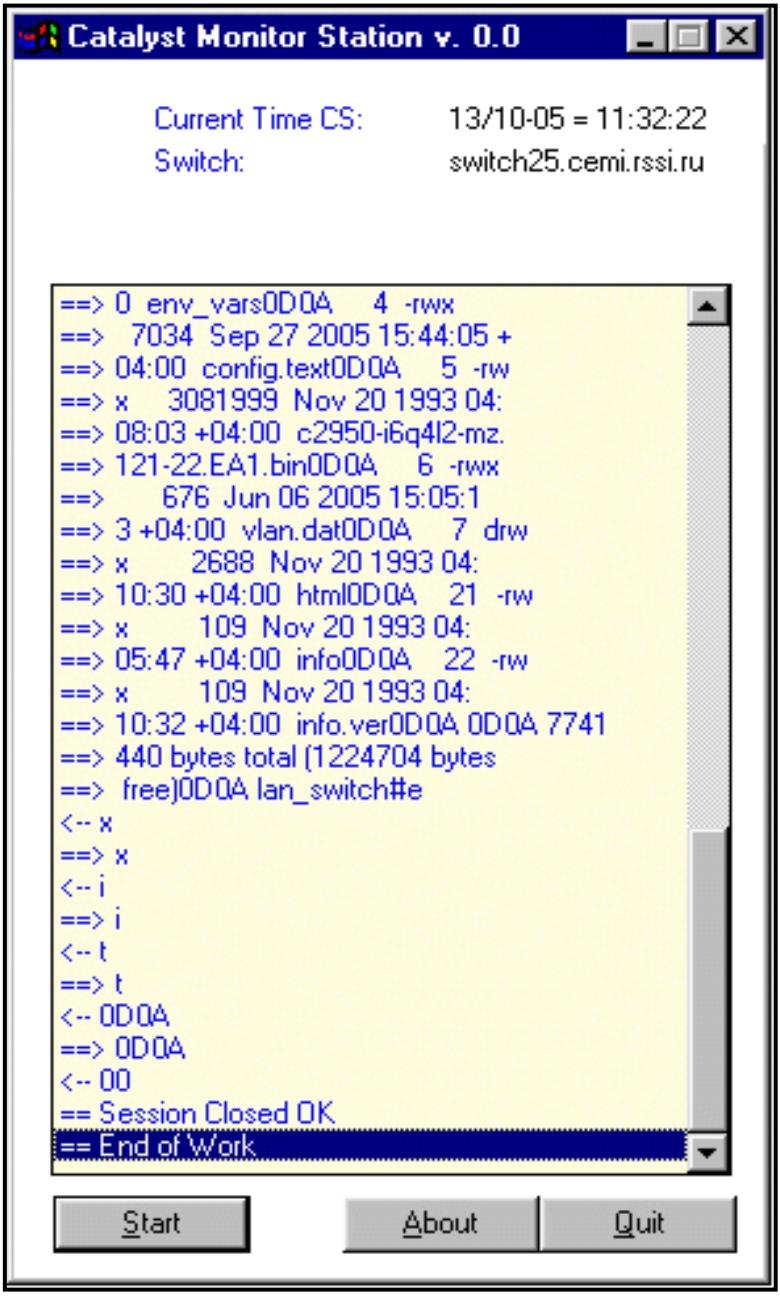


Рис. 12. Вид окна тестовой программы управления Cisco Catalyst

```
 A.Trenty = 17:40:04 |== Begin of work
 17/10-05 = 17:40:06 |== Session Opened
 17/10-05 = 17:40:06 |==> FFFB01 FFFB03 FFFD18
FFFD1F
 17/10-05 = 17:40:06 |<-- FFFD01 FFFD03 FFFB18
FFFC1F
 17/10-05 = 17:40:06 |==> 0D0A 0D0A User Access
Verification0D0A
 17/10-05 = 17:40:06 |==> 0D0A Username:
FFFA1801FFF0
 17/10-05 = 17:40:06 |<-- FFFA1800414E5349FFF0
 17/10-05 = 17:40:06 |<-- a
 17/10-05 = 17:40:06 |==> FFFE1F
 17/10-05 = 17:40:06 |<-- FFFC1F
 17/10-05 = 17:40:06 |==> a
 17/10-05 = 17:40:06 |<-- t
 17/10-05 = 17:40:06 |==> t
 17/10-05 = 17:40:06 |<-- -
 17/10-05 = 17:40:06 |==> -
 17/10-05 = 17:40:06 |<-- 9
 17/10-05 = 17:40:06 |==> 9
 17/10-05 = 17:40:06 |<-- 0
 17/10-05 = 17:40:06 |==> 0
 17/10-05 = 17:40:06 |<-- 4
 17/10-05 = 17:40:06 |==> 4
 17/10-05 = 17:40:06 |<-- 0D0A
 17/10-05 = 17:40:06 |==> 0D0A Password:
 17/10-05 = 17:40:06 |<-- *
 17/10-05 = 17:40:06 |<-- *
 17/10-05 = 17:40:06 |<-- *
 17/10-05 = 17:40:06 |<-- *
 17/10-05 = 17:40:07 |<-- *
 17/10-05 = 17:40:07 |<-- *
 17/10-05 = 17:40:07 |<-- 0D0A
 17/10-05 = 17:40:07 |==> 0D0A lan_switch#
 17/10-05 = 17:40:07 |<-- d
 17/10-05 = 17:40:07 |==> d
 17/10-05 = 17:40:07 |<-- i
 17/10-05 = 17:40:07 |==> i
 17/10-05 = 17:40:07 |<-- r
 17/10-05 = 17:40:07 |==> r
```

```
17/10-05 = 17:40:07 |<-- 0D0A
17/10-05 = 17:40:07 |==> 0D0A
17/10-05 = 17:40:07 |==> Directory of flash:/0D0A 0D0A
91
17/10-05 = 17:40:07 |==>  -rwx           5  Sep 27 2005
17/10-05 = 17:40:07 |==>  15:44:05 +04:00  private-conf
17/10-05 = 17:40:07 |==> ig.text0D0A     3  -rwx       2
17/10-05 = 17:40:07 |==> 70  Jan 01 1970 04:01:23 +04:0
17/10-05 = 17:40:07 |==> 0  env_vars0D0A     4  -rwx
17/10-05 = 17:40:07 |==>   7034  Sep 27 2005 15:44:05 +
17/10-05 = 17:40:07 |==> 04:00  config.text0D0A   5  -rw
17/10-05 = 17:40:07 |==> x     3081999  Nov 20 1993 04:
17/10-05 = 17:40:07 |==> 08:03 +04:00  c2950-i6q412-mz.
17/10-05 = 17:40:07 |==> 121-22.EA1.bin0D0A   6  -rwx
17/10-05 = 17:40:07 |==>       676  Jun 06 2005 15:05:1
17/10-05 = 17:40:07 |==> 3 +04:00  vlan.dat0D0A   7  drw
17/10-05 = 17:40:07 |==> x        2688  Nov 20 1993 04:
17/10-05 = 17:40:07 |==> 10:30 +04:00  html0D0A  21  -rw
17/10-05 = 17:40:07 |==> x         109  Nov 20 1993 04:
17/10-05 = 17:40:07 |==> 05:47 +04:00  info0D0A  22  -rw
17/10-05 = 17:40:07 |==> x         109  Nov 20 1993 04:
17/10-05 = 17:40:07 |==> 10:32 +04:00  info.ver0D0A 0D0A
7741
17/10-05 = 17:40:07 |==> 440 bytes total (1224704 bytes
17/10-05 = 17:40:12 |==>  free)0D0A lan_switch#
17/10-05 = 17:40:12 |<-- e17/10-05 = 17:40:12 |==> e
17/10-05 = 17:40:12 |<-- x
17/10-05 = 17:40:12 |==> x
17/10-05 = 17:40:12 |<-- i
17/10-05 = 17:40:12 |==> i
17/10-05 = 17:40:12 |<-- t
17/10-05 = 17:40:12 |==> t
17/10-05 = 17:40:12 |<-- 0D0A
17/10-05 = 17:40:12 |==> 0D0A
17/10-05 = 17:40:12 |... Normal End of Session
17/10-05 = 17:40:12 |== Session Closed OK
17/10-05 = 17:41:16 |== End of Work
```

Рис. 13. Журнал (протокол транзакций) исполнения команды DIR

Можно заметить (строки **17:40:07-17:40:12**), что вместо подавленной согласно опциям Telnet команды **GA** при завершении многострочного текстового вывода протоколом управляемого объекта выполняется задержка до установленного значения таймаута (в программе задано значение 5 с). Разумеется, это неудобно для практической реализации; при формировании эксплуатационной версии следует алгоритмически разрешить этот вопрос.

Конечно же, приведенный журнал невозможно использовать в практической работе с программой конечного пользователя. Он включает по сути отладочную информацию, в то время как информационная его часть не структурирована по строкам. Поэтому, вместе с журналом формируется протокол работы, пример которого для вышеприведенного случая команды дан на рис. 14.

```
User Access Verification

Username: at-904
Password:
lan_switch#dir
Directory of flash:/

   91  -rwx           5  Sep 27 2005 15:44:05
+04:00  private-config.text
    3  -rwx         270  Jan 01 1970 04:01:23
+04:00  env_vars
    4  -rwx        7034  Sep 27 2005 15:44:05
+04:00  config.text
    5  -rwx     3081999  Nov 20 1993 04:08:03
+04:00  c2950-i6q4l2-mz.121-22.EA1.bin
    6  -rwx         676  Jun 06 2005 15:05:13
+04:00  vlan.dat
    7  drwx        2688  Nov 20 1993 04:10:30
+04:00  html
   21  -rwx         109  Nov 20 1993 04:05:47
+04:00  info
   22  -rwx         109  Nov 20 1993 04:10:32
+04:00  info.ver

7741440 bytes total (1224704 bytes free)
lan_switch#exit
```

Рис. 14. Протокол пользователя - исполнение команды DIR

Приведенный протокол работы полностью соответствует аналогичному протоколу, получаемому при использовании команды **DIR** стандартной программой **telnet.exe**, входящей в комплект программ Windows. Тем самым можно считать исполнение результирующего модуля функционально успешным.

Особую трудность в реализации протокола Telnet вызвала алгоритмизация и программная обработка многострочного вывода в случае, когда формируемый вывод приостанавливается для прочтения заполненного "стандартного" экрана. В этом случае, оказывается, выводится отдельной строкой последовательность симолов " **–More—**", затем 9 символов **BS**[9], затем 8 символов пробела и еще 9 символов **BS**. Далее этот цикл может быть повторен. Предполагается, что на стандартном ANSI-терминале получается мигающая надпись. Два фрагмента такого вывода показаны в [58], причем выяснилось, что указанные символьные комбинации могут быть в физически разных транзакциях ввода (символы **BS** показаны квадратами).

Указанная трудность успешно преодолена следующим образом. Оказалось достаточным распознавать в транзакции ввода последовательность "**--More--**", и подавать на вывод дополнительный пробел как сигнал продолжения многострочного вывода. В то же время, из строк формируемого протокола изымается часть согласно количеству принятых символов **BS**. Полученный результат адекватен «чистому» протоколу вывода.

Проведенные эксперименты после завершения отладки реализующей программы проверены на коммутаторах Cisco Catalyst-2935 с IOS C2900XL Software 12.1(22)EA1, работающем в холостом режиме, и Cisco Catalyst-2924 с IOS 12.0(5.2)XU, работающем в стандартном режиме эксплуатации. Результаты адекватны, помех в работе коммутаторов не обнаружено.

Таким образом, проведенные исследования показали принципиальную возможность создания программных средств полуавтоматического и автоматического управления сетевыми коммутаторами Cisco Catalyst по протоколу Telnet. Эти программные средства работоспособны в реальном времени как Windows-приложения. Выполненная реализация таковых средств в начальном варианте проверена на актуальных сетевых коммутаторах Cisco Catalyst-2924 и Cisco Catalyst-2935 и показала результаты, адекватные стандартным методам управления. Далее было необходимо реализовать аппарат типовых сценариев, а на более поздних этапах и связь со сторонними управляющими средствами.

[9] Backspace.

Предложенный автором подход управления сетевыми коммутаторами открывает возможность создания нового варианта типовой технологии управления локальной вычислительной сетью, доступной в эксплуатации пользователям средней квалификации, а также интеграции средств управления коммутаторами в программный комплекс управления локальной сетью предприятия/организации.

1.3. Синтез языка управления работой сетевых коммутаторов Cisco

Как было показано в предыдущем разделе, в ходе разработки средств автоматизированного управления сетевыми коммутаторами Catalyst, в рамках построения заявленной системы сетевого мониторинга и аудита, было обеспечено автоматическое установление TelNet-сеанса связи с коммутатором, исполнение заранее написанной последовательности команд и корректное завершение сеанса.

Заявленной целью автоматизированного управления сетевыми коммутаторами на начальных этапах проектирования системы сетевого мониторинга и аудита была необходимость оперативно, автоматически реагировать на нарушения сетевыми объектами правил поведения в сети. Однако, уже в то время ожидалось, что подобные средства весьма удобно эксплуатировать и в полуавтоматическом режиме для оперативного управления коммутатором: подключения новых пользователей, выдачи различной справочной информации о состоянии коммутатора и т.п.

На том этапе работы представлялось, что достаточным решением проблемы автоматизации станет организация заранее подготовленных и тщательно выверенных блоков (последовательностей) команд управления с возможностью расширения их макросредствами для подключения меняющихся параметров. Так, для временного прекращения доступа некоторого сетевого устройства к ЛВС казалось достаточным подать следующие команды[10] (см. рис. 15):

[10] Разумеется, после установления связи и успешного прохождения аутентификации. Во всех примерах ниже, где явно не указано иное, это предполагается выполненным.

```
config terminal
mac address-table static &M vlan &V drop
exit
exit
```

Рис. 15. Простая процедура временного запрета по Static-списку

в которых подчеркнутые значения, служащие макровызовами, программно заменить на актуальные перед передачей команды коммутатору. Актуальные же значения предполагалось получить из других частей программного комплекса (например, в качестве информации от наблюдающей станции сетевого мониторинга) или обеспечить их ввод оператором при ручном вызове процедуры перед её стартом.

Однако, тщательный анализ показывает, что простое объединение в пакет последовательности команд управления коммутатором ещё не может отвечать целям эффективного управления.

В самом деле, во-первых, каждая конкретная команда может быть отклонена коммутатором, причём не только по причине неправильного синтаксиса (это легко устраняется предварительной отладкой), но также потому, что в данной конфигурации команда невыполнима. К примеру, попытка добавить дополнительную строку к списку Secure некоторого интерфейса даже при верном синтаксисе команды не будет выполнена, если предельное количество строк для данного интерфейса уже задействовано. Иными словами, требуется анализ корректного завершения поданных команд или, если угодно, анализ наличия/отсутствия ошибки в поданной команде.

Немаловажным также является то, что кажущаяся алгоритмическая детерминированность управления на самом деле представляет собой реальный процесс в сетевой среде, который может по тем или иным причинам быть прерванным. Результаты такого прерванного сеанса можно установить только специальными запросами о состоянии управляемого объекта, так что эти запросы и анализ ответной информации следует предусмотреть до выдачи управляющих команд.

Далее, оказалось, что в ряде случаев для выдачи верной команды управления требуется дополнительная информация, которая не может быть априори известна. Так, в вышеприведенном примере после получения второй команды коммутатор может сообщить, что указанный MAC-адрес не является статическим (STATIC), а принадлежит, к примеру, некоторому интерфейсу с включенной Secure-функцией и, таким образом, является SECURE, а не STATIC. В этом случае следует выполнить совершенно другие команды управления для реального блокирования сетевого устройства. Для добавления же очередной строки к Secure-списку нужно получить актуальное количество строк списка, проверить его на предельное значение, увеличить на 1, задать новое значение и лишь после этого выполнить команду добавления нового устройства в список. Номер интерфейса и тем более количество строк в его Secure-списке легко узнаются соответствующими командами, но, как видим, необходим аппарат съема числовых и текстовых значений из получаемого от коммутатора потока информации, необходимы арифметические операции над числовыми переменными и операции сравнения как числовых, так и текстовых значений с возможностью разветвления последовательности подачи команд.

Более того, проектируемое средство управления отнюдь не является единственным управляющим клиентом. В ряде ситуаций вполне возможно вмешательство системных инженеров, например, после отключения подачи электропитания, которые посчитают необходимым подать на сетевой коммутатор «свои» управляющие команды, не проходящие через обсуждаемый интерфейс. Разумеется, все такие команды будут изменять конфигурацию коммутатора.

Практика работы показывает, что возможны и самопроизвольные перезапуски коммутатора с восстановлением ранее запомненной конфигурации. При экспериментах с сетевыми коммутаторами в ЦЭМИ РАН, оказалось возможным подобрать последовательность формально легальных команд на сетевом коммутаторе Cysco Catalyst 2950 (IOS 12.1), которая с некоторой вероятностью вызывала перезагрузку IOS.

Таким образом, запоминание текущей ситуации (например, номера интерфейса, к которому относится данный MAC-адрес) на стороне управляющего клиента не гарантирует однозначной корректности срабатывания управляющих процедур. Вся дополнительная информация в целях эффективного управления должна

быть запрошена предварительными командами перед выдачей целевых управляющих команд, снята и проанализирована по качественному составу и количественно. Именно эта особенность управления сетевыми коммутаторами и обуславливает привлечение высококвалифицированного технического персонала для практической работы.

Наконец, поскольку управление коммутатором с помощью ряда заранее подготовленных процедур предполагается к исполнению малоквалифицированными лицами или вообще осуществляется по инициативе автоматики, следует обеспечить возможность индикации однозначного результата заданной процедуры как выполненной либо не выполненной в целом. Вероятно, не помешает также возможность появления в протоколе доступных текстов как специфицирующих ситуацию.

Таким образом, проблема надежного управления коммутатором с помощью заранее написанных целевых управляющих последовательностей команд предстаёт проблемой дополнения управляющих команд средствами анализа корректности завершения исполнившейся команды, средствами съема актуальной информации во внутренние переменные, ряда операций с этими переменными, средствами разветвления процесса и иными.

Актуальной проблемой является создание языка организации процесса управления работой коммутаторов Cisco Catalyst на базе TelNet-команд или, иначе, скриптовой надстройки над TelNet-командами управления.

Настоящая работа посвящена опыту разработки такого языка и средств его исполнения, включенных в мониторную программу, представляющую часть программно-технического комплекса сетевого мониторинга.

1.3.1. Анализ объекта

Сетевые коммутаторы Cisco Catalyst, широко используемые в целях оперативного управления локальными сетями, в своём базовом математическом обеспечении всегда имеют поддержку протокола TelNet. Характеристические особенности этого протокола применительно к усечённому варианту, используемому в коммутаторах Cisco, исследованы в [12].

Применительно к поставленной проблеме создания языковой надстройки, объектом исследования выступает процесс выдачи управляющих команд и приёма ответной информации. Этот процесс обладает следующими основными особенностями и свойствами.

● С момента заявки на аутентификацию пользователя, процесс является неразрывным во времени. Превышение некоторого предела временной задержки перед подачей очередной команды (обычно 3 мин.) неминуемо ведёт к разрыву сеанса.

● Команды управления образованы буквами латинского алфавита с немногочисленными специальными символами и всегда начинаются с буквы. Терминальными символами окончания ввода являются символы конца строки (совокупность **0D0A**h) и вопросительного знака. Вместо одного пробела всегда можно указать несколько.

● При ожидании ввода команды всегда присутствует терминальное приглашение (промпт), начинающееся символами "**lan_switch**" и заканчивающееся диезом "**#**". Между этими группами могут находиться другие термы, идентифицирующие текущий уровень управления.

● Существует команда, последовательное повторение которой всегда выводит с любого места управляемый объект на начальный уровень управления ("**lan_switch#**"), а поданная на начальном уровне, эта команда приводит к корректному завершению сеанса работы.

● Существует единственный однозначный признак успешного завершения выполнения поданной команды – очередной промпт без промежуточно выданных строк. Однако, этот признак достаточен, но не необходим: выданная промежуточная информация может являться, а может и не являться признаком некорректности или невыполнимости команды. В то же время, появление строки, начинающейся с процента ("**%**"), однозначно свидетельствует об ошибке выполнения команды. В частности, все ошибки синтаксического уровня всегда вызывают появление ремарки, предваряемой процентом.

● Команды запроса информации влекут за собой вывод, в котором как числовые, так и текстовые величины могут быть форматированы как по правому, так и по левому краю.

Приведённых свойств процесса управления достаточно для того, чтобы предложить следующие особенности языковой среды.

■ Каждая команда языка управления долна занимать ровно 1 строку, начинаться с позиции 1 и заканчиваться символами конца строки

■ В отличие от прямых команд управления коммутатором, команды синтезируемого языка (в дальнейшем именуемые «псевдокоманды», или сокращенно ПСК) должны начинаться с небуквенных символов.

■ Целесообразно ввести типовой признак Ошибки выполнения команды, устанавливаемый в случаях появления в выходной информации коммутатора строк, начинающихся с символа «процент».

■ В связи с многовариантностью используемых при выдаче средств форматирования, создаваемый язык должен давать возможность определять контекстное присутствие нужной группы символов вне зависимости от точной позиции в строке выдачи. При съёме числовых показателей нужно учесть возможность лидирующих пробелов и прерывания распознавания числа по первому нецифровому символу.

Многоуровневость управления неминуемо влечёт за собой необходимость последовательной проверки как минимум двух различных числовых значений. Проверка или съём каждого требует максимум двух текстовых и одной числовой переменной. Таким образом, учитывая необходимость хранить результаты одной предыдущей проверки, достаточно использовать 3 текстовых и 3 числовых переменных в языке для формирования любых, сколь угодно сложных алгоритмов проверки, плюс организации разветвления алгоритма по условию.

Определив особенности языковой среды, далее приступим к формальному изложению языковых блоков (процедур), семантики и синтаксиса предлагаемых языковых средств. При формальном изложении языка в данной работе считаются принятыми следующие соглашения.

• Полужирный прямой шрифт использован для обозначения элементов языка точно так, как они должны быть заданы в тексте процедур.

• Полужирный курсив использован для идентификации объектов, которым при написании процедур должны быть приданы конкретные значения.

• Квадратные скобки без жирности используются для указания необязательных элементов или конструкций.

• Фигурные скобки и вертикальная черта без жирности определяют необходимость выбора одного из нескольких указанных элементов.

1.3.2. Организация процедур

Единый командный файл, выполняемый за один сеанс обращения к коммутатору, будем называть **процедурой**. Каждая процедура должна иметь уникальное среди прочих имя. Все процедуры сохраняются в конце конфигурационного файла после последнего конфигурационного оператора, либо отдельным файлом.

Процедура состоит из заголовка и остальных строк, которыми могут быть исполняемые TelNet-команды, псевдокоманды скрипта (сокращённо «ПСК») и комментарии. Строки набираются в обычном текстовом редакторе. Каждая команда или ПСК начинается с первой позиции новой строки и должна умещаться в одну строку. Пустые строки не допускаются. Все процедуры записываются подряд одна за другой, начало следующей означает конец предыдущей. Концом всех процедур является конец конфигурационного файла мониторной программы.

В данной реализации языка не предусмотрена возможность комментариев после содержательной части строки; комментарии должны быть отдельными строками. Таким образом, например, в псевдокоманде **:Метка** именем метки считается всё после двоеточия до конца строки, включая пробелы. За корректностью идентификаторов меток следует следить программисту.

1.3.3. Семантика

В данной опытной версии языка может использоваться 9 переменных.

Переменная V считается предопределенной и отражает vlan. Её значение определяется специальным оператором конфигурационного файла и не может изменяться. При появлении в тексте

исполняемой команды макровызова &V, перед исполнением команды такая конструкция заменяется на номер vlan.

Переменные M и I также предопределены в каждой процедуре и не могут быть изменены внутри них. Они предназначены для определения MAC-адреса интересуемого объекта и интерфейса (порта коммутатора). Вставка конкретных значений в текст исполняемых команд осуществляется по макровызовам &M и &I. По &M вставляется MAC-адрес в принятой в командах управления коммутаторами форме (3 группы 16-ричных символов через две точки), по &I вставляется одно- или двузначное целое число. Если процедура использует входными параметрами MAC-адрес и/или интерфейс, то соответствующие символы должны присутствовать в её заголовке, тогда эти параметры будут запрошены от оператора перед исполнением процедуры.

Переменные X, Y и Z являются свободными числовыми переменными. Их значения могут устанавливаться и меняться внутри процедуры. Перед началом процедуры значения переменных не определены. Существует специальная псевдокоманда проверки, установлено ли значение переменной. Для включения значения переменной в строку исполняемой команды или комментария следует использовать макровызов ─ знак амперсенда и имя переменной. При попытке использования значения неустановленной переменной, даже в !-комментарии, исполнение скрипта блокируется.

Единственным способом установки значения числовой переменной является приём этого значения из выходного потока коммутатора с помощью оператора #. В дальнейшем можно увеличить или уменьшить на 1 это значение специальными псевдокомандами.

Переменные U, W и T являются свободными текстовыми переменными. Их значения могут устанавливаться и меняться внутри процедуры. Перед началом процедуры значения переменных не определены. Существует специальная псевдокоманда проверки, установлено ли значение переменной. При попытке использования значения неустановленной переменной, даже в !-комментарии, исполнение скрипта блокируется.

Единственным способом установки значения текстовой переменной является приём этого значения из выходного потока коммутатора с помощью оператора $.

1.3.4. Синтаксис

[ИмяПр] [M] [I] [{D}] — строка-заголовок новой процедуры. ИмяПр обязано быть уникальным среди прочих. Команды и псевдокоманды следуют строками далее. Конец процедуры определяется по началу следующей процедуры или концу конфигурационного файла. Признаки M и I обозначают необходимость указания вручную соответственно MAC-адреса и интерфейса для исполнения процедуры. Признак D должен быть задан ровно в одной процедуре и отмечает процедуру, которая должна автоматически вызываться для блокировки (drop) ПК по данным наблюдающей станции.

'Комментарий — строка-комментарий для создателя процедур. Всё содержимое строки, начинающейся с одинарной кавычки, интерпретатором не анализируется и осуществляется переход к следующей строке. Состояние флагов не изменяется.

!Комментарий — строка-комментарий для пользователя. Появление этой строки вызывает включение комментария в вывод TelNet-сеанса и общий протокол работы мониторной программы. Таким образом, данный оператор является исполняемым.

Исполняемая-команда — строка, передаваемая на исполнение коммутатору. Опознается по отсутствию в первой позиции одного из специальных символов, означающих псевдокоманду. Может содержать макровызовы любой из переменных, однако они должны быть определены. Перед передачей исполняемой команды коммутатору, интерпретатор сбрасывает флаг Ошибки. Завершение исполнения исполняемой команды автоматически сбрасывает все заданные псевдокомандами # и $ требования на контекстный поиск и установку значений переменных. В случае, если выданный по данной команде коммутатором текст содержал хоть одну строку, начинающуюся с символа %, устанавливается флаг Ошибки.

:Метка — псевдокоманда, определяющая метку в процедуре. При прохождении в порядке следования пропускается, флагов интерпретатора не изменяет. Возникшая необходимость перехода на метку вызывает просмотр операторов до конца процедуры с поиском данного оператора с нужной меткой (см. пояснение в предыдущем подразделе «Семантика»). При нахождении исполняется следующий после найденной метки оператор. Стало быть, в скрипте допускаются дублирующие метки; целесообразность их использования определяет программист.

>Метка — псевдокоманда немедленной передачи управления на указанную метку. Метка ищется ниже по тексту процедуры.

%Метка — псевдокоманда условной передачи управления на указанную метку. Метка ищется ниже по тексту процедуры. Управление будет передано, если флаг Ошибки установлен. Данная ПСК не сбрасывает флаг Ошибки.

#"Образец"Имя<Начало — псевдокоманда задания режима поиска среди выводных строк коммутатора строки, содержащей Образец. При нахождении таковой, числовой переменной Имя будет присвоено значение, содержащееся в найденной строке, начиная с позиции Начало, до первого нецифрового символа или до конца строки. Лидирующие пробелы отбрасываются, однако пробел после первого же цифрового символа уже считается концом числа. В случае, если Начало больше длины строки, в которой найден Образец, исполнение процедуры прекращается с аварийным сообщением. Для выяснения, произошло ли реальное присвоение, следует воспользоваться оператором ?Имя%>Метка.

$"Образец"Имя<Начало,Длина — псевдокоманда задания режима поиска среди выводных строк коммутатора строки, содержащей Образец. При нахождении таковой, текстовой переменной Имя будет присвоено значение, содержащееся в найденной строке, начиная с позиции Начало, числом позиций Длина. В случае, если Начало больше длины строки, в которой найден Образец, либо Начало+Длина выходит за пределы строки, исполнение процедуры прекращается с аварийным сообщением. Для выяснения, произошло ли реальное присвоение, следует воспользоваться оператором ?Имя%>Метка.

?Имя%>Метка — псевдокоманда проверки, имеет ли переменная Имя значение. При отсутствии значения осуществляется переход на указанную метку. Оператор следует использовать только для свободных (см. «Семантика») переменных.

?Имя=Константа>Метка — псевдокоманда проверки соответствия значения числовой переменной Имя и числовой Константы. Переход на метку осуществляется при равенстве. Если переменная Имя не определена, дальнейшее исполнение процедуры блокируется.

?Имя="Константа">Метка — псевдокоманда проверки соответствия значения текстовой переменной Имя и текстовой Константы. Переход на метку осуществляется при точном совпадении.

Если переменная Имя не определена, дальнейшее исполнение процедуры блокируется.

=Имя+ — псевдокоманда увеличения значения числовой переменной на 1. Если переменная Имя не определена, дальнейшее исполнение процедуры блокируется.

=Имя- — псевдокоманда уменьшения значения числовой переменной на 1. Если переменная Имя не определена, дальнейшее исполнение процедуры блокируется.

^+ — псевдокоманда успешного завершения процедуры. Исполнение процедуры прекращается, индицируется успешный результат. При отсутствии данного оператора в потоке исполнения и прекращении работы процедуры по иным причинам, результат процедуры не считается успешным.

^- — псевдокоманда аварийного завершения процедуры. Исполнение процедуры прекращается, результат не достигнут. Данная ПСК не является обязательной для индикации неуспеха и служит только для эффективности написания процедур.

1.3.5. Результаты

Рассмотренная версия языковых средств полностью реализована в расширенной тестовой программе управления коммутатором TAMCICON и впоследствии вставлена в Мониторную программу, исполняемую на ПК-сателлите в общей схеме сетевого мониторинга ЛВС ЦЭМИ РАН, описанной в [8].

Даже простейшие применения скрипта дают возможность легко осуществить многошаговые процедуры с приёмом значений из вывода, формируемого коммутатором. Пример простой процедуры исключения MAC-адреса из Secure-списка приведен на рис. 16.

Как видим, в простой процедуре удаления используется 12 строк, составляющих заголовок процедуры, строку комментария и 10 команд, среди которых 7 исполняемых и 3 ПСК.

Первая ПСК задаёт в дальнейшем выводе необходимость искать строку с текстом “**Maximum**”, причём при нахождении таковой строки присвоить переменной X числовое значение, находящееся в той же строке и записанное с позиции 30.

Далее следует команда show, вызывающая вывод ряда строк, среди которых и будет искаться строка с текстом “**Maximum**”.

```
[Del-Secure-Simple]MI
' The procedure tested 05.10.2006
#"Maximum"X<30
show port-security interface fastethernet 0/&I
=X-
config terminal
interface fastethernet 0/&I
no switchport port-security mac-address &M
switchport port-security max &X
exit
exit
^+
```

Рис. 16. Пример простой процедуры удаления MAC-адреса из Secure-списка

Далее следует ПСК, уменьшающая значение переменной X на 1. Строго говоря, перед этим неплохо было бы проверить, получила ли эта переменная значение, и является ли оно большим чем 1 (нельзя задать значение 0), но оставим эти замечания для окончательного варианта процедуры, отметив только, что языковые возможности ПСК позволяют всё это сделать.

Далее следует 6 команд прямого управления, осуществляющие нужную конфигурацию: вход в управление нужным портом, исключение из списка Secure нужного MAC-адреса, задание нового значения максимального числа Secure-адресов на этом порту и выходы на исходный уровень управления. Вообще-то перед подобными операциями порт необходимо перевести в состояние Shutdown, а после них – опять в активный режим, но эти замечания также оставим для конечного варианта процедуры.

Команды login и завершающая команда exit не задаются, они считаются присутствующими по умолчанию. Интерпретирующий блок мониторной программы при связи с коммутатором их вставляет всегда.

Обратим внимание, что в тексте команд процедуры встречаются макровызовы &M, &I и &X. Как следует из заголовка процедуры, переменные M и I являются её параметрами и, стало быть, будут запрошены от оператора до старта процедуры. Переменная же X определяется в ходе исполнения самой процедуры. Стало быть, все макровызовы перед подачей команды на коммутатор будут заменены на реальные значения.

```
  User Access Verification
  Username: at-904
  Password:
  !05/10-06 = 03:51:54+ Find on <Maximum>  X= < 30
  lan_switch#show    port-security    interface
fastethernet 0/7
  Port Security              : Enabled
  Port Status                : Secure-up
  Violation Mode             : Restrict
  Aging Time                 : 0 mins
  Aging Type                 : Absolute
  SecureStatic Address Aging : Disabled
  Maximum MAC Addresses      : 20
  !05/10-06 = 03:52:01+ Assigned X= 20
  Total MAC Addresses        : 20
  Configured MAC Addresses   : 20
  Sticky MAC Addresses       : 0
  Last Source Address        : 0004.00e4.1a62
  Security Violation Count   : 3592
  !05/10-06 = 03:52:06+ Evaluated X= 19
  lan_switch#config terminal
  Enter configuration commands, one per line.
End with CNTL/Z.

  lan_switch(config)#interface fastethernet 0/7
  lan_switch(config-if)#no   switchport   port-
security mac-address 0000.0000.1302
  lan_switch(config-if)#switchport        port-
security max  19
  lan_switch(config-if)#exit
  lan_switch(config)#exit
  lan_switch#exit
```

Рис. 17. Протокол коммутатора по процедуре удаления из Secure-списка

Протокол связи с коммутатором, соответствующий указанной процедуре, с вкраплениями интерпретирующего блока, дан на рис. 17. Для лучшего понимания, подаваемые коммутатору команды выделены жирностью, результат исполнения псевдокоманд — жирным курсивом.

```
   05/10-06 = 03:50:36  = Start LAN Monitor
Program, v.1.54
   05/10-06 = 03:50:36  Starting Log Level is set
to  2
   05/10-06 = 03:50:36  Configuration File Opened
   05/10-06 = 03:50:36   #|' Testing Configuration
File
   05/10-06 = 03:50:36   ||comm=2
   05/10-06 = 03:50:36   ||loglev=3
   05/10-06 = 03:50:36   ||asmaxc=1300
   05/10-06 = 03:50:36   ||udp=no
   05/10-06 = 03:50:36   ||udptime=3
   05/10-06 = 03:50:36
||switch=switch25.cemi.rssi.ru
   05/10-06 = 03:50:36   ||login=at-904
   05/10-06 = 03:50:36   ||vlan=194
   05/10-06 = 03:50:36   ||show=yes
   05/10-06 = 03:50:36   ||keepres=yes
   ..........
   05/10-06 = 03:50:36   ||[Del-Secure-Simple]MI
   05/10-06 = 03:50:36   #|' The procedure tested
05.10.2006
   05/10-06 = 03:50:36   ||#"Maximum"X<30
   05/10-06 = 03:50:36   ||show port-security
interfa fa 0/&I
   05/10-06 = 03:50:36   ||=X-
   05/10-06 = 03:50:36   ||config term
   05/10-06 = 03:50:36   ||interfa fa 0/&I
   05/10-06 = 03:50:36   ||no switchport port-sec
mac-addr &M
   05/10-06 = 03:50:36   ||switchport port-sec max
&X
   05/10-06 = 03:50:36   ||exit
   05/10-06 = 03:50:36   ||exit
   ..........
   05/10-06 = 03:50:36  Log Level is set to  3
   05/10-06 = 03:50:36  Thread Comm is succeffully
created
   05/10-06 = 03:50:38  COM2 opened successfully
   05/10-06 = 03:50:38  Thread TelNet is
succeffully created
   05/10-06 = 03:50:39  The Midnight is, sDay=<05>
   05/10-06 = 03:50:46  =Command Block <Del-Secure-
Simple>started
   05/10-06 = 03:51:53  == TelNet Session Opened
   05/10-06 = 03:52:26  :... Normal End of TelNet
Session
   05/10-06 = 03:52:26  == TelNet Session Closed OK
   05/10-06 = 03:54:28  Comm Thread closed
   05/10-06 = 03:54:29  TelNet Thread closed
   05/10-06 = 03:54:29  x End LAN Monitor Program
```

Рис. 18. Протокол исполнения мониторной программой процедуры рис. 16

Из протокола работы мониторной программы на рис. 18 легко видеть, что оператор начал вызов процедуры удаления и ввод параметров в 03:50:46, в 03:51:53 процедура стартовала, а в 03:52:26 уже была завершена. Таким образом, на исполнение процедуры ушло примерно полминуты. Указанный вызов был осуществлен дистанционно, через Dial-Up. Работа через обычный TelNet потребовала бы несколько минут вследствие ручного набора команд, а исполнение тех же шагов с помощью «современного» графического средства Claster Management Suite — порядка 15 минут, значительная доля которых ушла бы на подгрузку объёмных графических апплетов через медленный модем 33600.

```
 [Add-Secure-Simple]MI
' The procedure tested 05.10.2006
#"Maximum"X<30
show port-security interface fastethernet 0/&I
=X+
config term
interface fa 0/&I
switchport port-sec max &X
switchp port-s mac-addr &M
exit
exit
^+
```

Рис. 19. Пример простой процедуры пополнения Secure-списка

Обратная процедура пополнения Secure-списка приведена на рис. 19, а протокол вывода коммутатора при её исполнении — на рис. 20.

Как уже было отмечено выше, приведенные процедуры очень просты, не содержат проверок допустимости и результатов каждого шага, и приведены здесь только в целях иллюстрации практического применения созданной реализации скриптового языка. Реально используемые процедуры содержат больше операторов, причём весьма значительную (до 80%) их часть составляют именно псевдокоманды скрипта [59]. Поскольку ПСК интерпретируются блоками приложения и не предназначены для передачи через TelNet, их исполнение не вызывает заметного увеличения времени выполнения таких сложных процедур.

```
  Username: at-904
  Password:
  !05/10-06 = 03:24:18+ Find on <Maximum>  X= < 30
  lan_switch#show port-security interface
fastethernet 0/3
  Port Security              : Enabled
  Port Status                : Secure-up
  Violation Mode             : Restrict
  Aging Time                 : 0 mins
  Aging Type                 : Absolute
  SecureStatic Address Aging : Disabled
  Maximum MAC Addresses      : 14
  !05/10-06 = 03:24:26+ Assigned X= 14
  Total MAC Addresses        : 14
  Configured MAC Addresses   : 14
  Sticky MAC Addresses       : 0
  Last Source Address        : 0000.0000.0000
  Security Violation Count   : 0
  !05/10-06 = 03:24:31+ Evaluated X= 15
  lan_switch#config term
  Enter configuration commands, one per line.  End
with CNTL/Z.
  lan_switch(config)#interface fa 0/3
  lan_switch(config-if)#switchport port-sec max 15
  lan_switch(config-if)#switchport-s mac-addr
0000.3131.1313
  lan_switch(config-if)#exit
  lan_switch(config)#exit
  lan_switch#exit
```

Рис. 20. Результат исполнения простой процедуры рис. 19

1.3.6. Заключение по языковым средствам

Разработанные языковые средства и их программная интерпретация реализованы в версиях многофункциональной тестовой программы и впоследствии Мониторной программы, начиная с лета 2006г. При разработке этих программ, помимо автоматического вызова процедуры изоляции нужного сетевого устройства по его MAC-адресу, с самого начала была заложена возможность полуавтоматической работы процедур, т.е. их вызова Оператором.

Реальный процесс управления работой коммутатора, описанный в [8], оказался настолько простым и удачным по эффективности, что сразу же после появления работоспособной версии полнофункциональной тестовой программы **TAMCICON** с этими возможностями, только она и использовалась для оперативного управления действующим сетевым коммутатором. Иные средства более не применялись. Практически сразу было создано более 10 процедур на описанном языке, их каталог постоянно пополняется, а сами процедуры совершенствуются.

Таким образом, предложенный язык может служить основой и для не зависимых от системы сетевого мониторинга и аудита средств полуавтоматического управления сетевыми коммутаторами Cisco Catalyst. При этом, поскольку тексты процедур хранятся в обычном виде, блок процедур легко можно настроить на конкретные особенности IOS управляемого коммутатора.

1.4. Полнофункциональная программа TAMCICON управления коммутатором

Сетевые коммутаторы производства Cisco Catalyst обеспечивают множество функций управления пользовательскими и специальными сетевыми устройствами. Одним из наиболее часто используемых наборов функций является ряд функций управления списками подключения пользовательских сетевых устройств (рабочие станции, серверы, роутеры, сетевые принтеры и пр.). В практической работе наиболее часто востребованы такие функции этого ряда, как:

– подключение нового пользователя к нужному интерфейсу (порту) сетевого коммутатора;

– отключение имеющегося пользователя от питающего его порта;

– вывод текущего списка сетевых устройств, подключенных к интересуемому порту коммутатора;

– сохранение текущей конфигурации коммутатора в его энергонезависимой памяти в целях обеспечения рестарта при автоматическом или ручном перезапуске коммутатора;

– снятие справки о состоянии интересуемого порта коммутатора;

– снятие полной справки о текущей конфигурации коммутатора.

В зависимости от принятой в эксплуатирующей коммутатор организации системы информационной безопасности, конкретная реализация привязки сетевых устройств к портам коммутатора может быть выполнена различными способами, например, использованием статических списков MAC-адресов, либо использованием SECURE-списков. Конкретное исполнение указанных выше процедур, особенно подключения и отключения пользователей, в каждом из этих случаях исполняется совершенно различными командами управления.

В целях обеспечения функций управления, сетевые коммутаторы предоставляют веер управляющих средств, от тривиального протокола Telnet до громоздких комплексных пакетов типа Cisco Works. Однако, все эти средства требуют весьма высокой квалификации лица, осуществляющего функции управления; используют либо длинные цепочки труднозапоминаемых символьных команд (Telnet), либо весьма громоздкие (с задействованием Java VM и пр.) графические средства управления со сложным интерфейсом. Во всех случаях требуется хорошее знание предметной области, умение справляться с возможными нестандартными ситуациями (занят интерфейс / пользователь не на том порту, на котором предполагалось / исчерпан лимит списков и мн. др.), достаточно мощный управляющий компьютер с целым рядом установленных пакетов поддержки, повышенная требовательность к ошибкам набора и ввода команд. По имеющимся наблюдениям, даже системные администраторы с многолетним опытом работы, используя Telnet, не в состоянии сразу без ошибок набрать все требуемые команды управления, предпочитая набирать их по частям с использованием встроенного Help через ввод “?”.

В самом деле, включение, к примеру, некоего MAC-адреса 1234.5678.9abc в порт 16 коммутатора требует, как вариант, следующих действий – см. рис. 21. Вводимое Администратором выделено жирным, курсивные жирные комментарии в фигурных скобках обозначают сложное действие или необходимость принятия решения Администратором.

На самом деле даже эта процедура упрощена по сравнению с реально достаточной. Нужна проверка того, что задаваемый адрес отсутствует на всех портах коммутатора!

```
  {Вызов протокола Telnet на нужный сетевой адрес коммутатора}
  User Access Verification
  Username: NNNN
  Password: TTTTT
  lan_switch#show port-security interface fastethernet 0/16 address
            Secure Mac Address Table
  Vlan    Mac Address       Type                Ports  RemainingAge
                                                        (mins)
  ----    -----------       ----                -----  --------
   194    0005.5d49.bf22    SecureConfigured    Fa0/16       -
   194    0013.d44a.aa7e    SecureConfigured    Fa0/16       -
   194    0013.d4b8.2847    SecureConfigured    Fa0/16       -
   194    0040.0541.319b    SecureConfigured    Fa0/16       -
   194    00c0.262c.1799    SecureConfigured    Fa0/16       -
   194    00c0.26a6.6406    SecureConfigured    Fa0/16       -
  ------------------------------------------------------------
  Total Addresses: 6
  {Удостовериться, что нужного адреса нет в списке, число MAC-адресов
меньше предела}
  lan_switch#show port-security interface fastethernet 0/16
  Port Security              : Enabled
  Port Status                : Secure-up
  Violation Mode             : Restrict
  Aging Time                 : 0 mins
  Aging Type                 : Absolute
```

```
  SecureStatic Address Aging : Disabled
  Maximum MAC Addresses      : 6
  Total MAC Addresses        : 6
  Configured MAC Addresses   : 6
  Sticky MAC Addresses       : 0
  Last Source Address        : 0017.4215.e584
  Security Violation Count   : 108
```
{Удостовериться, что порт в рабочем состоянии, режим Secure включен, и текущее число Secure-устройств в списке не превышает допустимого максимума}
```
  lan_switch#config term
  Enter configuration commands, one per line.  End with CNTL/Z.

  lan_switch(config)#interface fastethernet 0/16
  lan_switch(config-if)#shutdown
```
{Обязательное отключение работы порта, чтобы не произошло автоматического изменения содержимого под влиянием приходящих пакетов}
```
  lan_switch(config-if)#switchport port-security maximum 7
  lan_switch(config-if)#switchport port-sec mac-address 1234.5678.9abc
```
{Удостовериться, что адрес принят - нет сообщений об ошибках}
```
  lan_switch(config-if)#no shutdown
  lan_switch(config-if)#exit
  lan_switch(config)#exit
  lan_switch#exit
```

Рис. 21. Управление коммутатором при добавлении MAC-адреса вручную

Т.к. ошибка даже в одном символе при вводе приведёт к необходимости повторить набор всей команды заново, легко видеть, что подобный ввод достаточно сложных по синтаксису команд, сопряжённый с необходимостью постоянно контролировать вывод, отнюдь не является простым для исполняющего Оператора действием. Если же ответы коммутатора не соответствуют ожидаемым, требуется исполнение ряда действий, включающих, помимо анализа ситуации, также ввод ряда команд, устраняющих некорректность ситуации.

Следует также иметь в виду, что управление коммутатором исполняется в реальном режиме времени, так что с подачей команды **shutdown** порт будет в нерабочем состоянии, и все пользователи, прикреплённые к этому порту, в это время будут находиться без связи с Интернетом. Анализ ситуации и необходимые действия займут заметное время (~10 мин), так что открытые к моменту начала операций соединения у пользователей порта заведомо будут разорваны.

В то же время, представляется, все перечисленные выше в начале подраздела стандартные операции представляют собой по сути совокупность достаточно простых действий, опционально использующие один или два параметра: номер интерфейса (порта) коммутатора и, возможно, MAC-адрес сетевого устройства. Учитывая частоту приведённых операций в повседневной практике управления коммутаторами, логично было бы ожидать появления прикладных пакетов, использующих средства автоматизации программирования как метода оптимизации управлением коммутатора. Однако, на актуальном рынке программного обеспечения подобные средства, обеспечивающие лёгкий интерфейс с гибким построением реализующих процедур, отсутствуют до настоящего времени.

Как было указано в предыдущем разделе, автором данной работы был разработан язык описания синтаксических процедур как надстройка над языком управления, используемом в командных строках управления сетевыми коммутаторами через Telnet. Реализованный транслятор и блоки связи по Telnet были встроены в описываемую программу. Впоследствии, как будет показано в Главе 2, аналогичный интерфейс встроен и в центральную программу сетевого мониторинга. Такая центральная программа (названная в рамках построения системы круглосуточного сетевого мониторинга «Мониторной программой») постоянно загружена на постоянно включённом ПК, что обеспечивает её круглосуточную готовность к необходимым действиям.

Однако, практика показала, что описанная методика имеет практическую ценность независимо от системы сетевого мониторинга, как автономное средство управления сетевыми коммутаторами, доступное в использовании для низкоквалифицированного персонала. В силу этого, была выполнена отдельная разработка «**TAM Cisco Console**», использующая описанный язык создания процедур и реализованный транслятор, однако представляющая собой самостоятельную консоль управления сетевым коммутатором.

Конечно, программа требует конфигурации – на Интернет-имя управляемого устройства, login, пароль, vlan, местонахождение библиотеки процедур и несколько дополнительных параметров, таких, как степень подробности показа действий в протоколе работы (**Debug – Full – Short**), открытие окна протокола и некоторых других.

Программа отображает все выходные данные в одно окно (рис.23), нижняя часть которого отведена под протокол и может быть показана на экране кнопкой **Show** (на рисунке отсутствует). Окно содержит следующие объекты.

Пиктограмма 

в верхнем левом углу символически показывает объект управления Cisco и консоль управления. Во время Telnet-сеанса на пиктограмме показывается наличие связи между ними: 

.

Идентификационные признаки включают версию программы, текущую дату и время (на том ПК, на котором исполняется консоль), имя или IP-адрес управляемого объекта (в начале активного соединения имя заменяется на IP-адрес).

Группа управления сосредоточена в верхней правой части. Включает:

– поле флажка Начало;

– поле Комментарий справа от флажка Начало;

– Таймер (не виден в состоянии покоя) слева от поля процедуры;

– поля выбора Процедуры и ввода MAC-адреса и Порта, в обычном состоянии не реагируют.

Поле Telnet-сеанса занимает всю ширину в центральной части окна. Здесь отображается исполнение процедур. Во многом соответствует окну стандартного сеанса TelNet, однако помимо генерируемых команд управления и ответной выдачи коммутатора присутствуют также сообщения некоторых операторов языка управления процедурами.

Группа кнопок находится ниже поля сеанса. В данной версии действуют кнопки:

– Hide – скрывает изображение нижней части окна, в которой показывается протокол работы (при этом надпись на кнопке меняется на Show);

– ReRead – заново считывает файл библиотеки процедур. Кнопка создана для оперативной отладки процедур специалистом;

– About – выводит справочное окно с фотографией автора и кратким описанием цели программы, версии компилятора и его лицензией (рис. 22);

Рис. 22. Справочное окно программы About

– **ShortHelp** – выводит дополнительное окно Помощи, в котором более подробно поясняется цель программы и находится полный список команд процедур (для оперативных справочных целей);

– **Quit** – кнопка завершения работы программы;

– **Start** – инициация Telnet-сеанса, становится доступной после выбора процедуры и завершении ввода всех необходимых параметров.

Флажок **Видеть ПСК** (на рисунке не отражён) между кнопками Quit и Start предназначен для отладочных целей и обычно сброшен.

Нижняя часть окна содержит окно протокола работы программы, радио-кнопки управления подробностью вывода протокола и заглавие окна. При уверенной работе и отсутствии проблем эта часть выходной информации Оператору не нужна. В то же время, при отладке процедур или непонятных сбоях сеансов TelNet, а также при обучении новичка помогает понять действия программы. Три уровня подробности протокола (***Short*** / ***Full*** / ***Debug***) предусматривают:

– режим **Short** – выдачу только основных сообщений об успешно прошедших этапах и всех сообщений об ошибках с уточненной диагностикой;

– режим **Full** – выдачу всех сообщений как об успешно прошедших этапах, так и всех сообщений об ошибках с уточненной диагностикой (этот режим используется специалистом при написании и отладке TelNet-процедур);

– режим **Debug** – выдачу всей отладочной информации о каждом этапе TelNet-сеанса (каждой принятой строке и т.д.), содержит очень много (сотни) сообщений по каждой Telnet-сессии. Режим предназначен для отладки программы.

Щелчок мышью во время работы на этой области вызывает временное блокирование выводимых в окно сообщений, чтобы лучше их расмотреть (работа процедуры при этом продолжается). Повторный щелчок восстанавливает вывод протокола в окно.

Пользователь, включив флажок ручного управления (**Начало**), выбирает по списку нужную процедуру и, если она требует, вводит необходимый MAC-адрес и/или номер порта, после чего нажимает кнопку **Start** и ждёт завершения. Результат чётко виден в специальной строке под номерами MAC и порта на русском языке и может быть удачным или неудачным.

В случае неудачи, в основном окне малиновым цветом на русском языке кратко поясняется причина ошибки. Язык управления обладает достаточно богатыми возможностями, чтобы описать с его помощью практически все возможные ситуации, которые можно предусмотреть, от корректного исполнения каждой поданной на коммутатор команды до съёма и проверки нужных параметров, как текстовых, так и числовых. Разумеется, само создание процедур на предложенном языке является задачей высококвалифицированного специалиста (Администратора). Однако, после отладки такой процедуры она может быть включена в библиотеку и в дальнейшем использоваться Операторами, плохо знакомыми с предметной областью, Telnet'ом, командами управления коммутатором или синтаксисом языка разработки процедур.

Таким образом достигается возможность исполнять сложнейшие действия над коммутатором низкоквалифицированным рабочим персоналом без риска повредить тонкую настройку IOS на состав сетевых устройств.

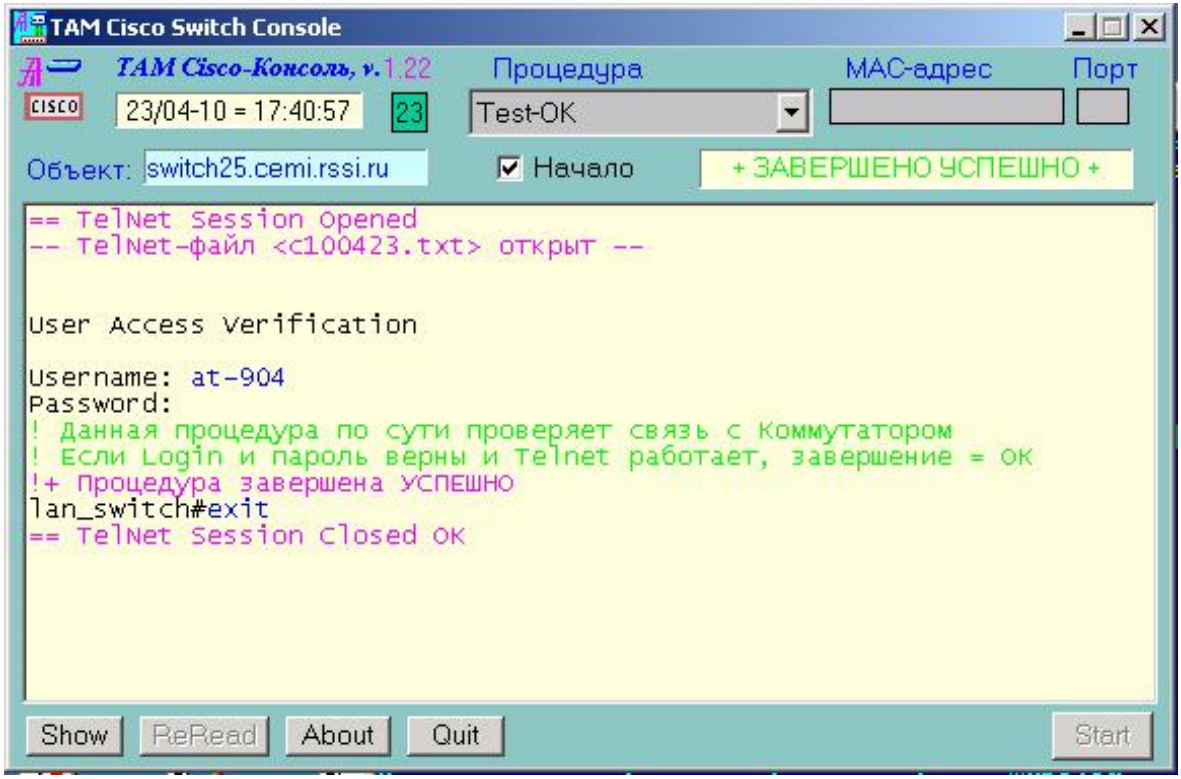


Рис. 23. Общий вид окна программы TAM Cisco Console

Так, например, при добавлении некоторого сетевого объекта на порт коммутатора в режиме SECURE может создаться одна из неприемлемых ситуаций. Указанный объект может уже быть присоединённым к нужному или другому порту, требуемый порт может не находиться в режиме SECURE или вообще быть отключенным, может быть достигнуто предельное число возможных подключений к данному порту. Наконец, может произойти сбой при управления коммутатором или разрыв связи. Все эти ситуации возможно предусмотреть с помощью проверок специальными командами управления коммутатором и анализом их результатов с помощью псевдокоманд языка (разрыв связи автоматически детектируется исполняющими блоками программы).

Смысл Secure-порта в том, что принятие информации на него допускаетсялишь от фиксированного числа пользователей, представленных MAC-адресами. Отдельно задаются адреса и отдельно указывается предельное их число на порту. При наличии только одного MAC-адреса предельное число **1** не указывается. Поэтому при добавлении нового адреса сначала нужно увеличить предельное число **maxcount** и лишь потом добавлять новый MAC-адрес. В случае, если по тем или иным причинам MAC-адрес не будет добавлен, нужно вновь изменить предельное число, уменьшив его на 1 либо, если он был равен 1, вообще убрать строку с указанием maxcount.

Тестовая процедура, написанная на предложенном языке в целях добавления нового объекта на некоторый порт, содержит 82 оператора языка, из которых только 21 является командами управления коммутатором через Telnet. В остальные входят: 1 строка заголовка, 11 меток, 14 строк невыводимых при работе процедуры комментариев программиста и 35 псевдооператоров управления (включая 8 выводимых комментариев ситуаций). Полученная процедура предусматривает практически все ситуации, которые могут случиться во время исполнения рабочего цикла. Текст актуальной на момент написания данной работы версии процедуры приведён на рис. 24 (слева на рисунке специально вставлены номера строк; далее с их помощью будет пояснен смысл некоторых операторов). Для удобства восприятия, реальные команды коммутатору выделены жирным, псевдооператоры-метки начала новых блоков подчеркнуты.

```
   001 [Add-Secure]MI
   002 ' Процедура добавления Secure-MAC, версия
09.04.2010
   003 #"&M"Y<51
   004 ' Выдача всех MAC-адресов на нужном порту
   005 show port-security interface fastethernet
0/&I address
   006 ' Если Y имеет реальное значение, goto NoDef
- это нормально
   007 ?Y%NoDef
   008 !* Mac-Address &M уже есть на порту &Y
   009 ^-
   010 :NoDef
   011 #"Maximum"X<30
   012 #"Total"Z<30
   013 $"Port Sec"W<30,3
   014 ' Приказ на выдачу параметров всех портов
   015 show port-security interface fastethernet
0/&I
   016 ' Если W не имеет значения, то порт не типа
Secure
   017 ?W%PortSecN
   018 ' Если значение W не "Ena"[bled], то порт не
типа Secure
   019 ?W="Ena">PortSecY
   020 :PortSecN
   021 !* Порт &I не является Secure-портом
   022 ^-
   023 :PortSecY
   024 config term
   025 interface fastethernet 0/&I
   026 shutdown
   027 ' Переменная Z получила число из строки
"Total Secured";
   028 ' значение 0 значит, что не надо увеличивать
MaxCount
   029 ?Z=0>NoNeedIncr
   030 ' Увеличение X как счётчика MaxCount
   031 =X+
   032 switchport port-security max &X
   033 %AnyBad
   034 >ExecAdd
   035 :NoNeedIncr
   036 ! Порт был пуст, max увеличивать не надо
   037 :ExecAdd
   038 ' Сл. Строка фиксирует "MAC Found on the
other port" ситуацию
   039 $"Found"U<1,5
   040 switchport port-security mac-address &M
   041 %AnyBad2
   042 ?U%NormAdd
```

```
   043 >AnyBad2
   044 :NormAdd
   045 ! MAC-Addr &M успешно добавлен к порту &I
   046 no shutdown
   047 exit
   048 exit
   049 ' Это единственное место нормального завер-
шения процедуры
   050 ^+
   051 :AnyBad
   052 !* Проблемы при увеличении max. Изучайте.
   053 no shutdown
   054 exit
   055 exit
   056 ^-
   057 :AnyBad2
   058 ' Данный блок срабатывает при ошибке уста-
новки нового MAC;
   059 ' будет попытка уменьшить на 1 MaxCount
   060 ?Z=0>NoNeedDecr
   061 =X-
   062 !* Проблемы с добавлением MAC. Восстановим
max = &X
   063 switchport port-security max &X
   064 %AnyMoreBad
   065 :NoNeedDecr
   066 no shutdown
   067 exit
   068 exit
   069 !* Maxcount восстановлен. MAC не добавлен.
   070 ' Теперь поинтересуемся, почему. Выведем
список всех MAC
   071 #"&M"Y<51
   072 show port-security address
   073 ?Y%EndNoInf
   074 !* MAC &M обнаружен на порту &Y
   075 :EndNoInf
   076 ^-
   077 :AnyMoreBad
   078 !*= Проблемы! MAC не добавлен, max не вос-
становлен.
   079 no shutdown
   080 exit
   081 exit
   082 ^-
```

Рис. 24. Реальная процедура добавления MAC-адреса

В случае успеха будет выдан комментарий:

MAC-Addr &M успешно добавлен к порту &I

В случае неудачи будет выдана причина неуспеха. Можно видеть, что среди конкретных причин неуспеха присутствуют:

**** Mac-Address &M уже есть на порту &Y*** – Смысл ясен из текста сообщения: попытка установить MAC-адрес, который уже есть на порту.

**** Порт &I не является Secure-портом*** – Смысл ясен из текста: заданный порт не находится в режиме Secure. Требуются действия Администратора по выяснению причин этого.

*** MAC &M обнаружен на порту *jj*** – Смысл ясен из текста сообщения: данный MAC-адрес обнаружен на другом порту коммутатора.

**** Проблемы с добавлением MAC. Восстановим max = &X*** – Информационное сообщение о попытке восстановления значения **maxcount** после того, как не удалось добавить желаемый MAC-адрес. Работа процедуры будет продолжена.

**** Maxcount восстановлен. MAC не добавлен.*** – Итоговое сообщение после вышеуказанной ситуации. Конфигурация коммутатора корректна.

****= Проблемы! MAC не добавлен, max не восстановлен.*** – Итоговое сообщение после вышеуказанной ситуации. Конфигурация коммутатора некорректна, требуется вмешательство Администратора!

Команды ввода логина и пароля, также как и завершающая команда **exit**, подаются программой автоматически.

Прокомментируем «сердце» процедуры – блок строк 39÷43.

Строка 39 является псевдокомандой языка управления. Смысл её в том, что она приказывает анализировать все последующие строки, выдаваемые коммутатором, на присутствие контента "**Found**", и если он будет найден, то присвоить переменной **U** текстовое значение из той же строки, в которой найден контент "**Found**", начиная с позиции 1, числом 5 символов. Перед исполнением команды в строке 39, текущее значение переменной **U** устанавливается неопределенным.

Строка 40 является командой добавления MAC-адреса на заранее указанный порт. Перед выдачей команды коммутатору для исполнения процессор формирования команд заменит макроконструкцию "**&M**" на введённый оператором MAC-адрес.

Строка 41 содержит условную передачу управления на метку **AnyBad2** в случае, если после исполнения предыдущей команды коммутатор выдал любую строку, начинающуюся с символа «процент» (“**%**”) – именно такой вид имеют все типовые сообщения коммутатора об ошибках.

Строка 42 является псевдокомандой языка управления. Это условная передача управления на метку **NormAdd** в случае, если переменная **U** не получила значения. Иными словами, если после подачи коммутатору команды строки 40 в ответ *не был* выдан текст, содержащий слово “**Found**” точно в таком виде. Тем самым проверяется ситуация, когда коммутатор сообщил, что добавляемый MAC-адрес уже присутствует на каком-либо ином порту. В дополнение, эта же команда отменяет все заданные анализы будущего контента на содержание “**Found**” и принудительно сбрасывает значение переменной **U**.

Описанные две проверки полностью охватывают все аварийные ситуации, которые могут возникнуть при наборе основной команды добавления MAC-адреса строки 40. Все остальные проверки построены по аналогичным принципам анализа контента на образец и «захвата» некоторой части строки как числового или текстового значения. Разумеется, одновременно могут быть заданы несколько приказов на анализ контента по отношению к различным внутренним переменным языка (напр., строки 11÷13). Учитывая особенности формата выводимых коммутатором строк, числовые значения в которых всегда начинаются с определённых позиций и продолжаются до пробела, в приказах на приём числовых значений (например, в строке 11) указывается только стартовая позиция числа.

Описанная процедура при своём исполнении выдаст однозначный признак успешности завершения на русском языке, а также, в случае неуспеха, комментарий, также на русском языке, с кратким описанием причины неуспеха (напр., текст «**Maxcount восстановлен. MAC не добавлен**.»).

Приведённый пример показывает возможности и простоту синтаксиса языка. Ниже на рис. 25 дан полный экран исполнения процедуры добавления MAC-адреса (обрезанные верх и низ экрана содержат стандартные строки, точно совпадающие с аналогичными на рис. 23).

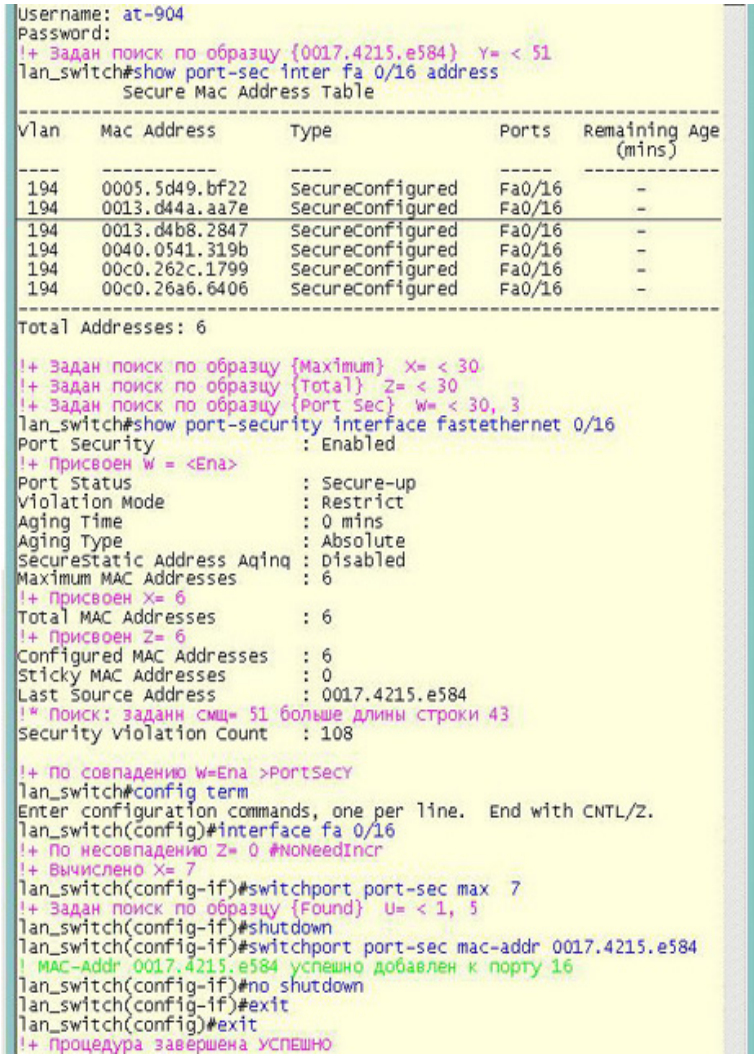


Рис. 25. Полный экран исполнения процедуры добавления MAC-адреса

Программный модуль при исполнении выводит главное окно, сокращённый вид которого показан на рис. 22. В окне присутствуют элементы управления и встроенное окно показа исполняемых Telnet-команд, куда также выводится дополнительная информация о ходе исполнения программы и комментарии. При необходимости, окно может быть расширено вниз для просмотра текущего фрагмента протокола работы. По исполнении программы, полный протокол работы формируется в специальном файле. Отдельными файлами сохраняются все исполненные процедуры в виде Telnet-протокола.

Разумеется, блок ввода автоматически контролирует набираемые оператором символы, блокируя недопустимые (например, при наборе MAC-адреса допускаются ровно 12 цифр или букв латинского алфавита в интервале **A** ÷ **F**). Также контролируется полнота задания параметров в зависимости от заголовка процедуры.

Как исполняемые команды управления коммутатором, так и операторы макроязыка протоколируются на экране и в файл, позволяя впоследствии в случае необходимости проанализировать ситуацию квалифицированному специалисту.

Исполняемые команды коммутатора, выдаваемый коммутатором ответный текст, сопровождающая информация интерпретатора макроязыка и комментарии выполняются разными цветами. Для исполнения этого при программировании использован специальный блок **RichTextEdit**, созданный энтузиастами языка PowerBASIC [60]. Подсказки коммутатора даны чёрным цветом, посылаемые исполняемые команды – синим, комментарии выделены малиновым или зелёным.

Таким образом, создано средство автоматизации управления сетевым коммутатором, доступное для использования персоналом низкой квалификации, сохраняющее все достоинства интеллектуальной реакции на конкретную ситуацию. Описанное средство не требует длительного обучения персонала, просто и надёжно в эксплуатации, не требует инсталляции на исполняющий компьютер. Настройка **TAM Cisco Console** на сетевые адреса и входную информацию (логины и пароли) нужных коммутаторов исполняется один раз при установке средства Администратором с помощью любого текстового редактора и впоследствии может быть изменена.

Особо следует отметить простоту отладки формируемых процедур квалифицированным пользователем. Тексты процедур содержатся в отдельном файле, который может быть отредактирован в любой момент любым текстовым редактором. На основной панели программы имеется кнопка **ReRead**, нажатие которой загружает новую версию этого файла как библиотеку процедур. Таким образом, исправив текст той или иной процедуры, можно немедленно его задействовать без перезагрузки программы. В ходе исполнения процедуры в расширенном окне можно просматривать протокол работы, формируемый с различной, устанавливаемой пользователем, степенью подробности (рис. 26).

Разработка выполнена на языке высокого уровня **PowerBASIC for Windows 9.05** [61] [62] (serial 513311953), содержит 1962 строки. Для контроля вводимых символов на допустимость и обеспечения смыслового выделения текста разными цветами в одном и том же окне используются subclass-процедуры блока **RichTextEdit** с включением Windows API. Объём программного модуля составляет 233Кб, занимаемая память при исполнении – 4Мб[11]. Программа не требует инсталляции, имеет окно встроенной Помощи по синтаксису команд и работоспособна на Windows XP / 7+. Версия программы **TAM Cisco Switch Console 1.25** зарегистрирована в ФИПС 17.07.2012 г под номером 2012618011.

Известных автору аналогов среди мировых публикаций не обнаружено.

Практическая эксплуатация **TAM Cisco Console** на двух рабочих местах в ЦЭМИ РАН в течение ряда лет показала, что даже указанная процедура добавления SECURE-адреса (самая длинная из всех) исполняется менее чем за 15с. За это время большинство установленных TCP/IP соединений, проходящих через конфигурируемый порт, несмотря на его кратковременный **shutdown**, остаются нерассоединёнными. Таким образом, помимо возможности задействования низкоквалифицированного персонала для управления технически сложными коммутаторами, налицо ещё и выигрыш для пользователей от применённой автоматизации.

[11] Сравните с ресурсными характеристиками известных графических пакетов!

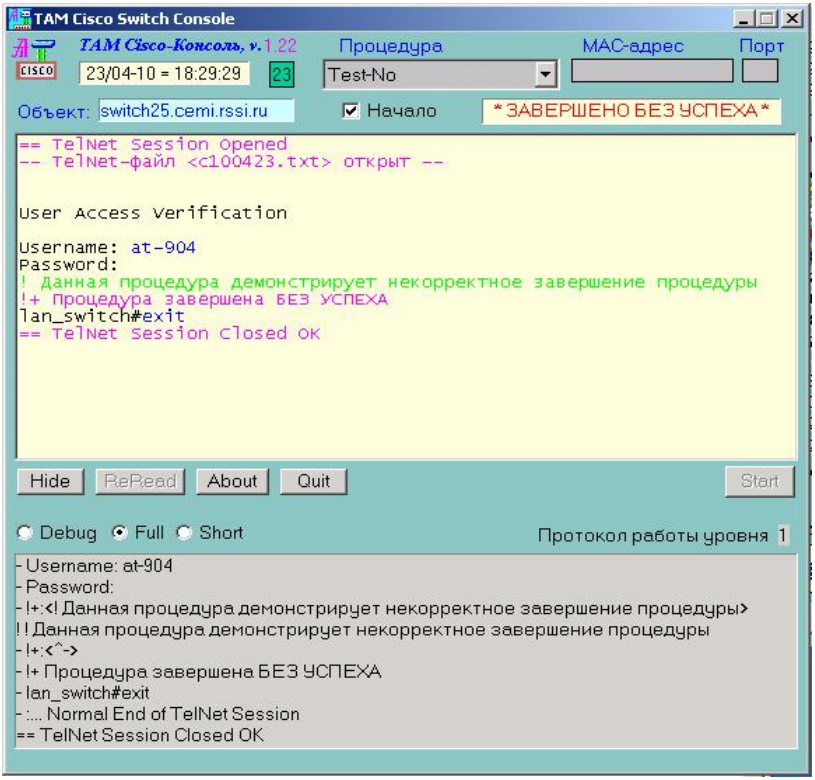


Рис. 26. Расширенный вид окна TAM Cisco Console с показом протокола работы

Данная программа легко переносится на другое рабочее место или в иную сеть любой организации (учреждения), не требует инсталляции и адаптируется под нужный коммутатор любым текстовым редактором.

Блоки процедур могут быть легко изменены, пополнены или адаптированы под другую версию интерфейса Cisco также с помощью любого текстового редактора. Разумеется, подобная работа требует специалиста высокой квалификации.

Поскольку файл конфигурации первоначально ищется в текущем каталоге, то вызов программы из разных каталогов может использоваться для выбора управления нужным устройством.

ГЛАВА 2. ИНФОРМАЦИОННЫЕ ПОТОКИ МОНИТОРНОЙ ПРОГРАММЫ

2.1. Мониторная программа как средство интеграции данных наблюдающей станции в локальной сети

Средства сетевого мониторинга, разработанные автором [1], позволяют получить достоверную информацию о функционировании объектов корпоративной вычислительной сети (КВС). Постоянный мониторинг более чем 80 сетевых параметров дает ценную информацию о работе сети в целом и ряде важных сетевых объектов в частности, актуальную в решении задач проектирования и управления сетью, экономических расчетах за трафик, и иных [13]. При этом мониторинг исполняется с помощью наблюдающей станции (НС) – выделенного ПК, оснащенного специальным программным обеспечением, эксплуатирующегося под управлением MS-DOS 6.22. Важнейшая часть контролируемых параметров, в том числе большинство общесетевых характеристик, доступна для наблюдения на экране НС. Все параметры также фиксируются в периодических отчетах НС, формируемых на HDD. Каждые 15 минут формируется краткий, каждый час – полный отчет. Как правило, в целях полноты мониторинга работу НС прерывать нежелательно; 1 раз в сутки серия отчетов записывается из рабочего каталога в архив в упакованном формате в виде единого блока. Один или более раз в неделю, обычно в начале часа, работа НС прерывается оператором на несколько минут для переписи вновь образованных за это время блоков на дискету. Анализ отчетов исполняется на другом ПК.

В необходимых случаях в текущий мониторинг могут быть внесены коррективы. К примеру, оператор, не прерывая общего процесса сбора информации, может задать ряд фильтров, указав логический ip-адрес или физический MAC-адрес интересуемого объекта, и разрешить дампирование отфильтрованных пакетов на HDD на нужный отрезок времени. В MS-DOS, однозадачной среде, для снятия дампа на дискету нужно приостановить мониторинг.

Описанные варианты использования средств сетевого мониторинга в версиях, разработанных до 2005г., страдают очевидным недостатком. Как видим, актуальная сетевая информация была доступна либо визуально на экране НС в момент ее появления, либо

после снятия блока отчетов, обычно через несколько суток. Ряд параметров (например, текущий список сетевых пользователей) доступен на экране в течение 15 минут до формирования очередного отчета. Таким образом, изучение актуальных данных средств сетевого мониторинга было с необходимостью отнесено во времени, либо приводило к прерыванию текущего мониторинга. На практике, использовались две НС [7]. Основная НС вела непрерывный мониторинг, вспомогательная НС включалась лишь для проведения специальных исследований. В обоих вариантах, данные мониторинга были доступными лишь с рабочих мест установки этих НС.

Как уже было отмечено [8], такой изолированный вариант эксплуатации средств сетевого мониторинга не отвечает одной из весьма соблазнительных целей – актуального задействования данных мониторинга для последующего использования, возможно, автоматизированного. Программой исследований, между тем, было предусмотрено непосредственное использование данных сетевого мониторинга в целях формирования комплекса автоматизированных средств управления сетью с применением сетевого коммутатора.

Для решения указанной проблемы предпринято следующее.

Разработана версия мониторной программы (МП), демонстрирующей информацию о ряде актуальных сетевых параметров. МП выполнена в виде Windows-приложения реального времени. Связь с DOS-программой наблюдающей станции осуществляется по serial-порту ежесекундно по инициативе МП. В свою очередь, НС, для сокращения объема программирования ее дополнительных блоков, работает в режиме отклика: при появлении на serial-порте кода связи выдается отклик с текущим временем; группа фиксированных актуальных параметров должна быть запрошена специально. Структурно блок-схема подобного кооперативного использования НС и МП показана на рис. 27. На этом рисунке оборудование группы информационной безопасности (ИБ), находящееся в специальном помещении, выделено штрих-пунктирной линией.

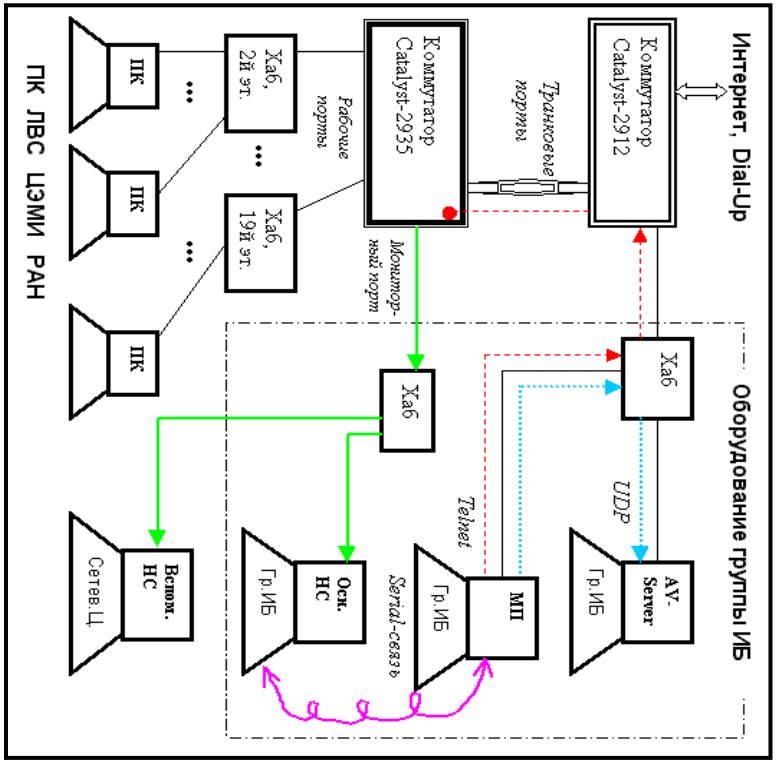


Рис. 27. Схема взаимодействия НС и мониторной программы

Впоследствии хаб, связывающий Антивирусный сервер и компьютер с МП, был заменён на коммутатор Cisco SG 200-08, для улучшения коннективности.

Красные пунктирные стрелки показывают цепь управления коммутатором Cisco-2935. Стоит обратить внимание, что сам компьютер с МП подключён через вышестоящий к управляемому коммутатору Cisco-2935 коммутатор Cisco-2912. Это сделано для того, чтобы исключить возможность потери связи при управлении Cisco-2935 в случае перевода его портов в административный **shutdown**.

Фиолетовая винтовая линия показывает физическое соединение НС с МП через serial-порты, это единственный доступный способ передачи информации между компьютерами с MS-DOS и Windows.

Зелёная сплошная линия показывает снятие информации с мониторного порта коммутатора Cisco-2935 и отображение её на двух НС.

Наконец, голубая точечная линия показывает передачу информации от МП к Антивирусному серверу, для последующего показа в Интернете (см. 2.3).

При отборе актуальных параметров для демонстрации в МП из всего множества выбраны характеризующие общее состояние сети и общее состояние программы НС. Поскольку ряд параметров в НС сохраняется накопительным итогом с начала последнего 15-минутного цикла наблюдений, имело смысл также отобразить параметры этого цикла. Далее при описании параметров сохранены введенные автором в [2] термины.

В итоговой версии МП (1.80) к отображаемым параметрам относятся следующие (см.рис. 28).

- Текущее время ПК, на котором запущена МП.
- Текущее время ПК, который используется как НС.
- Текущее время ПК, на котором стоит AV-Server (пояснено далее).

Группа параметров, относящаяся к текущему циклу наблюдений (выделено рамочкой.

- Прошедшее время в секундах со времени последнего отчета, выполненного наблюдающей станцией, после которого счетчики были сброшены – этот период назван «циклом наблюдения» (обычно 15 мин., однако в начале каждого часа текущий цикл наблюдения принудительно заканчивается и начинается новый цикл).

■ Число байт в сетевых пакетах, прошедших обработку на НС со времени начала последнего цикла наблюдения, включая все служебные поля, кроме поля контрольной суммы Ethernet-пакета.

■ Число байт в сетевых пакетах, отказанных наблюдающей станцией к обработке вследствие переполнения внутренних буферов при высоких скоростях в сети.

■ Число байт в обработанных сетевых пакетах, являющихся бродкастовыми (т.е. широковещательными).

■ Три числовых значения в байтах относительно обработанных сетевых пакетов: пришедших из гейта (**Internet-From**) на какой-либо из узлов локальной сети; ушедших на гейт (**Internet-To**) с какого-либо узла локальной сети; и прочие пакеты, естественно составляющие внутрисетевые (**InterLAN**) пересылки. Разумеется, суммарное значение этих трех позиций должно совпадать с числом байтов в обработанных наблюдающей станцией пакетах.

■ Число различных технологических сетевых адресов (MAC-адресов), замеченных в испускании сетевых пакетов с момента начала цикла наблюдения. Справа вне рамки дано максимальное зафиксированное чисто нод с момента запуска программы.

Далее следует группа параметров, для каждого из которых даётся три значения: текущее (за последнюю секунду), максимальное за отчётный цикл наблюдения и максимальное с момента старта последних суток.

■ Мгновенная скорость в килобайтах в секунду;

■ Рабочих циклов НС в секунду;

■ Полезных рабочих циклов в секунду, т.е. таких, во время которых был обработан пришедший пакет;

■ Заполнение внутренних буферов программы НС, показывающее загрузку алгоритма обработки пакетов.

■ Последний номер отчета, сформированный наблюдающей станцией по завершении предыдущего цикла наблюдения.

Внизу левой части экрана ведётся пополняемый список MAC-адресов, для которых выполнена изоляция от работы в сети вследствие нарушения технологии работы.

Ещё ниже находятся кнопки управления работой МП.

На рисунке показана также необязательная к демонстрации правая часть экрана МП. Многое в ней совпадает с аналогичной частью экрана программы TAMCICON, описанной в разделе 1.4. Так,

например, в верхней части правой стороны экрана находится окно, показывающее ход исполнения автоматической процедуры из числа допустимых к выбору вручную либо запускаемой автоматизированно по сигналу НС. Однако, в отличие от программы TAMCICON, над этим окном есть кнопки выбора показа желаемой связи по COM-порту с НС, по UDP с сервером или сеанса TelNet. В основном, это окно использовалось при отладке соответствующих связей.

Под этим окном находятся элементы ручного управления коммутатором, аналогичные программе TAMCICON. В случае необходимости Оператор может запустить необходимую процедуру управления коммутатором вручную, начав с нажатия кнопки-флажка **Manually**.

В нижней части справа находится окно протокола работы программы. На рисунке видна конечная часть библиотеки процедур, установочные сообщения и сведения о двух исключенных MAC-адресах. При щелчке на этом окне вывод в него приостанавливается (что и изображено на рисунке), при повторном щелчке вывод восстанавливается.

Как видим, принимаемые мониторной программой данные сетевого мониторинга носят исключительно общий характер, однако включают как основные показатели активности сети, так и ряд технологических показателей качества работы самой наблюдающей станции. Вид окна околофинальной версии мониторной программы приведен на рис. 28.

Следует отметить, что иконка слева вверху окна включает в себя изображение глаза. Этот глаз открыт, если связь с НС была активна в последнюю секунду, иначе глаз показывается прикрытым. На практике, глаз прикрывается примерно 1 раз в минуту.

Мониторная программа первоначально написана на языке **PowerBASIC for Windows 7.04**, впоследствии **9.05** [60] – Windows-диалекте языка той же фирмы, которая выпустила **PowerBASIC for DOS**, использованный для программирования НС сетевого мониторинга. Особую трудность при программировании вызвало исполнение программы в режиме немодального (modeless) диалога. Языки высокого уровня вообще, и **PowerBASIC for Windows** в частности, не обладают особо развитыми средствами для исполнения программ реального времени (кстати, пригодность Windows как операционного средства для задач реального времени также неочевидна автору данной работы).

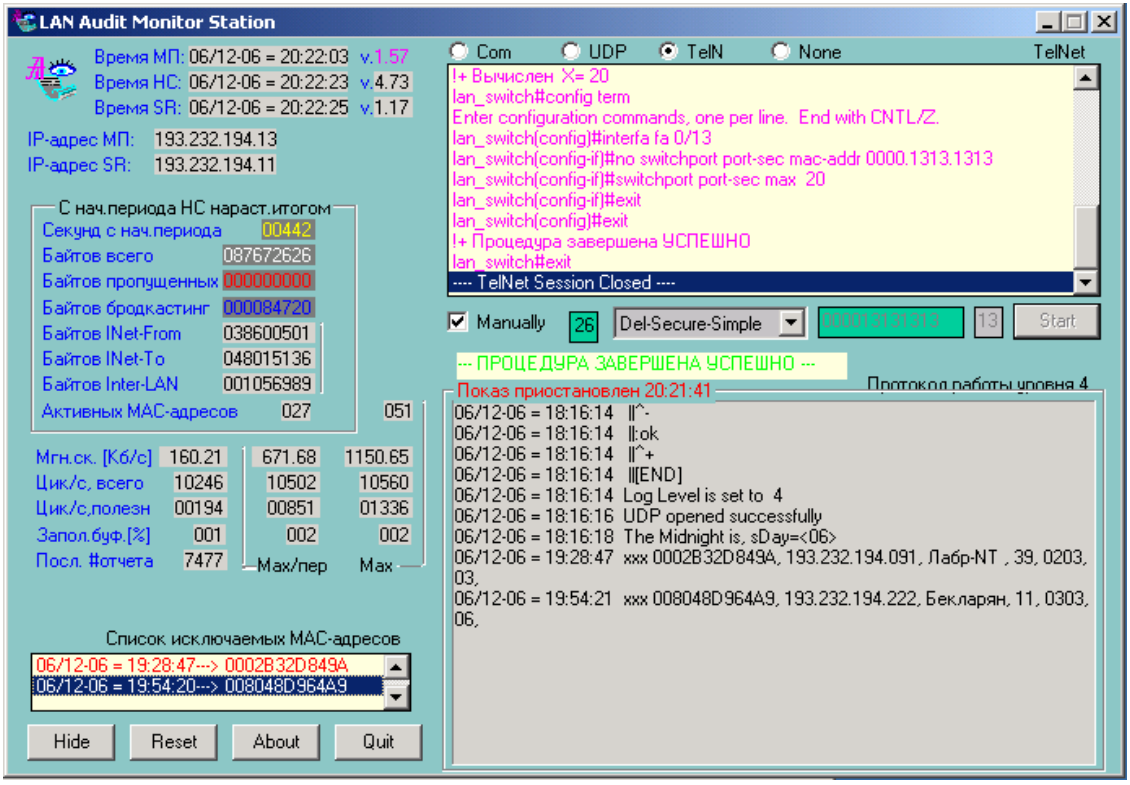


Рис. 28. Вид расширенного окна мониторной программы

Тем не менее, оказалось возможным для Windows-среды обеспечить устойчивое соединение по serial-порту с DOS-приложением с использованием операторов синхронного вывода и асинхронного ввода по serial-порту.

Развитие МП от её первоначального вида можно посмотреть по [65].

Более интересно отслеживание ряда показателей сетевой активности, достигнутых за время цикла наблюдения, а также и за более длинный период. Интересно знать максимальное число пользователей за истекшие сутки, максимальную замеренную скорость в сети.

В принципе, все получаемые наблюдающей станцией данные могут быть поданы мониторной программе. Основной объем информации НС включает данные по конкретным сетевым устройствам, например, местонахождение всех сетевых устройств, порты их подключения к центральному коммутатору Cisco Catalyst-2924, фиксация почтовых отправлений, адреса сайтов, к которым обращаются пользователи. Объем этих данных, однако, столь значителен, что работа с ними не планировалась.

Еще более интересной представляется возможность передачи принятых мониторной программой данных от НС для общего сведения. С момента получения устойчивой связи НС – МП, реальной становится подача актуальных материалов наблюдения в раздел «Мониторинг» на существующий Антивирусный сайт [74]. На рис. 27 эта работа указана двунаправленной пунктирной малиновой стрелкой между МП и антивирусным сервером ***AV-Server***. Передача данных организована с помощью UDP.

Основная цель всего проекта, а именно оперативное получение от НС технологических адресов сетевых устройств, нарушающих правила поведения в корпоративной сети, для последующего оперативного автоматического управления сетью, связана с необходимостью разработки взаимодействия МП – Cisco Catalyst. Эти работы подробно описаны в первой главе данной работы.

Версия **1.78** мониторной программы **LAN Audit Monitor Station** зарегистрирована в ФИПС 21.05.2014 г под номером 2014614763.

2.2. Автоматическая изоляция некорректных объектов КВС по информации сетевого мониторинга

Базовые средства сетевого мониторинга, разработанные автором [1], позволяют получать достоверную информацию о функционировании объектов корпоративной вычислительной сети (КВС). Вместе с тем, как уже давно [8] [13] указывалось, пассивные методы наблюдения и контроля за сетевыми объектами в настоящее время уже не способны обеспечить эффективную защиту сетевых ресурсов от некорректно настроенных ПК, сетевых вирусов и прочих нарушений регламента работы сетевых объектов.

Общей тенденцией развития корпоративных сетей становится увеличение Интернет-трафика. Выборочные месячные данные за два последних года по КВС ЦЭМИ РАН [64] приведены на рис. 29 (традицией подобных примеров стал сентябрь, однако в сентябре 2006г. по техническим причинам наблюдающая станция сетевого мониторинга заметное время не функционировала, поэтому для сравнения с сентябрём 2005г. взят октябрь 2006 г.). С учётом того, что выходящий в Интернет трафик в сентябре 2005г., как уже было ранее в 2003г. [66], во многом определён вирусной активностью первой недели месяца, легко видеть, что входящий Интернет-трафик увеличился более, чем на 30%. Аналогичная постоянная тенденция наблюдалась и ранее [2].

Отметим, что приведённые в этих таблицах объёмы внутрисетевых пересылок (графа **Intro-LAN**) отражают лишь часть реальной информации, поскольку начавшаяся в 2005г. реорганизация КВС ЦЭМИ РАН вывела из-под мониторинга значительную часть сети в виде обособленного сегмента. После обособления административно-финансового сегмента [1] в 2003 г., это уже второй сегмент, закрытый от мониторинга по внутрисетевым пересылкам. Однако, поскольку роутеры этих сегментов остаются включенными в мониторируемую часть сети, данные Интернет-трафика включают и трафик к/от этих роутеров, так что приведённые данные по Интернет-трафику аутентичны.

	Сентябрь 2005 г. [Мбайты]			Октябрь 2006 г. [Мегабайты]		
День месяца	**From**	**To**	**Intro-LAN**	**From**	**To**	**Intro-LAN**
1	957,77	537,64	232,89	1167,47	630,99	33,69
2	2398,39	9636,91	452,99	3915,88	1154,54	926,15
3	370,37	20537,37	166,12	3471,36	1292,49	389,99
4	370,40	19491,82	70,76	3433,17	1019,59	437,31
5	5119,86	15756,46	590,97	1955,65	1115,59	432,45
6	4458,89	11167,88	209,14	3472,00	998,69	282,02
7	1686,56	4248,92	349,69	382,14	723,64	35,65
8	2165,40	1058,24	381,48	331,76	701,93	68,83
9	1218,55	870,29	174,62	4631,09	1160,20	460,35
10	144,48	270,14	79,36	3319,10	1347,75	248,18
11	177,13	206,02	67,58	3475,10	950,59	247,03
12	1465,23	574,14	674,21	1800,37	601,04	273,39
13	2443,51	671,44	401,17	5277,78	778,15	295,05
14	1282,31	625,15	697,65	6050,14	472,20	68,89
15	2362,89	559,80	286,91	644,27	278,22	63,09
16	1362,32	860,37	208,54	2427,20	793,15	464,63
17	777,34	276,33	64,01	2543,36	818,37	369,55
18	179,92	229,20	98,54	4096,70	724,89	352,81
19	3921,99	847,04	427,57	2915,62	820,20	347,13

20	3358,59	704,17	310,10	4037,20	1383,45	218,06
21	2379,07	874,67	533,65	1933,63	257,82	50,58
22	2900,08	878,96	512,19	488,78	249,68	53,29
23	2167,28	1007,60	397,78	3213,67	748,56	547,37
24	470,89	454,93	83,39	2128,31	653,94	517,81
25	1349,05	648,12	76,60	4149,36	924,18	339,43
26	4853,72	1379,29	480,82	2278,89	550,87	446,30
27	6825,59	1515,85	403,43	4762,22	561,32	178,02
28	4541,68	1026,62	1212,73	4737,33	327,24	25,15
29	2921,41	1278,14	474,07	365,84	164,74	17,43
30	2856,80	1150,72	453,79	2712,81	659,33	484,28
31				3284,34	2431,98	418,78
Итого [Мб]	**67487,47**	99344,23	10572,75	**89402,54**	25295,33	9092,69

Рис. 29. Трафик за сентябрь 2005 г. и октябрь 2006 г.

Для защиты КВС от внешних воздействий предусмотрен целый ряд мер, таких, как закрытые порты на маршрутизаторе Cisco [12], сплошная антивирусная проверка приходящих на официальный почтовый сервер писем, брандмауэры и антивирусные средства на рабочих станциях и серверах [66]. В то же время, **сеть практически не защищена от несанкционированных воздействий изнутри**, начиная от некорректной настройки ПК и до сетевых атак, вызываемых сетевыми вирусами, рассылками спама или хакерскими трюками.

Характерным примером служит ошибочная простановка сетевых настроек на привнесённом ПК, когда вместо выделенного IP-адреса был указан IP-адрес корпоративного сервера DNS. В течение нескольких часов работа всей сети была заблокирована, причём единственным ПК, который успешно выходил в Интернет и «не почувствовал» ситуации, был именно тот принесённый на полдня из дома ноутбук, который и послужил причиной паралича в работе сети.

Представлялось, что с вводом в действие сетевого коммутатора Cisco Catalyst КВС будет защищена от подобных неприятностей: разрешительный доступ по каждому порту каждому сетевому устройству по его фиксированному MAC-адресу, казалось бы, не оставил возможности несанкционированной подмены одного ПК другим. Однако, с появлением операционной системы Windows XP рядовым пользователям стала доступной возможность подмены реального MAC-адреса сетевого адаптера любым другим, который пользователь может ввести непосредственно. Естественно, с появлением подобной возможности следовало ожидать, что найдётся желающий её использовать. Так и случилось.

В этих условиях к началу 2004 г. было запроектировано построение на базе сетевого мониторинга средств автоматизированной внутрисетевой защиты. Резко повысив эффективность средств мониторинга на наблюдающих станциях (НС) [67], была затем разработана мониторная программа (МП), работоспособная в Windows-среде, способная принимать оперативную информацию от НС по serial-кабелю [2.1]. Одновременно была модифицирована КВС ЦЭМИ РАН введением управляемого сетевого коммутатора Cisco Catalyst [1.1]. Таким образом, необходимые технические предпосылки разработки средств автоматизированной защиты были обеспечены.

Впоследствии, как было указано выше, была исследована возможность полуавтоматического управления сетевыми коммутато-

рами Cisco Catalyst и создана пилот-версия программы управления, работоспособная по протоколу Telnet.

К этому моменту выяснилось, что различные версии IOS Cisco по-разному могут обеспечить изоляцию сетевых устройств. Предполагавшийся ранее к использованию простой метод подачи единой команды **drop** запрета приёма информации от сетевого устройства по его технологическому сетевому MAC-адресу, как выяснилось, не является универсальным [59]. Актуальной стала проблема разработки гибких настраиваемых программных средств управления сетевыми коммутаторами.

Эта проблема была решена разработкой специального языка как надстройки над Telnet-командами [1.3], интерпретатор с которого встроен в новую версию мониторной программы. Создана серия эффективных процедур на указанном языке, автоматически запускаемых на МП по сигналу с НС с MAC-адресом нарушителя, для изоляции последнего.

Следует сразу сказать, что термин «изоляция» предложен не случайно. Вероятно, идеальным при обнаружении нарушения было бы отключение нарушителя от всех сетевых функций. Однако, этому есть ряд препятствий.

Во-первых, технологическая структура локальной вычислительной сети (ЛВС) ЦЭМИ РАН такова, что центром «звезды» является один сетевой коммутатор Cisco Catalyst с 24 портами, из которых 4 носят служебный характер (2 транковых, 1 мониторный, 1 резерв). В ЛВС свыше 250 устройств, так что на одном порту коммутатора находятся в среднем 15-20 сетевых объектов (рабочих станций и серверов локального назначения). Отказ приёма сетевых пакетов нарушителя, таким образом, не устранит его влияния на ПК, подключенные к тому же порту (как правило, это ПК того же этажа либо того же подразделения, где находится нарушитель), однако изолирует нарушителя от остальных 200 рабочих станций и, разумеется, от Интернета.

Во-вторых, технологические процессы, проводимые в научном институте с помощью компьютерной техники, имеют разный приоритет. Ведущиеся долговременные модельные расчеты или расчет зарплаты в определённых случаях могут быть более приоритетны, чем блокирование сетевого червя. В таких случаях сохранение доступа на соседний сетевой ПК решает проблему.

В КВС ЦЭМИ РАН действует в составе технических средств сетевого мониторинга и аудита мониторная программа (МП), исполняющая следующие основные функции:

– получение оперативной информации о состоянии ЛВС и нарушениях в сети от наблюдающей станции (НС) по serial-соединению;

– агрегирование оперативной информации о состоянии КВС за текущий 15-минутный период отчёта НС и за последние сутки;

– установление связи по сетевому протоколу со специальной серверной программой-приложением и передача на сервер оперативной информации для её отражения на сайте (подробнее описано ниже);

– автоматическое исполнение процедуры изоляции сетевого нарушителя посредством исполнения соответствующей процедуры в сеансе связи с сетевым коммутатором Cisco Catalyst-2935;

– возможность исполнять нужные процедуры управления сетевым коммутатором вручную, по запросу оператора.

Схема взаимодействия компонентов системы сетевого мониторинга и аудита отражена на рис. 26. По сравнению с предложенной на стадии начала проекта схемой, действующая отличается местом подключения ПК группы Информационной безопасности к КВС: в реальной схеме, как это видно на рисунке, коммутация выполнена через вышестоящий по отношению к управляемому сетевой коммутатор, так что **команды автоматического управления всегда приходят по транковому порту и, следовательно, ненаблюдаемы ни с какого ПК ЛВС ЦЭМИ РАН**.

Работа наблюдающей станции в целом, за исключением обнаружения нарушителей и передачи этой информации в МП, достаточно полно освещена выше. Для фиксации нарушения, разработанный ранее протокол связи по serial-соединению пополнен блоком информации с MAC-адресом нарушителя. Реализована фиксация только одного типа нарушений, а именно испускания сетевым объектом TCP- или UDP-пакета с невыделенным для него IP-адресом. Встроенные в МП средства автоматической связи с управляемым сетевым коммутатором Cisco Catalyst при обнаружении нарушения запускают процедуру изоляции нарушителя. Предусмотрено блокирование изоляции для специально выделенных сетевых объектов – к таким относился роутер административно-финансового сегмента ЛВС ЦЭМИ РАН [68] и ряд других сетевых ПК. После исполнения процедуры изоляции нарушителя, МП по

протоколу UDP передаёт информации об изоляции нарушителя на сервер. В обычном режиме, по тому же протоколу на сервер постоянно передаётся полученная от НС оперативная и агрегированная информация о состоянии ЛВС.

Экран МП (рис. 28) отражает текущее время всех компонент системы сетевого мониторинга, оперативную и агрегированную информацию о состоянии ЛВС и список текущих нарушителей. В расширенном варианте экрана (после нажатия кнопки **Show** появляется правая часть, а сама кнопка приобретает имя **Hide**) доступны средства ручного управления коммутатором и отладки. Здесь же содержится ряд последних сообщений, поступивших в протокол работы (отражение их может быть заморожено щелчком мыши по этой части экрана для спокойного изучения, это фиксируется словами "**Показ приостановлен**" и рамкой вокруг протокола, как это видно на рис. 28).

На том же рисунке видны в нижней левой части в специальном окне MAC-адреса нарушителей, отключённых от КВС. Обратившись к протоколу работы в правой части полного окна, можно видеть подробные характеристики нарушителей со временем их фактической изоляции.

Мониторная программа позволяет также исполнять ручное управление коммутатором, аналогично программе **TAMCICON**, описанной выше. Для этого достаточно установить флажок **Manual** (при этом автоматический вызов процедур изоляции будет заблокирован), выбрать из browse-списка нужную процедуру по её имени, ввести в дополнительных окнах MAC-адрес и номер порта, если это требуется, и, если введённые параметры корректны, запустить процедуру кнопкой **Start**. Ручное управление включается на 30 секунд, после чего флажок **Manual** автоматически сбрасывается (если процедура к этому времени ещё не завершилась, они продолжается). Необходимость ввода MAC-адреса и/или порта зафиксированы в описании процедуры, и оператору выдаются подсказки подсветом соответствующих окон. При вводе 16-ричных цифр MAC-адреса и 10-ных цифр номера порта исполняется контроль каждого вводимого символа, а также введённой части информации. Таким образом, технически сложное управление сетевым коммутатором сведено к легко формализуемым простым действиям в дружественной среде.

В правой верхней части экрана имеется окно, в котором можно наблюдать (по желанию Оператора) текущие пакеты serial-связи с НС, UDP-связи с сервером или Telnet-команды связи с сетевым коммутатором. На рис. 28 в этом окне видны последние выполненные команды процедуры.

Экран также содержит сведения о версии каждого программного средства компонент сетевого мониторинга и сетевую информацию, используемую в UDP-протоколе. Логин, пароль и IP-адрес сетевого коммутатора на экране не отражаются в целях безопасности. Иконка глаза в левом верхнем углу экрана показывает нормальное состояние связи с НС (открытый глаз) или таймаут (закрытый).

Выполненная изоляция объекта сразу же отражается на экране НС в режиме показа списка нод. На рис. 30 представлен вид этого экрана после выполнения удалений, отражённых на рис. 28. Жёлтый цвет, напомним, означает использование некорректного IP-адреса, красная звёздочка – выполнение изоляции. К моменту фиксации, изолированы сервер Лаборатории Лабренца (комната 203, ip=193.232.194.91) и ПК Бекларяна (комната 303, ip=193.232.194.48).

Работа системы сетевого мониторинга и аудита в её финальном варианте проверена на ЛВС ЦЭМИ РАН. Сетевой коммутатор Cisco Catalyst-2935, использующийся для управления локальной сетью, уверенно исполняет процедуру отсечения ПК, нарушающих правила эксплуатации сети. Созданные математические и программные средства позволяют использовать как STATIC-адреса сетевого коммутатора, так и SECURE-списки.

Следует отметить, что теоретически изоляция по SECURE-спискам может быть выполнена двумя способами: с указанием и без указания порта нарушителя. Дело в том, что список подсоединённых устройств на каждом порту сопровождается в каждой строке номером интерфейса (см., напр., такой список в выдаче коммутатора на рис. 25 по команде **show port-sec inter fa0/16 address**). Процедура может запросить список содержимого *всех* портов, и среди них определить нужный. Однако, в реальной схеме изоляции всё же использована процедура с указанием порта, поскольку ещё при начальном проектировании таблицы устройств в НС было предусмотрено указание порта. Кроме того, вывод таблицы всех подключённых устройств, хотя и более надёжен, занимает много места и длиннее на несколько минут по времени исполнения.

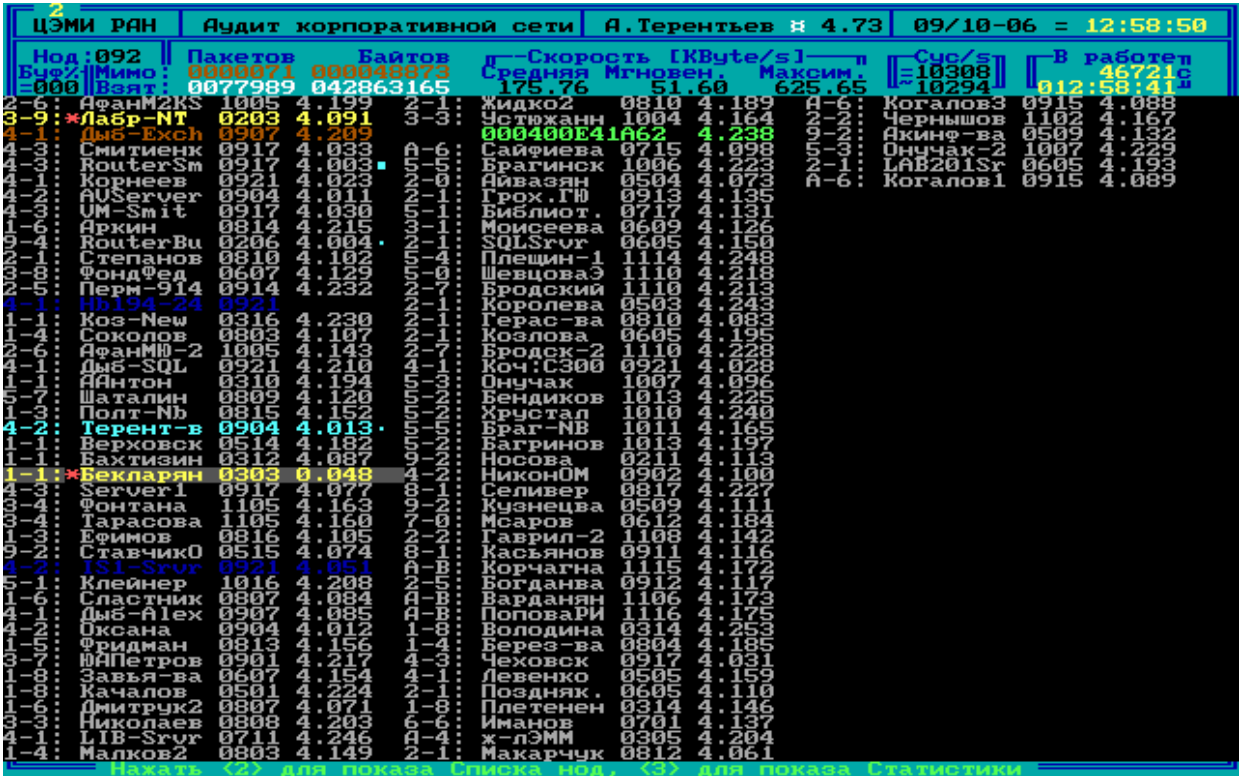


Рис. 30. Экран НС после удаления двух нод

Вся процедура изоляции занимает менее минуты, обычно 10-15 секунд. За время эксплуатации описанных средств разрывов связи с сетевым коммутатором во время исполнения процедур изоляции не наблюдалось.

Автору к моменту публикации данной работы неизвестны аналоги описанных программно-технических средств автоматической изоляции ПК, нарушающих правила работы в корпоративной сети.

Получение работоспособной системы, представляется, не означает завершение работ по совершенствованию средств сетевого мониторинга и аудита. Ряд вопросов решён в минимальном объёме: в данной модификации отслеживаются далеко не все типы сетевых нарушений, которые могут представлять интерес. Не исследовано управление коммутаторами, имеющими смешанный режим STATIC- и SECURE-адресов по разным портам. Отсутствует работа с несколькими сетевыми коммутаторами. Имеются и другие возможные направления развития работ.

Вместе с тем, автор считает поставленную в начале разработки проекта задачу автоматической изоляции сетевых нарушителей успешно выполненной.

2.3. Передача информации мониторной программы в Интернет

Достигнутые успехи в создании мониторной программы и наличие в составе группы Информационной безопасности (впоследствии – Антивирусной службы ЦЭМИ РАН) Антивирусного сервера естественным образом поставили вопрос о передаче сводной информации по КВС ЦЭМИ РАН в Интернет.

Для решения этой задачи представлялось необходимым выполнить следующее:

– создать необходимое программное обеспечение для сервера и осуществить передачу информации в связке клиент-сервер для фиксации на сервере агрегированных данных МП;

– создать необходимое программное обеспечение для показа агрегированных данных, принятых от МП, в сети Интернет.

Поскольку данные агрегации МП показываются на экране с периодом в 3 секунды, следовало выбрать схожий интервал для изменения показателей, демонстрируемых в сети Интернет. Чтобы уйти от необходимости синхронизации процессов, был выбран интервал в 5 секунд.

Выбор протокола связи между серверной программой и МП был остановлен на UDP. Мини-сервер UDP внутри серверной программы излучает broadcasting на порту 150353[12] до тех пор, пока не получит ответ от МП. После простого handshake устанавливается устойчивое соединение, в процессе которого после инициирующего запроса от сервера МП шлёт очередную порцию данных. Отсутствие очередного пакета с любой стороны возвращает ситуацию в поиск handshake.

Однако, после первых эксплуатационных успехов, возникло неожиданное препятствие. Дело в том, что Windows-приложение на сервере способно работать только в случае, когда произведён вход Администратора в систему [69]. В то же время, в большинстве случаев сервера в целях информационной безопасности эксплуатируются без выполнения подобного входа. Естественным выходом в этой ситуации является перевод серверной программы в статус **системной службы**. В этом режиме, однако, программа не может самостоятельно открывать GUI-окна для управления ею, и приходится дополнительно иметь программу управления системной службой [70], включаемой в работу только после исполнения процедуры logon Администратором. Программа системной службы названа TAM LAN Server Station, соответственно программа управления – TAM LAN Server Control.

Системная служба, периодически получая информацию по UDP от МП, сбрасывает её на диск в определённый файл. Далее, эта информация используется программой управления для показа на экране, если она запущена, а также блоком демонстрации информации в Интернете. Чтобы не перегружать файловую систему сервера постоянными операциями ввода/вывода, для хранения данных использован виртуальный диск в оперативной памяти сервера системы Qsoft.

[12] Дата рождения автора.

Преимущество такой схемы состоит в том, что системная служба начинает работу сразу же при загрузке сервера, до выполнения процедуры Logon [73]. Программа управления системной службой может быть не загружена, что не мешает работе системной службы и исполнению передачи оперативных данных на сервер. Таким образом, функционирование сервера уже не зависит от места его установки: при размещении сервера в специально выделенном месте процедура Logon, как правило, не выполняется.

Общая структура системной службы как консольного приложения освещена в [70]. Там же показана работа с Service Control Manager (SCM) операционной системы Microsoft Windows Server 2003 R2 на языке PowerBASIC Console Compiler. Организация программы управления системной службой на языке PowerBASIC for Windows рассмотрена в [71], дополнительная информация приведена в [72] и [73]. В том же цикле статей приведены многочисленные примеры использования языка PowerBASIC при разработке различных приложений под Microsoft Windows, включая операции с реестром Windows.

Методы программной реализации самой системной службы и программы управления ею могут быть весьма различны. Современный отечественный стандарт этих методов предполагает использование алгоритмического языка C++. Целью данного цикла работ является, в том числе, желание автора продемонстрировать менее затратный и более элегантный путь написания подобных программ. Как системная служба, так и программа управления ею выполнены на хорошо известном мировым профессионалам-программистам языке PowerBASIC [60]. Однако, в отличие от системной службы, являющейся консольным приложением, программа управления системной службой является оконным (GUI-) приложением (компилятор PowerBASIC for Windows 9.05). Стоит пояснить, что, являясь мощной альтернативой языку среднего уровня C++, PowerBASIC предлагает язык высокого уровня, специфицированный для создания профессиональных Windows-приложений благодаря встроенной поддержке множества обращений к различным библиотекам Windows. Не всегда являясь оптимальным при создании приложений реального времени, он может быть легко дополнен при необходимости процедурами на Макроассемблере, что было показано автором на примере программирования наблюда-

ющей станции [1]. Все версии компиляторов являются платными (в настоящее время порядка $100 за каждую), лицензирование осуществляется вендором исключительно персонально для каждого пользователя.

Мониторная программа, принимающая информацию от НС, исполняется как полнооконное (GUI-) приложение на рабочем компьютере. Современный вид её окна приведён на рис. 31. Поскольку МП обменивается информацией как с НС, так и с серверной службой, есть возможность отображения в блоке служебной информации текущего времени НС и текущего времени сервера.

По сравнению с ранее описанными версиями МП, теперь на панели окна присутствуют флажки **Снимать НС при Shutdown** и **Автооткл**. Первый флажок, будучи установленным, вызывает при любом завершении работы МП посылку сигнала на НС, по которой завершается программная работа НС. Такая функция оказалась необходимой ввиду частых длительных перебоев питания (до нескольких раз в месяц) в зоне Антивирусной службы ЦЭМИ РАН. Естественно, что если компьютер с МП исполняет отключение (shutdown) по сигналу UPS, то для нормального закрытия и сохранения файлов MS-DOS на НС, её программа также должна быть корректно завершена.

Вид окна программы управления серверной службой с интервалом в полминуты от рис. 31 приведён на рис. 32. Желающие могут сравнить выданные характеристики загрузки сети.

Легко заметить, что существенно отличаются только текущие мгновенные скорости в сети. Уже второй и третий столбцы практически совпадают. Так же близки значения секунд с начала 15-минутного периода (823 и 742) и соответствующие данные о биллингах.

Стоит несколько слов сказать о технике управления консольного приложения системной службы из управляющей системной службой GUI-окна. Фактически, это совершенно разные программы, общей области памяти у них нет, и связь возможна только с помощью специальных обращений к Windows.

Рис. 31. Современный вид окна МП

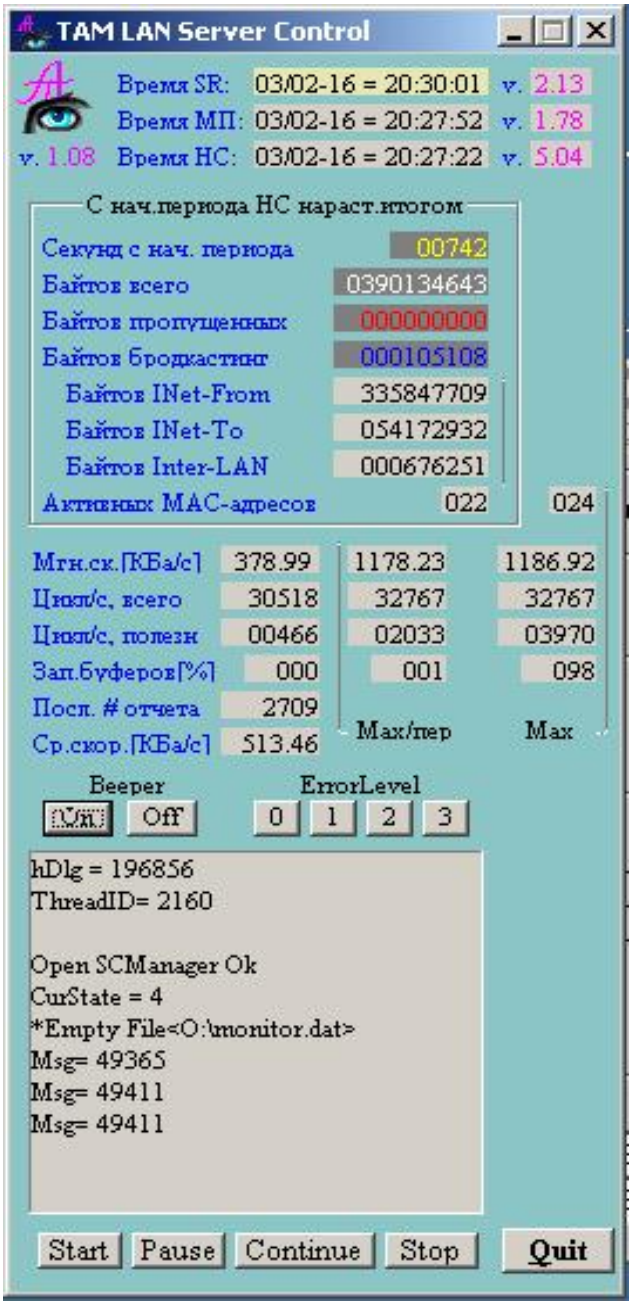


Рис. 32. Окно управляющей программы сервера

Для передачи информации от GUI-программы управления системному сервису (службе) на языке PowerBASIC for Windows используется аппарат ControlService в виде:

fOk=ControlService (hService, 201, ss) где

fOK – результат операции, должен быть **TRUE**;

ControlService – встроенная в компилятор PowerBASIC for Windows процедура обращения для передачи информации системной службе;

hService – идентификационный номер службы;

201 или иной – зарезервированный код информации для передачи службе;

ss – Service Status после исполнения операции.

При успешной операции, программе системной службы поступает сигнал с указанным номером (201 или иным). Вот так легко исполняется передача информации на языке высокого уровня PowerBASIC.

Несколько сложнее передать информацию от системной службы программе управления. Для этого используется реестр Windows. У каждой системной службы существует своя ветвь реестра, в которой допускается иметь произвольные именованные переменные нужного типа (например, символьного). В нужный момент (после прихода новых значений от МП и записи их в файл) выставляется значение переменной Signal в **1**. Управляющая программа в своём основном цикле постоянно проверяет эту ветвь реестра, и как только увидит значение **1**, сбрасывает его в **0** и, считав из файла новые данные, воспроизводит их на экране.

Остаётся осветить непосредственную подачу данных в Интернет. Как уже было сказано, системная служба, получив по UDP пакет данных от МП, сбрасывает его на виртуальный диск. Используется файл **O:\monitor.dat**. Открывая окно в браузере для показа сводных данных, используется файл R4MON.CGI, содержимое которого показано на рис. 33. Это – простенькая программа на Perl, считывающая указанный файл и выводящая структурированное окно агрегации данных, изображённое на рис. 34. Каждые 5 секунд операция повторяется с обновлением данных, до закрытия пользователем окна.

Версия 2.13 консольной программы системной службы **TAM LAN Server Station** зарегистрирована в ФИПС 21.05.2014 г под номером 2014617172.

Версия 1.08 GUI-программы управления **TAM LAN Server Control** зарегистрирована в ФИПС 21.05.2014 г под номером 2014617162.

```
#r4mon.cgi
###############
#!/perl/bin/perl
$LOCK_EX=2;
$LOCK_UN=8;
$datafile="O:/monitor.dat";
#########
 open(DATA,"+<$datafile");
 flock(DATA,$LOCK_EX);
 $Dat=<DATA>;
 chop($Dat);
 flock(DATA,$LOCK_UN);
 close(DATA);
####### Распределяем список по элементам и убираем сигнальный символ
 @Var=split(/,/,$Dat,30);
 $Var[6]=substr($Var[6],2);
print "Content-Type: text/html; charset=windows-1251\n\n";
print "<HTML><HEAD>\n";
print '<meta http-equiv="Pragma" content="no-cache">'; print "\n";
print '<meta http-equiv="Refresh" content="5">'; print "\n";
……
print "<TITLE>Текущий мониторинг ЛВС ЦЭМИ РАН</TITLE>\n";
print "<body bgcolor=#0066FF lang=RU style='font-size:10pt'><center>\n";
……
```

```
print "<TABLE cols=1 border=3 bgcolor=#00CCFF bordercolor=#0099CC>\n";
print "<TR><TD>\n";
print "<table cols=3 border=3 bgcolor=#99CCFF bordercolor=#333399"
print " cellspacing=1 cellpadding=3 rules=rows frame=box width=100%>\n";
print "<col align=left><col align=left><col align=right>\n";
……
print "<tr><th align=left rowspan=3 width=25%><small>"
print "Текущ. время, версия</small></th><th><small>Сервер</small></th>";
print '<td style="color:#006600; font-weight:bold; font-size:10pt">';
print "$Var[0]"; print "</td>\n";
……
print "$Var[20]"; print "</td>\n";
print "</table>\n";
print "</TABLE></CENTER></BODY></HTML>\n";
```

Рис. 33. Текст CGI-программы, отображающей сводные данные ЛВС

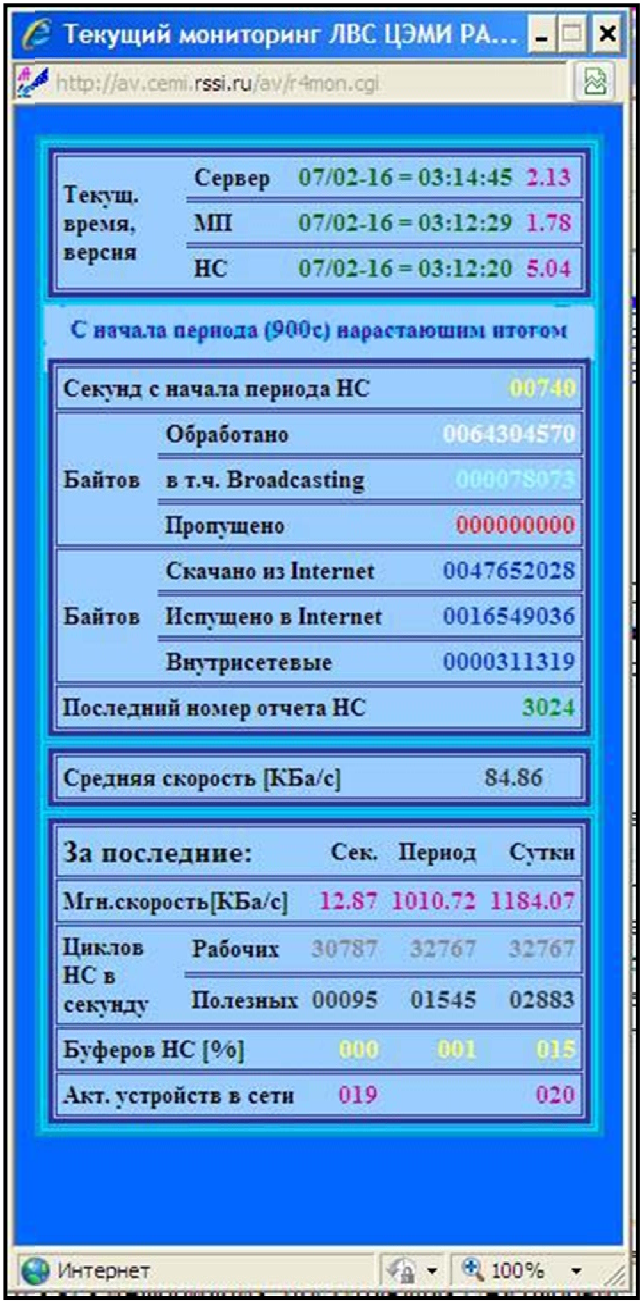


Рис. 34. Вид окна агрегации на Антивирусном сервере

ЗАКЛЮЧЕНИЕ

Том 3 данной работы посвящён использованию созданного профессионального варианта сетевого анализатора, основанного на обычном ПК и имевшемся в продаже сетевом адаптере. Работая в круглосуточном режиме и наблюдая все имеющие хождение пакеты в интересуемом участке локальной вычислительной сети, снабжённый процедурами, организующими круглосуточную работу, сетевой анализатор канального уровня является по сути средством постоянного сетевого мониторинга объектов ЛВС.

Ранее было показано, что с помощью синтезированных средств на ПЭВМ в MS-DOS возможно отслеживание некорректных сетевых пакетов, в частности, с некорректным исходным MAC-адресом, испускаемых главным образом с заражённых вирусами компьютеров.

С помощью разработанных ранее средств сетевого анализатора пакетов исследованы реальные особенности протокола Telnet на стандартном коммутаторе Cisco Catalyst-2935.

Разработан язык описания процедур управления сетевым коммутатором с возможностью приёма информации из стандартной выдачи коммутатора и использованием её в последующих командах управления коммутатором.

Разработана и опробована программа полуавтоматического управления коммутатором, позволяющая малоквалифицированным пользователям осуществлять ряд типовых процедур управления коммутатором, в частности, добавление и исключение пользовательских MAC-адресов в списках доступа коммутатора.

Доработана Мониторная программа, исполняемая на соседнем с Наблюдающей станцией Windows-компьютере и получающая от неё информацию по serial-кабелю, в том числе агрегированную информацию о наблюдаемой сети и MAC-адреса нарушающих соглашения сети компьютерах. Новая версия Мониторной программы снабжена блоком связи по Telnet-протоколу с коммутатором и интерпретатором созданного языка процедур управления коммутатором.

Таким образом, **решена задача автоматической изоляции поражённых сетевыми вирусами компьютеров от локальной сети института**.

Разработана концепция связи Мониторной программы, агрегирующей данные сетевой Наблюдающей станции, с действующим

Антивирусным сервером по Интернет-протоколу UDP. Создана системная служба на Антивирусном сервере, принимающая данные Мониторной программы и представляющая их в Интернете. Создана управляющая программа для этой системной службы. Таким образом, **созданы средства, представляющие агрегированные данные наблюдения в сети Интернет**.

Успешное внедрение предложенной технологии позволяет оперативно, без вмешательства человека, за короткое время изолировать заражённые ПК и/или сервера от локальной сети института.

Предложенная технология работы с сетевым мониторингом опробована в ЦЭМИ РАН и ряде сторонних организаций, в первую очередь из числа входящих в корпоративную сеть ЦЭМИ РАН.

Все разработки выполнены на соответствующих диалектах языка высокого уровня – PowerBASIC for DOS [61], PowerBASIC Console Compiler [62] и PowerBASIC for Windows [60].

Антивирусный сайт ЦЭМИ РАН [74] в разделе «Литература» включает все опубликованные работы автора с 1988 г. и многие работы коллег Отделения экономической информатики ЦЭМИ РАН по тематике информационной безопасности, в том числе тома данной монографии.

Следует отметить, что работа автора над пакетом сетевого анализатора и связанных с ним программ, не была лёгкой. На Учёном совете ЦЭМИ РАН в 2001 г после доклада о начальном этапе разработки, соответствующем [1], Председатель Учёного совета, директор института ЦЭМИ РАН академик В.Л. Макаров произнёс итоговую фразу: «Мы всего этого не понимаем, нам это не интересно, и Вы к нам по этому поводу более не обращайтесь». Естественно, автор так и сделал. Таким образом, суть и достигнутые конечные цели разработки остались для большинства сотрудников института неизвестными. Автор надеется, что публикация трёхтомной монографии восполнит в какой-то степени этот пробел.

ПЕРЕЧЕНЬ РИСУНКОВ

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Терентьев А.М. Сетевой мониторинг. Методы и средства. Том 1: монография / А.М. Терентьев. – Чебоксары: ИД «Среда», 2019 – 116 с. – ISBN 978-5-6042304-9-7.

2. Терентьев А.М. Методы и средства наблюдения загрузки локальных вычислительных сетей на примере ЦЭМИ РАН / Препринт #WP/2001/110 – М., ЦЭМИ РАН, 2001, – 74с. – ISBN 5-8211-0141-7.

3. Терентьев А.М. Сетевой мониторинг. Развитие и применения. Том 2: монография / А.М. Терентьев. – Чебоксары: ИД «Среда», 2020. – 108 с. – ISBN 978-5-907313-11-8.

4. Archive of CERT general posting, CERT Advisory CA-2001-19. “Code Red” Worm Exploiting Buffer Overflow in IIS Indexing Service DLL, https://www.cs.ait.ac.th/laboratory/CERT/CA101/msg00017.html

5. CERT Advisory CA-2001-26 “Nimda Worm”. URL: https://sedists.org/cert/2001/22

6. Cisco Systems, Inc. Internetworking Technology Handbook, 4-th Edition. / Indianapolis: CiscoPress, September 2003. ISBN 1587051192.

7. Терентьев А.М. Мониторинг корпоративной сети ЦЭМИ РАН в условиях использования коммутатора Cisco Catalyst / А.М. Терентьев, Н.Г. Ляпичева, Н.А. Кочетова // Развитие и использование средств сетевого мониторинга и аудита: сб. статей / под ред. А.М. Терентьева – М.: ЦЭМИ РАН, 2004. – Вып. 1 – С. 75-87.

8. Терентьев А.М. Информационная безопасность в крупных локальных сетях / А.М. Терентьев // Концепции. – 2002. – №1(9). – С. 25-30. Свидетельство Роскомпечати 014305.

9. URL: http://securityresponse.symantec.com/avcenter/venc/data/linux.sorso.html

10. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.d.worm.html

11. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.kazmor.html

12. Кочетова Н.А. Методы и средства защиты магистральных маршрутизаторов и серверов удаленного доступа производства Cisco Systems / Н.А Кочетова, Н.Г Ляпичева // Вопросы информационной безопасности узла Интернет в научных организациях: сборник статей / под ред. М.Д. Ильменского – М: ЦЭМИ РАН, 2001. – С. 10-42. (Рус.)

13. Терентьев А.М. Задачи полноценного аудита корпоративных сетей / А.М. Терентьев // Концепции. – 2003. – №1 (11). – С. 94-95. – Свидетельство Роскомпечати 014305.

14. An Ethernet Address Resolution Protocol, or Converting Network Protocol Address to 48.bit Ethernet Address for transnission on Ethernet Hardware. – Network Working Group, Request for Comments 826. – November, 1982.

15. Telnet Protocol Specifications. – Network Working Group, RFC 854. – May, 1983.

16. Telnet Option Specifications. – Network Working Group, RFC 855. – May, 1983.

17. Telnet Binary Transmissions. – Network Working Group, RFC 856. – May, 1983.

18. Telnet Echo Option. – Network Working Group, RFC 857. – May, 1983.

19. Telnet Suppress Go Ahead Option. – Network Working Group, RFC 858. – May, 1983.

20. Telnet Status Option. – Network Working Group, RFC 859. – May, 1983.

21. Telnet Timing Mark Option. – Network Working Group, RFC 860.

22. Telnet Extended Options: List Option. – Network Working Group, RFC 861.

23. Telnet End of Record Option. – Network Working Group, RFC 885.

24. TACACS User Identification. Telnet Option. – Network Working Group, RFC 927.

25. Output Marking Telnet Option. – Network Working Group, RFC 933.

26. Telnet terminal Location Number Option. – Network Working Group, RFC 946.

27. Telnet 3270 regime option. – Network Working Group, RFC 1041.

28. Telnet Data Entry Terminal Option: DODIIS Implementation. – Network Working Group, RFC 1043.

29. Telnet X.3 PAD Option. – Network Working Group, RFC 1053.

30. Telnet Window Size Option. – Network Working Group, RFC 1073.

31. Telnet Terminal Speed Option. – Network Working Group, RFC 1079.

32. Telnet Terminal-Type Option. – Network Working Group, RFC 1091. – February, 1989.

33. Telnet X display location Option. – Network Working Group, RFC 1096.

34. Telnet Subliminal-Message Option. – Network Working Group, RFC 1097.

35. The Q Method of Implementing Telnet Option Negotiation. – NW Group, RFC 1143.

36. Telnet Linemode Option. – Network Working Group, RFC 1184.

37. Telnet Remote Flow Control Option. – Network Working Group, RFC 1372.

38. Telnet Authentification: Kerberos. Version 4. – Network Working Group, RFC 1411.

39. Telnet Authentification SPX. – Network Working Group, RFC 1412.

40. Telnet Environment Option Interoperability Issues. – Network Working Group, RFC 1571.

41. Telnet Environment Option. – Network Working Group, RFC 1572.

42. Telnet CHARSET Option. – Network Working Group, RFC 2066.

43. Telnet Com Port Control Option. – Network Working Group, RFC 2217.

44. Telnet KERMIT Option. – Network Working Group, RFC 2810.

45. 250 Telnet Enchancements. – Network Working Group, RFC 2877.

46. Telnet Authentification Option. – Network Working Group, RFC 2941.

47. Telnet Authentification: Kerberos Version 5. – Network Working Group, RFC 2942.

48. Telnet Authentification Using DSA. – Network Working Group, RFC 2943.

49. Telnet Authentification: SRP. – Network Working Group, RFC 2944.

50. Telnet Data Encription Option. – Network Working Group, RFC 2946.

51. Telnet Encription: DES 64 bit CipherFeedback. – Network Working Group, RFC 2947.

52. Telnet Encription: DES 64 bit Output Feedback. – Network Working Group, RFC 2948.

53. Telnet Encription: CAST-128 64 bit Output Feedback. – Network Working Group, RFC 2949.

54. Telnet Encription: CAST-128 64 bit Output Cipher Feedback. – NW Group, RFC 2950.

55. Telnet Authentification: Using KEA and SKIPJACK. – Network Working Group, RFC 2951.

56. Telnet Encription: DES64 bit Cipher Feedback. – Network Working Group, RFC 2952.

57. Telnet Encription: DES64 bit Output Feedback. – Network Working Group, RFC 2953.

58. Терентьев А.М. Возможность полуавтоматического управления сетевыми коммутаторами Cisco Catalyst / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 2 / под ред. А.М. Терентьева. – М., ЦЭМИ РАН, 2005. – С. 14-27. – ISBN 5-8211-0365-7.

59. Терентьев А.М. Опыт синтеза языка управления работой сетевых коммутаторов Cisco / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 3 / под ред. А.М. Терентьева. – М., ЦЭМИ РАН, 2006. – С. 16-35. – ISBN 5-8211-0409-2 (978-5-8211-0409-0).

60. PowerBASIC for Windows [Электронный ресурс]. – Режим доступа: https://www.powerbasic.com

61. Zale, Robert S. PowerBASIC Compiler, version 3. User's Guide. – PowerBASIC, Inc. 316 Mid Valley Center. Carmel, CA 93923. – 335 с.

62. Zale, Robert S. PowerBASIC Compiler, version 3. Reference Guide. – PowerBASIC Inc. 316 Mid Valley Center. Carmel, CA 93923. – 335 с.

63. Терентьев А.М. Мониторная программа как средство интеграции данных наблюдающей станции в локальной сети / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 2 / под ред. А.М. Терентьева. – М., ЦЭМИ РАН, 2005. – С. 6-13. – ISBN 5-8211-0365-7.

64. Терентьев А.М. Автоматическая изоляция некорректных объектов КВС по информации сетевого мониторинга / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 3 / под ред. А.М. Терентьева. – М.: ЦЭМИ РАН, 2006. – С. 6-15. – ISBN 5-8211-0409-2 (978-5-8211-0409-0).

65. Терентьев А.М. Построение и развитие системы сетевого мониторинга / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 1 / под ред. А.М. Терентьева. – М., ЦЭМИ РАН, 2004. – С. 5-23. – ISBN 5-8211-0317-7.

66. Терентьев А.М. Антивирусная защита ПК в Windows 95/98/NT: справочное пособие по антивирусным средствам ЗАО "ДиалогНаука". 2-е изд. / А.М. Терентьев – М.: Перспектива, 2000. – 104 с. – ISBN 5-86225-490-0.

67. Терентьев А.М. Ускорение форматных преобразований в системах реального времени, реализованных на языке PowerBASIC для i386+ / А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 1 / под ред. А.М. Терентьева – М.: ЦЭМИ РАН, 2004. – С. 24-36. – ISBN 5-8211-0317-7.

68. Вегнер В.А. Разработка и реализация типового проекта выделенного сегмента ЛВС на примере ПК административно-финансовой группы ЦЭМИ РАН / В.А. Вегнер, Н.Г. Ляпичева, А.С. Львова, А.М. Терентьев // Развитие и использование средств сетевого мониторинга и аудита: сб. статей. Вып. 1/ под ред. А.М. Терентьева. – М.: ЦЭМИ РАН, 2004. – С. 88-101. – ISBN 5-8211-0317-7.

69. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Системная служба / А.М. Терентьев // Стратегии устойчивого развития мировой науки: материалы XXVII Международной научной конференции. №5 (27). Т. 1. – М.: Евразийское научное объединение, 2017. – С. 41-46. – ISSN 2411-1899.

70. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Управление системной службой / А.М. Терентьев // Интеграция науки в современном мире: материалы XXVIII Международной научной конференции. №6 (28). Т. 1. – М.: Евразийское научное объединение, 2017. – С. 28-33. – ISSN 2411-1899.

71. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Результаты / А.М. Терентьев // Теоретические и практические вопросы современной науки: материалы XXIX Международной научной конференции. №7 (29). Т. 1. – М.: Евразийское научное объединение, 2017. – С. 27-31. – ISSN 2411-1899.

72. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Дополнительные сведения / А.М. Терентьев // Научные аспекты современных исследований: материалы XXX Международной научной конференции. N8 (30). Т. 1. – М.: "Евразийское научное объединение", 2017. – С. 37-41. – ISSN 2411-1899.

73. Microsoft Windows 2000 Server. Справочник администратора. 2-е изд. / пер. с англ. – М.: СП ЭКОМ, 2003. – 1360 с.: ил. ISBN 5-9570-0002-7.

74. Антивирусный сайт ЦЭМИ РАН [Электронный ресурс]. – Режим доступа: http://av.cemi.rssi.ru

Для заметок

Научное издание

Терентьев Александр Макарович

СЕТЕВОЙ МОНИТОРИНГ.
АВТОИЗОЛЯЦИЯ НЕКОРРЕКТНЫХ ОБЪЕКТОВ СЕТИ.
ТОМ 3

Монография
Чебоксары, 2021 г.

Редактор *А.М. Терентьев*
Компьютерная верстка *Л.С. Миронова*
Дизайн обложки *Н.В. Фирсова*

Подписано в печать 30.03.2021 г.
Дата выхода издания в свет 15.04.2021 г.
Формат 60×84/16. Бумага офсетная. Печать офсетная.
Гарнитура Times. Усл. печ. л. 6,045. Заказ К-805. Тираж 500 экз.

Издательский дом «Среда»
428005, Чебоксары, Гражданская, 75, офис 12
+7 (8352) 655-731
info@phsreda.com
https://phsreda.com

Отпечатано в Студии печати «Максимум»
428005, Чебоксары, Гражданская, 75
+7 (8352) 655-047
info@maksimum21.ru
www.maksimum21.ru

ISBN 978-5-907411-18-0

Терентьев Александр Макарович (р.1953) окончил МИЭМ в 1978 г. по специальности «Прикладная математика». С 1980 – зав. Отделом СМО ИВЦ Моспродснаба, с 1982 – ст. научный сотрудник ведущих отраслевых институтов угольной, затем лесной промышленности страны. Кандидат технических наук (1988). Ph.D. Европейской академии Информатизации (Брюссель, 2003). Работает в ЦЭМИ РАН с 1988; в настоящее время – ведущий научный сотрудник Лаборатории программного обеспечения сетевых информационных технологий. Области научных интересов – проблемы информационной безопасности пользователей, рабочих станций и компьютерных сетей. Основной научной деятельностью А.М.Терентьева являются возглавляемые им направления работ: антивирусные средства персональных компьютеров; мониторинг корпоративной сети.

Предложенные А.М. Терентьевым новые для ЦЭМИ РАН направления работ по тематике информационной безопасности были доложены на Научно-технических Советах ЦЭМИ РАН (1999, 2000, 2010 гг.) и на Учёном Совете ЦЭМИ РАН в 2001 г.

Под руководством А.М. Терентьева создана и успешно функционирует Антивирусная служба ЦЭМИ РАН (2001 г.). Для обеспечения антивирусными средствами ПК и серверов ЛВС ЦЭМИ РАН создана и внедрена постоянно совершенствуемая технология корпоративной поддержки пользователей антивирусного пакета DrWeb. Число пользователей антивирусных средств, централизованно поддерживаемых Антивирусной службой, выросло с 15 (1999 г.) до 200 (2008 г.).

Универсальные средства наблюдения трафика корпоративной сети, предложенные в качестве нового направления работ ЦЭМИ РАН в 1999 г., были в необходимом объёме реализованы по гранту РФФИ 04-07-90260 «Система комплексного аудита и мониторинга корпоративной сети» (2004–2006). Реализация выполнена с использованием низкоуровневого круглосуточного сетевого мониторинга с автоматическим отсечением сетевых объектов, нарушающих работу сети.

Автором опубликовано 83 научные работы (2021 г). Он регулярно участвует в Программах Президиума РАН, был руководителем этапов госконтракта с Правительством Москвы (2007–2008 гг.). Ряд программных продуктов А.М. Терентьева зарегистрированы в ФИПС (2012–2014 гг.). Работы с его участием многократно отмечались в числе лучших работ Отделения (1999, 2000, 2003, 2005, 2010, 2018 гг.). А.М.Терентьев награждён медалью в память 850-летия Москвы, а также почётной грамотой Президента РАН (2013 г.).

История становления А.М. Терентьева как программиста описана на страницах «Виртуального компьютерного музея» (http://www.computer-museum.ru/articles/prgtales/1606/).

Интересы исследователя не ограничены узким кругом профессиональных работ. А.М. Терентьев являлся редактором первого в России специализированного журнала «КомпАс» по компьютерным играм (1994 г.): http://www.computer-museum.ru/games/compas.htm. Он также выступил модератором конференции FIDONet RU.GAME.STRATEGY (1996 г.). Помимо этого А.М. Терентьев – автор Help'ов к ранним версиям популярных отечественных антивирусных программ Doctor Web для Windows и SplDer Guard (2001 г.).

На Антивирусном сайте ЦЭМИ РАН (http://av.cemi.rssi.ru) в разделе «Литература» аккумулированы все печатные работы автора, начиная с 1998 г.

Издательский дом «Среда»
Делитесь знаниями
в среде профессионалов!