Федеральное агентство по образованию
Уральский государственный технический университет – УПИ

Серия «Программно-аппаратные средства обеспечения информационной безопасности»

А.Н. Андрончик, В.В. Богданов, Н.А. Домуховский, А.С. Коллеров, Н.И. Синадский, Д.А. Хорьков, М.Ю. Щербаков

Защита информации в компьютерных сетях

Практический курс

Рекомендовано Уральским региональным отделением Учебно-методического объединения высших учебных заведений РФ по образованию в области информационной безопасности для межвузовского использования в качестве ***учебного пособия*** *для студентов высших учебных заведений, обучающихся по специальностям 090102 — «Компьютерная безопасность», 090105 — «Комплексное обеспечение информационной безопасности автоматизированных систем», 090106 — «Информационная безопасность телекоммуникационных систем»*

Под редакцией к.т.н., доцента Н.И. Синадского

Екатеринбург
УГТУ–УПИ
2008

УДК 681.3.067
ББК 32.973
А 66

Рецензенты:

кафедра компьютерной безопасности и прикладной алгебры
ГОУ ВПО Челябинский государственный университет
(д-р физ.-мат. наук, проф. А. В. Рожков);

Д-р тех. наук, проф. А. А. Захаров
(кафедра информационной безопасности
ГОУ ВПО Тюменский государственный университет).

**Андрончик А. Н., Богданов В. В., Домуховский Н. А.,
Коллеров А. С., Синадский Н. И., Хорьков Д. А., Щербаков М. Ю.**

А 66 ЗАЩИТА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ.
ПРАКТИЧЕСКИЙ КУРС: учебное пособие / А. Н. Андрончик,
В. В. Богданов, Н. А. Домуховский, А. С. Коллеров, Н. И. Синадский,
Д. А. Хорьков, М. Ю. Щербаков; под ред. Н. И. Синадского.
Екатеринбург : УГТУ-УПИ, 2008. 248 с.

ISBN 978-5-321-01219-2

Учебное пособие раскрывает вопросы практического применения методов и средств защиты информации в компьютерных сетях. В пособии рассмотрены способы анализа сетевого трафика, сетевые атаки и их обнаружение, организация защищенных виртуальных сетей, вопросы межсетевого экранирования, применения технологий терминального доступа, организации служб каталогов, аудита безопасности компьютерных сетей. Основной акцент в пособии делается на практическое изучение материала.

Учебное пособие предназначено для студентов вузов, обучающихся по специальностям 090102 – Компьютерная безопасность, 090105 – Комплексное обеспечение информационной безопасности автоматизированных систем, 090106 – Информационная безопасность телекоммуникационных систем, при изучении дисциплины «Программно-аппаратные средства обеспечения информационной безопасности».

Пособие будет полезно преподавателям, слушателям потоков повышения квалификации по направлению информационной безопасности, а также специалистам-практикам в области защиты компьютерной информации.

Библиогр.: 22 назв. Рис. 153. Табл. 14.

УДК 681.3.067
ББК 32.973
ISBN 978-5-321-01219-2

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ACK – Acknowledgement field significant
AD – Active Directory
AH – Authentication Header
ARP – Address Resolution Protocol
CDP – Cisco Discovery Protocol
CIFS – Common Internet File System
CVE – Common Vulnerabilities and Exposures
CVSS – Common Vulnerability Scoping System
ESP – Encapsulating Security Payload
FTP – File Transfer Protocol
GRE – Generic Routing Encapsulation
HTTP – Hypertext Transfer Protocol
ICMP – Internet Control Message Protocol
IIS – Internet Information Services
IKE – Internet Key Exchange
IP – Internet Protocol
ISO – International Standard Organization
KDC – Key Distribution Centre
L2F – Layer-2 Forwarding
L2TP – Layer-2 Tunneling Protocol
LDAP – Lightweight Directory Access Protocol
MAC – Medium Access Control
MSCHAP – Microsoft Challenge Handshake Authentication Protocol
MSTS – Microsoft Terminal Services
NAT – Network Address Translation
NSS – Name Service Switch
NTP – Nessus Transport Protocol
PAM – Pluggable Authentication Modules
POP3 – Post Office Protocol

PPP – Point-to-Point Protocol
PPTP – Point-to-Point Tunneling Protocol
RDP – Remote Desktop Protocol
RFC – Request For Comments
RST – Reset the connection
S/MIME – Secure Multipurpose Internet Mail Extension
SHTTP – Secure HTTP
SKIP – Simple Key management for Internet Protocol
SMB – Server Message Blocks
SMTP – Simple Mail Transfer Protocol
SOA – Start of Authority
SSL – Secure Socket Layer
SYN – Synchronize sequence numbers
TCP – Transmission Control Protocol
TGT – Ticket Granting Ticket
TLS – Transport Layer Security
UDP – User Datagram Protocol
VPN – Virtual Private Network
АИС – Автоматизированная информационная система
ИП – Инструментальные проверки
МЭ – Межсетевой экран
ОС – Операционная система
ПИБ – Подсистема информационной безопасности
ПК – Персональный компьютер
ПО – Программное обеспечение
СЗИ – Средство защиты информации
СОА – Система обнаружения атак
СОИБ – Система обеспечения информационной безопасности
ЦС – Центр сертификации
ЭЦП – Электронно-цифровая подпись

СОДЕРЖАНИЕ

Введение ... 8
1. Основы захвата и анализа сетевого трафика ... 12
1.1. Общие сведения о программе ... 12
1.2. Установка программы и подготовка к захвату ... 13
1.3. Пользовательский интерфейс программы ... 15
1.4. Фильтр отображения пакетов ... 16
1.5. Поиск кадров ... 22
1.6. Выделение ключевых кадров ... 23
1.7. Сохранение данных захвата ... 23
1.8. Печать информации ... 24
1.9. Просмотр кадра в отдельном окне ... 25
1.10. Анализ протоколов Ethernet и ARP ... 25
1.11. Анализ протоколов IP и ICMP ... 28
1.12. Анализ протокола TCP ... 30
2. Выявление сетевых атак путем анализа трафика ... 34
2.1. Этапы сетевой атаки ... 34
2.2. Исследование сетевой топологии ... 34
2.3. Обнаружение доступных сетевых служб ... 39
2.4. Выявление уязвимых мест атакуемой системы ... 43
2.5. Реализации атак ... 44
2.6. Выявление атаки на протокол SMB ... 45
3. Защита компьютерной сети с использованием межсетевых экранов ... 50
3.1. Понятие межсетевого экрана ... 50
3.2. Компоненты межсетевого экрана ... 51
3.3. Политика межсетевого экранирования ... 55
3.4. Архитектура МЭ ... 56
3.5. Пример реализации политики МЭ ... 58
3.6. Сетевая среда лабораторной работы ... 61
3.7. Применение МЭ на основе двудомного узла ... 62
3.8. Применение МЭ на основе фильтрующего маршрутизатора ... 64
3.9. Применение МЭ на основе экранирующего узла ... 70
3.10. Применение технологии трансляции сетевых адресов ... 72

4. Системы обнаружения атак....76
4.1. Сигнатурный анализ и обнаружение аномалий....76
4.2. Обнаружение в реальном времени и отложенный анализ....79
4.3. Локальные и сетевые системы обнаружения атак....80
4.4. Распределенные системы обнаружения атак....81
4.5. Система обнаружения атак Snort....82
5. Организация виртуальных частных сетей....94
5.1. Задачи, решаемые VPN....94
5.2. Туннелирование в VPN....96
5.3. Уровни защищенных каналов....97
5.4. Защита данных на канальном уровне....99
5.5. Организация VPN средствами протокола PPTP....103
5.6. Защита данных на сетевом уровне....108
5.7. Организация VPN средствами СЗИ VipNet....114
5.8. Использование протокола IPSec для защиты сетей....125
5.9. Организация VPN средствами СЗИ StrongNet....130
5.10. Защита на транспортном уровне....135
5.11. Организация VPN средствами протокола SSL в Windows Server 2003....137
5.12. Организация VPN прикладного уровня средствами протокола S/MIME и СКЗИ КриптоПро CSP....144
6. Применение технологии терминального доступа....150
6.1. Общие сведения о технологии терминального доступа....150
6.2. Обеспечение безопасности ОС Windows Server 2003....152
6.3. Настройки сервера MSTS....161
6.4. Настройки протокола RDP....163
7. Службы каталогов....173
7.1. Общие сведения о службах каталогов....173
7.2. Структура каталога LDAP....177
7.3. Система единого входа в сеть на основе протокола Kerberos....184
7.4. Создание единого пространства безопасности на базе Active Directory 193
8. Аудит информационной безопасности компьютерных систем....198
8.1. Понятие аудита информационной безопасности....198
8.2. Методика проведения инструментальных проверок....202
8.3. Постановка задачи для проведения инструментальных проверок....204

8.4. Обнаружение сетевых узлов 207
8.5. Сканирование портов и идентификация ОС 213
8.6. Использование DNS для обнаружения и выяснения назначения сетевых узлов 216
8.7. Создание карт сети 219
8.8. Использование сканера безопасности Nessus 222
8.9. Анализ защищенности web-серверов 233
Список литературы 239
Перечень программного обеспечения 241
Приложение 1 Применение технологии виртуальных машин для имитации сетевых соединений 242
Приложение 2 Перечень сетевых служб ОС Windows Server 2003 244

ВВЕДЕНИЕ

Принципы и методы защиты информации в компьютерных сетях подробно излагаются во многих источниках. Вместе с тем ощущается недостаток пособий, в которых защита в компьютерных сетях была бы описана в виде минимально необходимом и в то же время достаточном для практического освоения основных принципов защиты на примере доступного программного обеспечения.

Развитие компьютерных сетей, интеграция локальных сетей в одну общую сеть приводят к росту происшествий и атак. Для защиты от атак человечество придумывает все новые механизмы, и в то же самое время хакерские атаки становятся все более и более изощренными. Однако основные принципы сетевой защиты сформулированы уже достаточно давно и остаются неизменными. В пособии мы не будем описывать изощренные технологии атак, часть из которых реализована только в теории. Соответственно не будем приводить и способы защиты от изощренных атак, так как считаем, что, разобравшись с основными принципами защиты от стандартных «классических» атак, читатель с успехом сможет освоить механизмы защиты и от более сложных. В процессе изложения будем предполагать, что защищается небольшая компьютерная сеть, в которой не обрабатывается критичная информация или информация, составляющая государственную тайну, так как требования к защите таких сетей предъявляются существенно более высокие.

Цель пособия — дать возможность читателям на практических примерах изучить способы защиты информации в небольшой компьютерной сети от стандартных сетевых атак.

По мере изложения теоретического материала читателям предлагаются практические задания, обозначенные абзацем «***ВЫПОЛНИТЬ!***». Выполнение заданий, а также получение ответов на содержащиеся в них вопросы являются необходимым условием освоения учебного материала. В процессе выполнения заданий в ряде случаев необходимо обеспечить функционирование в сети нескольких узлов, что зачастую бывает сложно организовать в компьютерных классах и невозможно на отдельно стоящем компьютере. Для изучения предлагается применять систему виртуальных машин VMware Workstation, позволяющую на одном компьютере имитировать наличие нескольких сетевых узлов.

Познакомившись с теоретической частью пособия и выполнив практические задания, читатели смогут, во-первых, обоснованно применять методы сетевой защиты в процессе своей практической деятельности, во-вторых, самостоятельно освоить те средства и системы, которые остались за рамками данного пособия.

Учебное пособие разработано на основе раздела «Защита в компьютерных сетях» дисциплины «Программно-аппаратные средства обеспечения информационной безопасности».

При проведении практических занятий применяются либо свободно распространяемые программные продукты, либо демонстрационные версии коммерческих систем.

Читаемый курс строится следующим образом. Сначала слушатели изучают особенности анализа сетевого трафика с целью выявления признаков сетевых атак. В курсе излагаются самые распространенные сетевые атаки, цель которых привести сетевые узлы в неработоспособное состояние. Чтобы не допустить причинения ущерба компьютерным системам, изучаются лишь те сетевые атаки, которые на сегодняшний день потеряли актуальность. В то же время, изучив описываемые атаки и разобравшись в их природе, слушатели самостоятельно смогут найти информацию о современных атаках в Интернет и в изданиях по компьютерной безопасности.

Получив представление о сетевых атаках, слушатели изучают меры защиты, начиная с межсетевого экранирования. *Цель занятий* — научиться защищать клиентскую сеть от внешнего проникновения. Изучаются программные фильтрующие маршрутизаторы и персональные сетевые фильтры.

Далее изучаются средства обнаружения сетевых атак, которые предназначены для выявления не только внешних атак, но и атак, исходящих из внутренней сети, например от вредоносных программ.

Большое внимание в курсе уделяется построению виртуальных частных сетей. Рассматриваются как средства, встроенные в ОС Windows 2000/XP, так и сертифицированные программные комплексы.

Отдельно в курсе изучаются способы повышения защищенности Windows-систем путем уменьшения количества функционирующих сетевых служб и упрощения их конфигурации. Рассматривается минимально необходимая для функционирования ОС конфигурация, позволяющая рабочей станции решать задачи по обеспечению, например, делопроизводства в ЛВС.

Также в курсе рассматривается перспективная с точки зрения обеспечения безопасности технология обработки конфиденциальных документов и баз данных с применением терминального доступа. Занятия проводятся на примере компонента Terminal Server, встроенного в ОС Windows Server 2003.

Завершается курс изучением вопросов аудита безопасности компьютерных систем. Демонстрируются средства, позволяющие не только найти уязвимые места в настройке сетевых узлов, но и предложить конкретные меры по их устранению. Кроме того, с целью противодействия подобному сканированию, осуществляемому извне и несанкционированно, описываются приемы, которые используют сканеры безопасности для анализа уязвимостей. Дополнительно изучаются средства, позволяющие выяснить степень соответствия используемых защитных механизмов требованиям нормативных документов, в частности стандарту ISO 17799.

Практические занятия по разделу проводятся по следующим темам:

1. Мониторинг состояния элементов сети с использованием анализаторов сетевого трафика MS Network Monitor, Ethereal.

2. Создание защищенных сегментов при работе в Интернет с использованием межсетевых экранов. Применение фильтрующих маршрутизаторов.
3. Безопасная настройка клиентского программного обеспечения. Защита рабочих станций с использованием персональных сетевых фильтров.
4. Организация VPN средствами протокола PPTP в ОС Windows 2000/XP.
5. Защита сетевого трафика с использованием протокола IPSec в ОС Windows 2000/XP.
6. Применение программного комплекса ViPNet для организации виртуальной частной сети.
7. Организация VPN средствами протокола SSL в ОС Windows Server 2003.
8. Применение СОА Snort для обнаружения скрытого сканирования, атак, использующих преднамеренное нарушение структуры сетевых пакетов, атак вида «отказ в обслуживании».
9. Применение программных средств аудита информационной безопасности с целью тестирования состояния защищенности компьютерных систем от несанкционированного доступа и выработки мер защиты от выявленных угроз.

Пособие состоит из восьми глав, библиографического списка, перечня программного обеспечения и двух приложений:

− *Глава 1*. Основы захвата и анализа сетевого трафика. Рассмотрена методика работы с сетевыми анализаторами с целью определения структуры сетевых пакетов. Материал рассматривается на примере свободно распространяемой программы Ethereal. Сведения, изложенные в главе, необходимы для изучения материала последующих глав.

− *Глава 2*. Выявление сетевых атак путем анализа трафика. В главе рассматриваются основные широко известные сетевые атаки. Основной упор делается на то, чтобы научиться распознавать данные атаки в массиве сетевого трафика. Рассматриваются этапы сетевого проникновения и выявляются признаки сетевых атак.

− *Глава 3*. Защита компьютерной сети с использованием межсетевых экранов. Содержит сведения о назначении и типах межсетевых экранов. Практические задания предназначены для получения навыков настройки межсетевых экранов на основе свободно распространяемых программных средств.

− *Глава 4*. Системы обнаружения атак. Излагается теория обнаружения атак, практические приемы выявления сетевых атак изучаются на примере распространенной СОА Snort. Подробно описывается интерфейс СОА Snort и приводятся примеры правил обнаружения распространенных атак.

− *Глава 5*. Организация виртуальных частных сетей. Данная глава является наибольшей по объему и количеству практических заданий. В ней рассматриваются протоколы VPN, применяемые на различных уровнях сетевой модели. Практические задания предназначены для изучения как стандартных средств организации VPN, реализованных в ОС Windows 2000/XP, так и специализированных программных пакетов на примере широко распространен-

ной системы VipNet. Слушатели осваивают технологии построения VPN с использованием протоколов PPTP, IPSec и SSL. Раздел, в котором описывается СЗИ VipNet, не подменяет документацию по данному программному продукту и специализированные учебные курсы. В разделе приводится методика организации занятий по изучению СЗИ VipNet в компьютерных классах общего назначения с использованием технологии виртуальных машин.

− *Глава 6*. Применение технологии терминального доступа. В главе рассматривается комплексная задача по организации защищенной обработки конфиденциальной информации в АИС на базе ЛВС с применением технологии терминального доступа. В качестве основы выбирается встроенная в ОС Microsoft Windows Server 2003 служба терминального доступа. На практических заданиях читатели отрабатывают предлагаемую в пособии методику организации защиты информации в ЛВС.

− *Глава 7*. Службы каталогов. Излагаются общие сведения о назначении и реализациях служб каталогов, описывается структура популярного протокола доступа к службе каталогов LDAP, организация принципа единой регистрации в сети на основе протокола Kerberos. Практические задания предполагают подробное изучение протоколов на основе взаимодействия рабочих станций под управлением ОС Linux и серверов на основе ОС Microsoft Windows Server 2003.

− *Глава 8*. Аудит информационной безопасности компьютерных систем. В главе вводится понятие аудита информационной безопасности и трех его основных типов. В результате выполнения практических заданий читатели получают навыки выполнения важнейшей составляющей активного аудита – инструментальных проверок. Читателям предлагается провести весь комплекс инструментальных проверок – от получения первичной информации о сетевых узлах до написания итогового отчета по результатам тестирования.

Библиографический список содержит 22 наименования источников, включая техническую документацию и учебные пособия, требующиеся для углубленного изучения отдельных тем.

Главы 1 и 3 написаны А. Н. Андрончиком, глава 2 — Н. И. Синадским, глава 4 — Д. А. Хорьковым, глава 5 — Н. И. Синадским, Д. А. Хорьковым, глава 6 — А. С. Коллеровым, Н. И. Синадским, М. Ю. Щербаковым, глава 7 — Н. А. Домуховским, глава 8 — В. В. Богдановым, Н. И. Синадским, Д. А. Хорьковым.

1. ОСНОВЫ ЗАХВАТА И АНАЛИЗА СЕТЕВОГО ТРАФИКА

Мониторинг и анализ сетевого трафика являются неотъемлемой частью процесса управления компьютерной сетью и используются для диагностики, тестирования и поиска неисправностей, для оптимизации структуры информационных потоков, а также выявления и решения проблем в обеспечении безопасности узлов компьютерной сети и информации, циркулирующей между ними.

Целью данного занятия является приобретение навыков захвата сетевого трафика в сегменте локальной сети и анализа собранной информации с помощью программного анализатора протоколов Ethereal. Для успешного достижения целей занятия слушателям необходимо повторить теоретический материал, касающийся назначения и функционирования протоколов стека TCP/IP.

Для изучения материала данной главы в учебном классе должен быть развернут сегмент локальной вычислительной сети на концентраторе или коммутаторе, включающий в себя рабочие станции с операционной системой Windows 2000/XP по количеству слушателей. При выполнении некоторых упражнений понадобится наличие сервера HTTP или подключение к сети Интернет. Для установки необходимого программного обеспечения на рабочих станциях должны быть доступны инсталляционные пакеты библиотеки WinPCap (версия не ниже 2.3) и анализатора Ethereal (версия не ниже 0.10.11).

1.1. Общие сведения о программе

Существует множество инструментальных средств, предоставляющих необходимые возможности для выполнения мониторинга сети и анализа сетевого трафика. Одним из таких средств является пакет Ethereal, представляющий собой программный анализатор протоколов. Анализатор протоколов переводит сетевой адаптер в режим «беспорядочного» приема кадров, записывает в свой буфер отфильтрованные кадры сетевого трафика, по запросам пользователя выводит на экран те или иные кадры из буфера и посредством декодера протоколов предоставляет пользователю информацию о значениях полей заголовка протокола и содержимое его блока данных.

Как и большинство программ такого класса, Ethereal содержит следующие основные компоненты: фильтр захвата, буфер кадров, декодер протоколов, фильтр отображения захваченных кадров и модуль статистики с элементами экспертной системы. К несомненным достоинствам Ethereal относятся:

- наличие реализаций для Unix и Windows;
- наличие исходного кода программы;
- возможность захвата трафика в сетевых сегментах различных базовых технологий;

− возможность анализа огромного числа протоколов (более 700);
− возможность экспорта/импорта файлов данных в формат распространенных анализаторов (несколько десятков форматов);
− мощная и удобная система поиска и фильтрации информации в буфере пакетов;
− наличие элементов экспертной системы;
− возможность сохранения на диск выделенного фрагмента пакета;
− наличие полезных утилит командной строки для осуществления захвата трафика и обработки сохраненных файлов.

1.2. Установка программы и подготовка к захвату

ВЫПОЛНИТЬ!

1. Установите библиотеку WinPCap и анализатор Ethereal, для чего последовательно запустите соответствующие файлы установки.

Некоторые дистрибутивы Ethereal содержат в себе инсталлятор требуемой версии библиотеки WinPCap.

2. Запустите Ethereal и разверните главное окно приложения на весь экран (для удобства работы).

Перед выполнением захвата сетевого трафика необходимо настроить параметры захвата или проконтролировать установленные значения некоторых из них так, чтобы собранная информация адекватно соответствовала решаемой задаче анализа трафика.

ВЫПОЛНИТЬ!

3. Выполните команду меню ***Capture ⇒ Options***.

В открывшемся диалоговом окне устанавливаются следующие параметры захвата кадров (рис. 1.1):
− Interface — сетевой адаптер;

Очень важно выбрать соответствующий сетевой адаптер, иначе запись кадров будет производиться из другого сегмента сети! В компьютере, имеющем всего один сетевой адаптер, среди возможных сетевых интерфейсов часто присутствует контроллер удаленного доступа!

− Buffer size — размер буфера захвата (по умолчанию 1 Мб);

При малом размере буфера существует опасность того, что при его заполнении запись новых кадров будет производиться поверх записанных ранее!

– Capture packets in promiscuous mode — использование режима беспорядочного захвата.

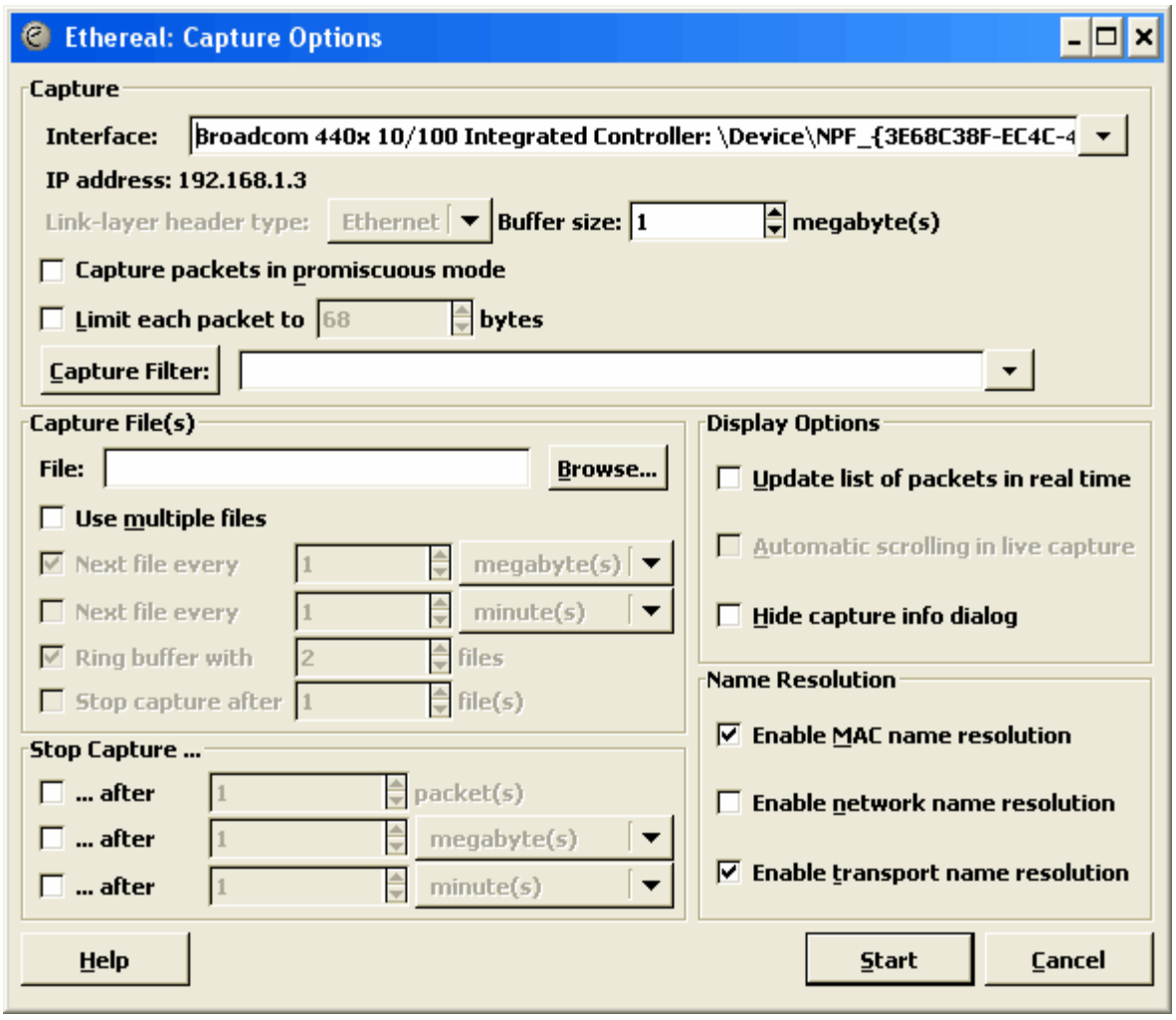


Рис. 1.1. Окно настройки параметров захвата

– Limit each packet to — запись только нескольких первых байт (определяется установленным значением параметра) каждого кадра;

– Capture Filter — фильтр захвата;

Фильтр захвата экономит объем буфера, отбрасывая «лишний мусор», однако увеличивает нагрузку на процессор, вследствие чего некоторые кадры могут быть потеряны. Поэтому в некоторых случаях вместо фильтра записи предпочтительнее использовать фильтр отображения кадров в буфере, а запись производить без фильтрации!

– Capture File(s) — файл захвата;

Опция полезна при осуществлении захвата трафика в течение длительного периода времени.

– Stop Capture — условия автоматического завершения захвата;

− Display Options — отображение пакетов в реальном времени и автоматический скроллинг окна информации;

Опции увеличивают нагрузку на процессор, вследствие чего некоторые кадры могут быть потеряны.

− Name Resolution — разрешение имен на физическом, сетевом и транспортном уровнях.

ВЫПОЛНИТЬ!

4. Уберите маркер напротив опции «Capture packets in promiscuous mode» для захвата только «своих» кадров (кадры с широковещательным адресом также будут захватываться). В таком режиме работы число захваченных пакетов будет существенно меньше, что облегчит выполнение заданий.

1.3. Пользовательский интерфейс программы

ВЫПОЛНИТЬ!

5. В командной строке сеанса MS-DOS для очистки кэша протокола ARP выполните команду `arp -d`. В Ethereal для запуска процесса захвата нажмите кнопку «Capture». В командной строке выполните команду `ping <имя_сервера>` (в качестве параметра команды можно использовать IP-адрес сервера). По завершении команды Ping остановите захват, нажав кнопку «Stop».

На экране монитора в программе Ethereal вы увидите несколько панелей с отображением сетевых пакетов, только что записанных в буфер. Общий вид окна приложения представлен на рис. 1.2. Пользовательский интерфейс программы содержит следующие компоненты:

− меню команд и панель инструментов;
− фильтр отображения пакетов;
− список пакетов в буфере;
− панель отображения декодера протоколов;
− панель отображения пакета в шестнадцатеричном коде и символах ASCII.

Панель со списком пакетов построчно отображает характеристики того или иного пакета (номер по порядку в буфере, время захвата, адреса источника и получателя, тип протокола и общая информация о нем). Перемещение по списку осуществляется с помощью мыши или клавиатуры, причем информация на двух других панелях обновляется автоматически. На панели декодера протоколов, нажимая указателем мыши на символы «+» или «–», можно отображать информацию о полях заголовков протоколов с требуемым уровнем детализации. При выборе того или иного служебного поля в заголовке оно автоматически выделяется на нижней панели, где отображается текущий пакет в шестнадцатеричном виде.

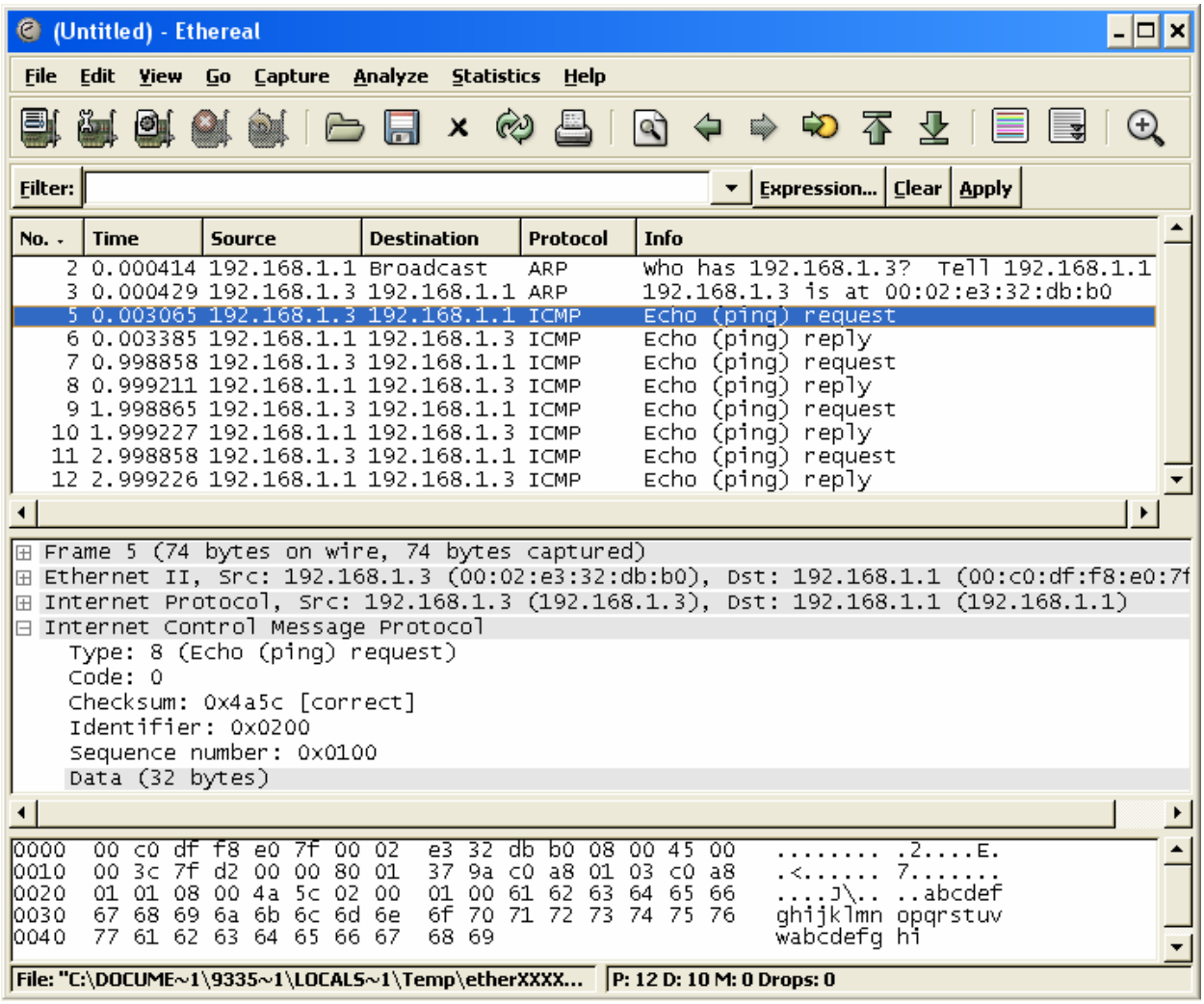


Рис. 1.2. Общий вид приложения Ethereal

1.4. Фильтр отображения пакетов

С помощью фильтра отображения можно быстро убрать «мусор». Выражение фильтрации может представлять собой просто название протокола, который присутствует в пакете на том или ином уровне вложенности. Например: arp — для отображения пакетов протокола ARP, tcp — для отображения пакетов, в которых присутствует заголовок протокола TCP.

ВЫПОЛНИТЬ!

6. Для отображения только ICMP-сообщений в строке ввода «Filter» (рис. 1.2) наберите «icmp» и нажмите кнопку «Apply».

Более сложные выражения фильтрации строятся с помощью зарезервированных слов, обычно представляющих собой названия полей заголовков того или иного протокола, знака операции сравнения и конкретного значения в шестнадцатеричном или десятичном виде. Наиболее часто используемые выражения фильтрации и их значения приведены в табл. 1.1.

Таблица 1.1

Выражения фильтрации и их значения

Выражение	**Значение выражения и пример записи**
frame.marked	Маркированный кадр `frame.marked == true`
frame.number	Номер кадра `frame.number == 150`
frame.time	Время захвата кадра `frame.time == "Feb 1, 2006 09:00:00"`
frame.pkt_len	Длина кадра `frame.pkt_len == 48`
eth.dst	Заголовок Ethernet: MAC-адрес назначения `eth.dst == 01:00:5e:00:00:02`
eth.src	Заголовок Ethernet: MAC-адрес источника `eth.src == 00:a0:cc:30:c8:db`
eth.type	Заголовок Ethernet: тип вложенного протокола `eth.type == 0x0800`
arp.hw.type	Заголовок протокола ARP: тип протокола канального уровня `arp.hw.type == 0x0001`
arp.proto.type	Заголовок протокола ARP: тип протокола сетевого уровня `arp.proto.type == 0x0800`
arp.opcode	Заголовок протокола ARP: код операции `arp.opcode == 0x0001`
arp.src.hw_mac	Заголовок протокола ARP: MAC-адрес источника `arp.src.hw_mac == 00:10:4b:30:c4:4a`
arp.src.proto_ipv4	Заголовок протокола ARP: IP-адрес источника `arp.src.proto_ipv4 == 10.1.0.1`
arp.dst.hw_mac	Заголовок протокола ARP: MAC-адрес назначения `arp.dst.hw_mac == 00:00:00:00:00:00`
arp.dst.proto_ipv4	Заголовок протокола ARP: IP-адрес назначения `arp.dst.proto_ipv4 == 10.1.0.2`
ip.version	Заголовок протокола IP: версия протокола IP `ip.version == 4`
ip.hdr_len	Заголовок протокола IP: длина заголовка `ip.hdr_len == 24`
ip.flags.df	Заголовок протокола IP: флаг фрагментации `ip.flags.df == 0`
ip.flags.mf	Заголовок протокола IP: флаг не последнего фрагмента `ip.flags.mf == 0`

Выражение	Значение выражения и пример записи
ip.frag_offset	Заголовок протокола IP: смещение фрагмента `ip.frag_offset == 0`
ip.ttl	Заголовок протокола IP: время жизни пакета `ip.ttl == 1`
ip.proto	Заголовок протокола IP: протокол вышестоящего уровня `ip.proto == 0x01`
ip.src	Заголовок протокола IP: IP-адрес источника `ip.src == 10.0.0.99`
ip.dst	Заголовок протокола IP: IP-адрес назначения `ip.dst == 224.0.0.2`
ip.addr	Заголовок протокола IP: IP-адрес `ip.addr == 10.2.0.0/16`
tcp.srcport	Заголовок протокола IP: порт источника `tcp.srcport == 1054`
tcp.dstport	Заголовок протокола IP: порт назначения `tcp.dstport == 21`
tcp.seq	Заголовок протокола IP: последовательный номер `tcp.seq == 4856133`
tcp.ack	Заголовок протокола IP: номер подтверждения `tcp.ack == 4856134`
tcp.flags.urg	Заголовок протокола IP: бит присутствия срочных данных `tcp.flags.urg == 0`
tcp.flags.ack	Заголовок протокола IP: бит присутствия подтверждения `tcp.flags.ack == 1`
tcp.flags.push	Заголовок протокола IP: бит выталкивания данных `tcp.flags.push == 0`
tcp.flags.reset	Заголовок протокола IP: бит сброса соединения `tcp.flags.reset == 0`
tcp.flags.syn	Заголовок протокола IP: бит синхронизации сессии `tcp.flags.syn == 1`
tcp.flags.fin	Заголовок протокола IP: бит завершения сессии `tcp.flags.fin == 0`
tcp.window_size	Заголовок протокола IP: размер приемного окна `tcp.window_size == 8760`
udp.srcport	Заголовок протокола UDP: порт источника `udp.srcport == 2364`
udp.dstport	Заголовок протокола UDP: порт назначения `udp.dstport == 53`

Выражение	Значение выражения и пример записи
icmp.type	Заголовок протокола ICMP: тип сообщения `icmp.type == 8`
icmp.code	Заголовок протокола ICMP: уточняющий код сообщения `icmp.code == 0x00`

В примерах записи выражений табл. 1.1 приведены выражения с операцией сравнения «Равно», которая записывается с помощью двойного знака равенства «==» (допустимо использование «eq»). Другие операции сравнения записываются с помощью следующих операторов:

a. `!=` `(ne)` — не равно, пример: `eth.type != 0x0800`;
b. `>` `(gt)` — больше, пример: `tcp.srcport > 1023`;
c. `<` `(lt)` — меньше, пример: `frame.pkt_len lt 60`;
d. `>=` `(ge)` — больше или равно, пример: `frame.pkt_len ge 60`;
e. `<=` `(le)` — меньше или равно, пример: `tcp.dstport <=1023`.

ВЫПОЛНИТЬ!

7. Выясните, что будет отображено в буфере захвата в случае использования фильтра, описанного с помощью выражений, приведенных в качестве вышеописанных примеров.

Значение любого выражения фильтрации возвращает переменную булевского типа. Таким образом, выражение `udp` означает присутствие в кадре заголовка протокола UDP, по аналогии с этим выражение `tcp.flags.syn` означает присутствие в заголовке протокола TCP бита синхронизации сессии в установленном состоянии (значение 1). К любому из выражений можно применить операцию логического отрицания, заключив его в скобки и поставив перед ним знак отрицания «NOT» или «!». Например, выражение `!(ip.addr == 10.0.0.1)` означает, что из буфера отображения необходимо убрать все пакеты, в которых встречается IP-адрес 10.0.0.1.

ВЫПОЛНИТЬ!

8. Объясните разницу между результатами использования выражений фильтрации `!(ip.addr == X.X.X.X)` и `ip.addr !== X.X.X.X`. Для выполнения упражнения в выражениях фильтрации используйте вместо адреса `X.X.X.X` реальный IP-адрес вашего узла.

В качестве выражений фильтрации можно использовать и составные выражения, которые образуются с помощью следующих логических операторов:

a. && (AND) — логическое И,
   пример: `(ip.dst==10.0.0.1) AND tcp.flags.syn`;

b. || (OR) — логическое ИЛИ,
   пример: `(ip.addr==10.0.0.1) OR (ip.addr==10.0.0.2).`

Другой удобный способ ввода выражения фильтрации состоит в следующем. На панели декодера протоколов отображается требуемое поле, в контекстном меню выбирается пункт «Apply as Filter» и далее исполняется либо команда «Selected», либо «Not Selected» в зависимости от задачи фильтрации (рис. 1.3).

ВЫПОЛНИТЬ!

9. Отобразите только ICMP-запросы (используйте поле «тип» в заголовке ICMP). Укажите результирующее выражение фильтрации с необходимыми пояснениями. После просмотра результата для отображения пакетов без фильтрации нажмите кнопку «Clear» в строке фильтра.

При необходимости создания сложного выражения фильтрации в меню «Apply as Filter» (рис. 1.3) выбирайте команды, начинающиеся с многоточия, при этом новое выражение будет добавлено к результирующему выражению фильтрации.

10. Отобразите все кадры, переданные вашим узлом, исключая сообщения ICMP.

При создании выражения фильтрации имейте в виду, что в буфере могут находиться кадры других узлов.

Укажите результирующее выражение фильтрации с необходимыми пояснениями. После просмотра результата для отображения пакетов без фильтрации нажмите кнопку «Clear» в строке фильтра.

В выражениях фильтрации первый операнд операции сравнения допускает использование указателя диапазона, если второй операнд представляет собой массив байт или строку символов. Указатель диапазона определяется с помощью квадратных скобок и может быть использован как применительно к кадру в целом (frame), так и с любым полем заголовка. Указатель диапазона допускает следующий синтаксис:

a. [i:j] начальное смещение i, длина j;
b. [i-j] начальное смещение i, конечное смещение j, включительно;
c. [i] начальное смещение i, длина 1;
d. [:j] начальное смещение 0, длина j;
e. [i:] начальное смещение i, до конца поля.

Например, записи `frame[6:3]` и `eth.src[:3]` идентичны и могут быть использованы для указания на код фирмы-производителя сетевого адаптера, передавшего кадр. Начальное смещение может иметь отрицательное значение, в этом случае оно отсчитывается от конца поля, причем последний байт поля имеет смещение, равное –1, предпоследний –2 и так далее. Напри-

мер, выражение `frame[-5:] == "hello"` определяет кадр, оканчивающийся строкой «hello».

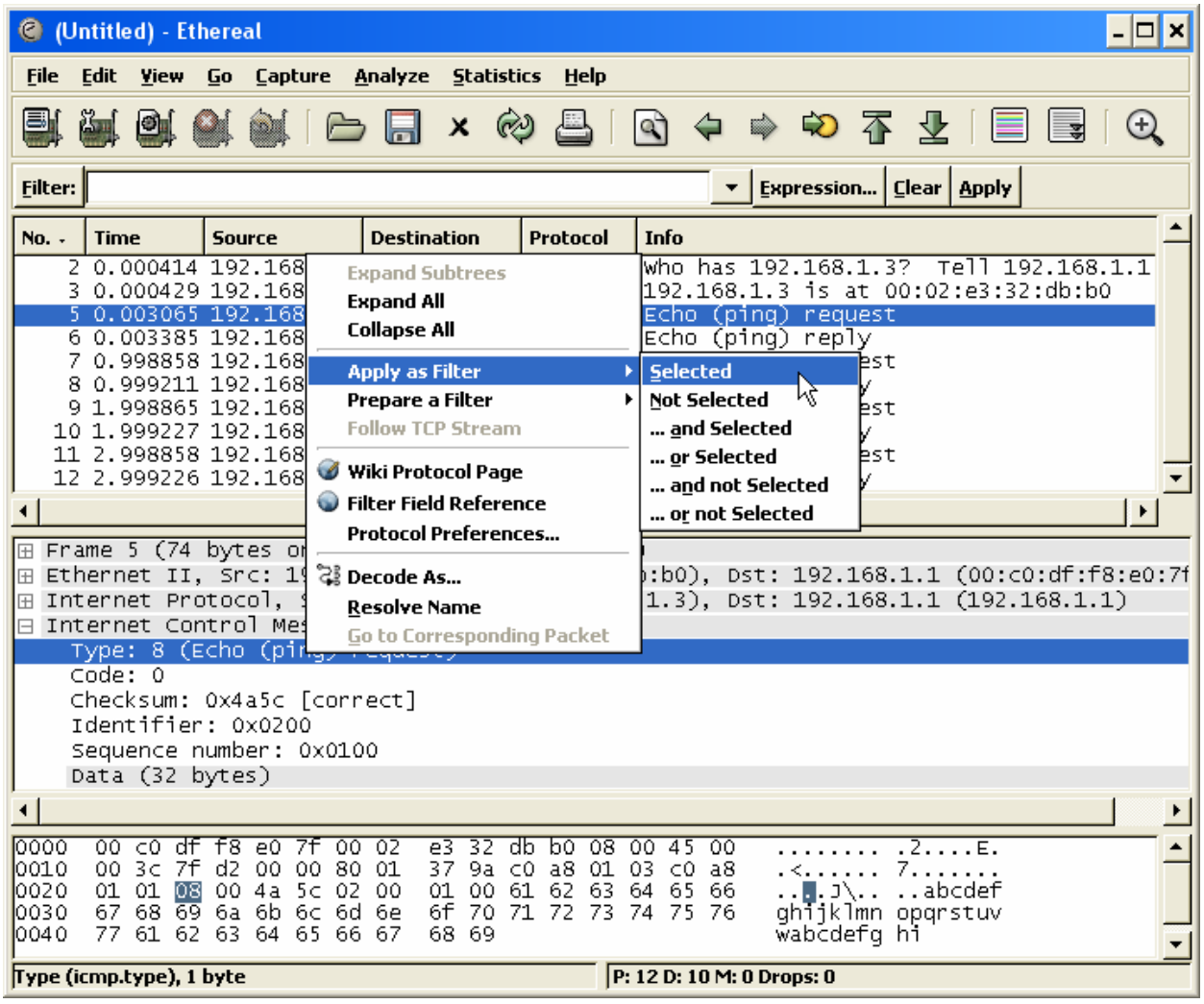


Рис. 1.3. Контекстное меню создания фильтра

Строка, как видно из предыдущего примера, записывается в кавычках. Запись массива байт осуществляется побайтно в шестнадцатеричном виде с разделителем «.» или «:», например `00.45.f5.2d`.

Используя символ «,» в указателе диапазона, можно перечислить несколько непересекающихся диапазонов, объединив их в одном операнде. Например, выражение `tcp[2,10,13-16] == 00.01.c0.f8.01.66` сравнивает в заголовке протокола TCP поле «Тип обслуживания» с «0x00», поле «Протокол» с «0x01» и поле «IP-адрес источника» с «0xc0f80166».

ВЫПОЛНИТЬ!

11. Отобразите ICMP-ответы, используя в выражении фильтрации операнд «frame» с указателем диапазона.

При создании выражения фильтрации имейте в виду, что в буфере могут находиться кадры других узлов.

Укажите результирующее выражение фильтрации с необходимыми пояснениями. После просмотра результата для отображения пакетов без фильтрации нажмите кнопку «Clear» в строке фильтра.

Быстро вернуться к тому или иному ранее вводимому выражению фильтрации можно с помощью списка истории ввода, доступ к которому осуществляется нажатием на кнопку с символом «▼», расположенную в строке фильтра (не забывайте нажимать кнопку «Apply» для применения того или иного фильтра к буферу кадров).

1.5. Поиск кадров

Поиск кадров в буфере, удовлетворяющих тем или иным критериям, осуществляется с помощью команды меню ***Edit ⇒ Find Packet***. Диалоговое окно определения критериев поиска пакетов изображено на рис. 1.4.

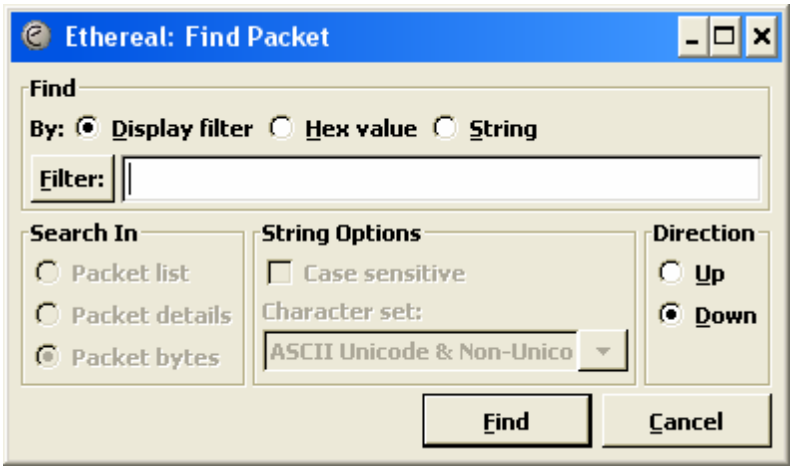


Рис. 1.4. Диалоговое окно определения критериев поиска кадров

Критерии поиска можно определять в виде выражения фильтрации (Display filter), шаблона в шестнадцатеричном виде (Hex value) и текстовой строки (String) в кодировке ASCII и (или) Unicode. В первом случае можно использовать все допустимые выражения фильтрации (табл. 1.1) и их логические комбинации. Во втором случае указывается шаблон для поиска в шестнадцатеричном коде. Поиск в строке может осуществляться в области общей информации о пакете (Packet list), в панели декодера протоколов (Packet details) и непосредственно в самом пакете (Packet bytes). Поиск может производиться вверх или вниз по списку пакетов (Direction).

Команды меню ***Edit ⇒ Find Next*** и ***Edit ⇒ Find Previous*** используются для поиска с заданными критериями следующего или предыдущего пакета соответственно.

ВЫПОЛНИТЬ!

12. Найдите все пакеты с помощью выражения фильтрации «`icmp.type==0`».
13. Найдите все пакеты по строке «reply» в области общей информации о пакете.
14. Найдите все пакеты по строке «reply» в панели декодера протоколов.
15. Проанализируйте результаты при разных вариантах поиска и дайте им объяснение.

1.6. Выделение ключевых кадров

В списке буфера ключевые или наиболее важные для дальнейшего анализа пакеты можно помечать с помощью команды ***Edit ⇒ Mark Packet*** (toggle) основного меню или команды Mark Packet (toggle) контекстного меню. Эта возможность полезна при дальнейшем поиске таких пакетов в большом буфере, так как они выделяются другим цветом, а также при сохранении, экспортировании и печати пакетов.

Информация о маркированных пакетах нигде не сохраняется, поэтому все маркеры будут потеряны при выгрузке файла данных.

ВЫПОЛНИТЬ!

16. Пометьте первый и последний пакеты, относящиеся к функционированию команды Ping.

1.7. Сохранение данных захвата

Сохранение данных в файле производится из меню ***File ⇒ Save*** или ***File ⇒ Save As***. Диалоговое окно сохранения данных изображено на рис. 1.5.

Обратите внимание, что сохранить можно все пакеты (All packets), только отображаемые (Displayed), выбранный пакет (Selected packet only), ранее маркированные с помощью основного или контекстного меню (Marked packet only и From first to last marked packet) или указанный диапазон пакетов (Specify a packet range). По умолчанию Ethereal сохраняет данные в файле типа Libpcap, совместимом по формату с файлами программы TcpDump, но путем указания определенного формата в строке ввода «File Type» этого диалогового окна данные захвата можно сохранять для экспорта в другие программы анализа трафика (около двадцати поддерживаемых в настоящее время форматов).

Не забывайте сохранять данные, прежде чем начинать другой сеанс записи.

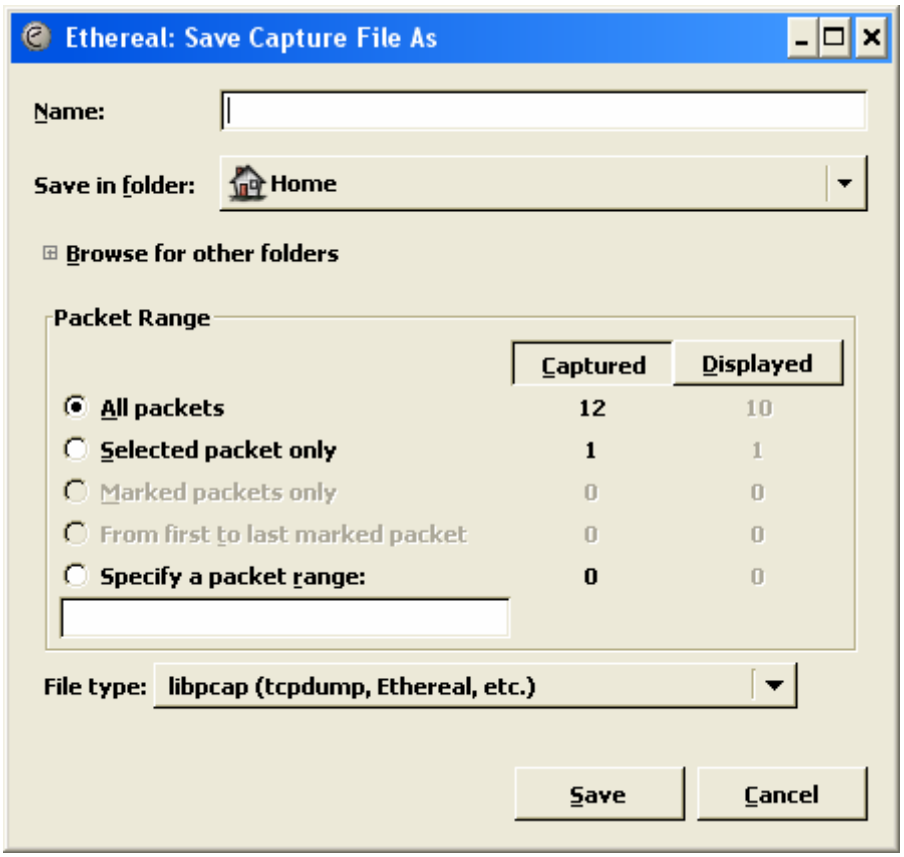


Рис. 1.5. Диалоговое окно сохранения данных

ВЫПОЛНИТЬ!

17. Сохраните все захваченные кадры в файле с именем «arp-ping» в каталоге, предлагаемом программой по умолчанию.
18. Сохраните в файле с именем «ping» только трафик команды Ping.

1.8. Печать информации

Распечатка информации о том или ином пакете или их множестве осуществляется посредством выполнения команды Print основного или контекстного меню. Диалоговое окно печати данных изображено на рис. 1.6.

При печати есть возможность осуществить вывод в указанный файл (Output to file) в виде простого текста (Plain text), определив диапазон распечатываемых пакетов (Packet Range) и формат вывода информации (Packet Format). Опции панели «Packet Range» полностью идентичны опциям соответствующей панели диалогового окна сохранения данных. При определении формата вывода в панели «Packet Format» есть возможность включить общую характеристику пакета (информацию верхней панели основного окна — «Packet summary line»), информацию, отображаемую на панели декодера протоколов с той или иной степенью детализации (Packet details) и собственно сам пакет в шестнадцатеричном виде (Packet bytes).

ВЫПОЛНИТЬ!

19. Выберите указателем мыши в списке пакетов первый ICMP-запрос и сохраните в файле «1.txt» информацию о нем с максимально возможной детализацией всех заголовков в декодере протоколов.

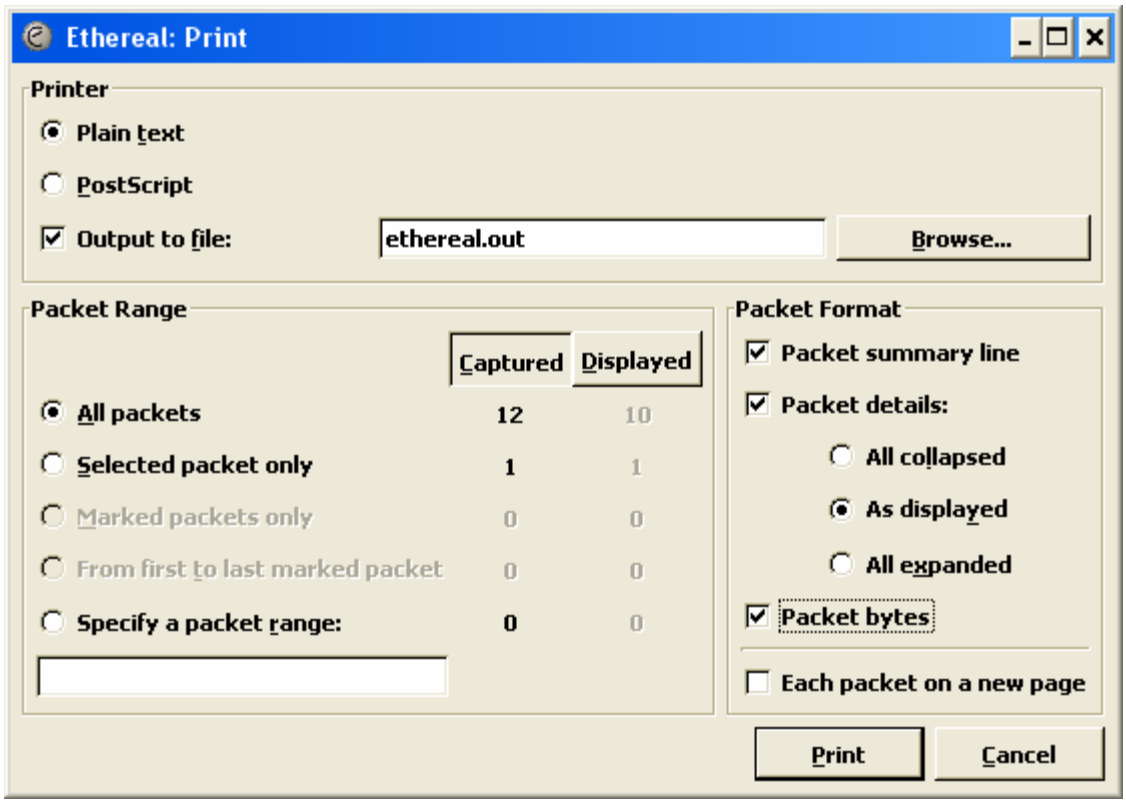


Рис. 1.6. Диалоговое окно печати данных

1.9. Просмотр кадра в отдельном окне

При составлении отчетов с использованием «скриншотов», а иногда и при анализе данных для просмотра двух пакетов одновременно удобно использовать возможность отображения пакета в отдельном окне.

Это реализуется с помощью команды «Show Packet in New Window» контекстного или основного меню программы «View». Окна, отображающие различные пакеты, показаны на рис. 1.7.

ВЫПОЛНИТЬ!

20. Пользуясь информацией об изображенных на рис. 1.7 пакетах, приведите обоснованные доводы, доказывающие их взаимосвязь.

1.10. Анализ протоколов Ethernet и ARP

При анализе протоколов Ethernet и ARP, которые находятся в иерархии протоколов ниже IP, для выключения отображения «лишней» информации на

панелях программы целесообразно отключить в программе анализ заголовка IP. Это реализуется с помощью команды «Enabled Protocols…» основного меню программы «Analyze». В диалоговом окне данной команды необходимо найти протокол IP, убрать соответствующий маркер, затем последовательно нажать кнопки «Apply» и «OK» (рис. 1.8).

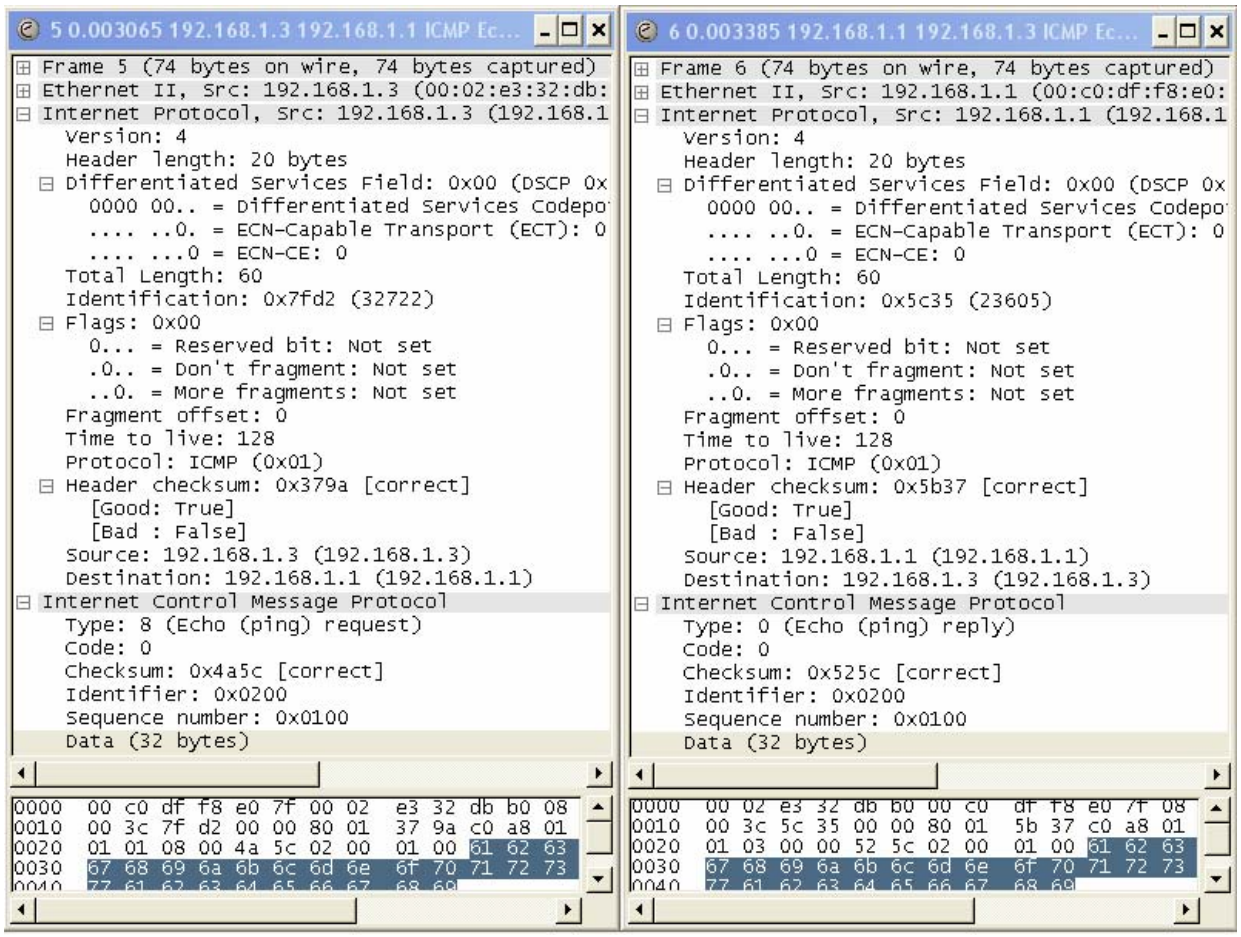


Рис. 1.7. Отображение пакетов в отдельных окнах

ВЫПОЛНИТЬ!

21. Отключите анализ заголовка IP.

В ряде случаев при отключении анализа заголовка IP отображаемые в списке буфера IP-адреса источника и получателя могут измениться!

22. Отобразите в отдельных окнах пакеты запроса и ответа протокола ARP и ответьте на следующие вопросы:
    a. Какое значение поля «тип протокола» в кадре Ethernet указывает на протокол ARP?
    b. По какому MAC-адресу отправлен запрос ARP?
    c. По какому MAC-адресу отправлен ответ ARP?
    d. Каким полем идентифицируются запрос и ответ ARP?
    e. В каких полях заголовка ARP передан запрос вашего узла?

f. В каких полях заголовка ARP передан ответ вашему узлу?

23. Загрузите созданный вами файл «1.txt» в редактор, допускающий выделение символов различным цветом.
24. Выделите различным цветом поля заголовка Ethernet в шестнадцатеричном представлении пакета.
25. Укажите, где находится поле контрольной суммы кадра Ethernet?
26. Захватите сетевой трафик вашего узла при обращении к стартовой странице поисковой системы Google и ответьте на следующие вопросы:
    a. Какие IP-адреса отображаются для узлов, участвующих в обмене по протоколу IP?
    b. Какие MAC-адреса имеют узлы, участвующие в обмене по протоколу IP?

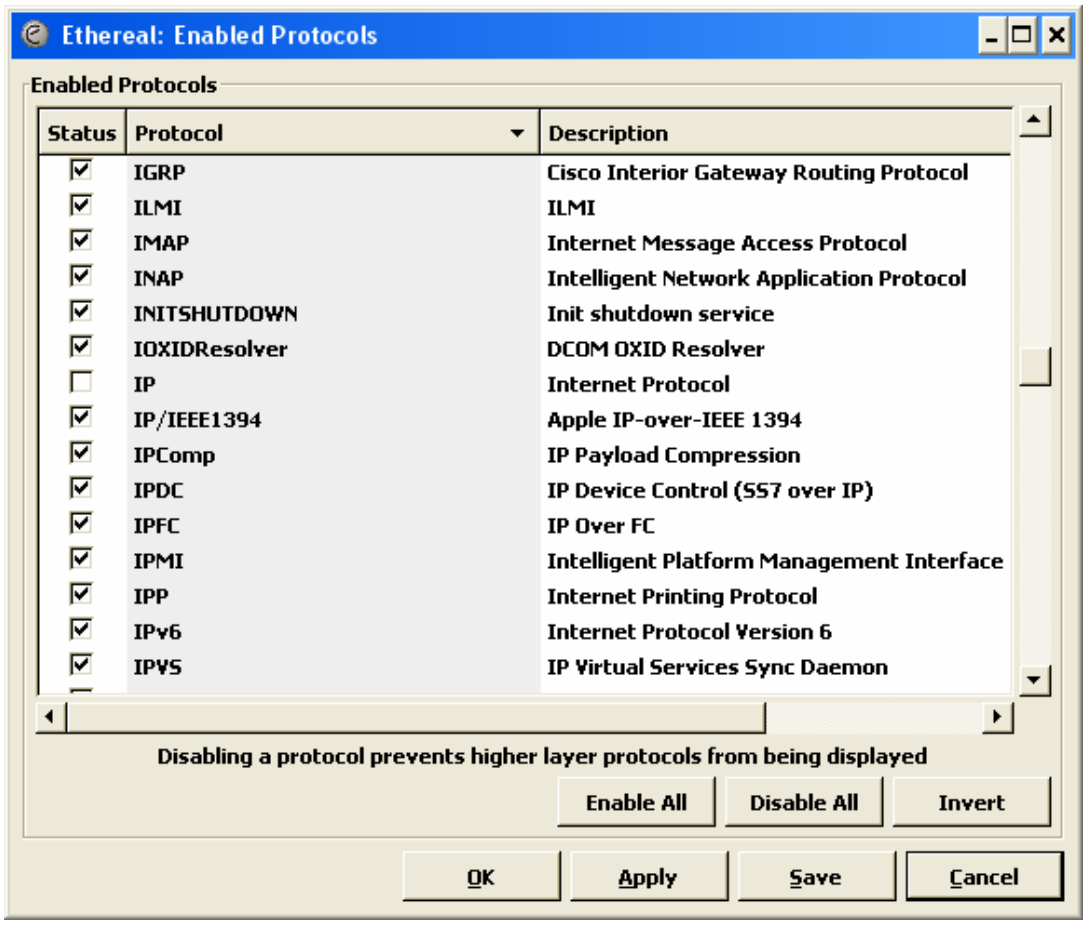


Рис. 1.8. Окно выбора протоколов для анализа

27. Включите анализ заголовка IP и ответьте на следующие вопросы:
    a. Какие IP-адреса отображаются для узлов, участвующих в обмене по протоколу IP?
    b. Какие MAC-адреса имеют узлы, участвующие в обмене по протоколу IP?
    c. Какой IP-адрес имеет узел с MAC-адресом, присутствующим во всех кадрах с протоколом IP? Какова роль этого узла?

1.11. Анализ протоколов IP и ICMP

ВЫПОЛНИТЬ!

28. Переключитесь в текстовый редактор и выделите различными цветами поля заголовка IP в шестнадцатеричном представлении пакета. Опишите назначение этих полей.
29. Загрузите созданный вами файл «ping». Сохраните два кадра «запрос — ответ» с требуемой детализацией для анализа полей ICMP и опишите назначение этих полей.
30. Проведите захват трафика команд Ping и PathPing при одинаковых значениях параметров -l и -n и проанализируйте различия в трафике этих команд.
31. Захватите сетевой трафик вашего узла при трассировке маршрута к поисковой системе Google (команда TraceRt) и ответьте на следующие вопросы:
    a. Почему MAC-адреса назначения и источника у всех кадров одинаковы и чьи это адреса?
    b. Почему узлы присылают ICMP сообщение «type 11»?
    c. Почему различные узлы присылают ICMP сообщение «type 11» на запрос к одному и тому же узлу?
    d. Сколько таких узлов, какие у них IP-адреса?
    e. Какова структура ICMP сообщения «type 11»?
    f. Какие поля ICMP одинаковы, а какие различны в последних трех запросах?
32. С помощью фильтра отобразите только ICMP-запросы. Приведите выражение фильтрации и объясните, почему выражения `icmp.type == 8` и `ip.src == X.X.X.X` (где `X.X.X.X` — IP-адрес вашего узла) не приводят к желаемому результату.
33. Ответьте на следующие вопросы:
    a. Каковы размеры кадров Ethernet, заголовков IP и сообщений ICMP, меняются ли они в процессе выполнения команды?
    b. Фрагментируются ли IP-дейтаграммы, передаваемые узлом?
    c. Какие поля заголовка IP меняются, а какие остаются неизменными в каждом пакете трафика?
    d. Какое поле заголовка IP изменяется в каждой тройке передаваемых кадров и для каких целей оно служит?
    e. Каким образом можно быстро определить число промежуточных маршрутизаторов на маршруте, если известно, что последний запрос, находящийся в буфере, достиг целевого узла?
34. С помощью фильтра отобразите только ICMP-сообщения, получаемые вашим узлом, и ответьте на следующие вопросы:
    a. Меняются ли в процессе выполнения команды размеры заголовков IP и сообщений ICMP? Чем можно объяснить данную ситуацию?

b. Имеются ли в буфере кадры, которые нельзя фрагментировать, и от какого узла они получены?

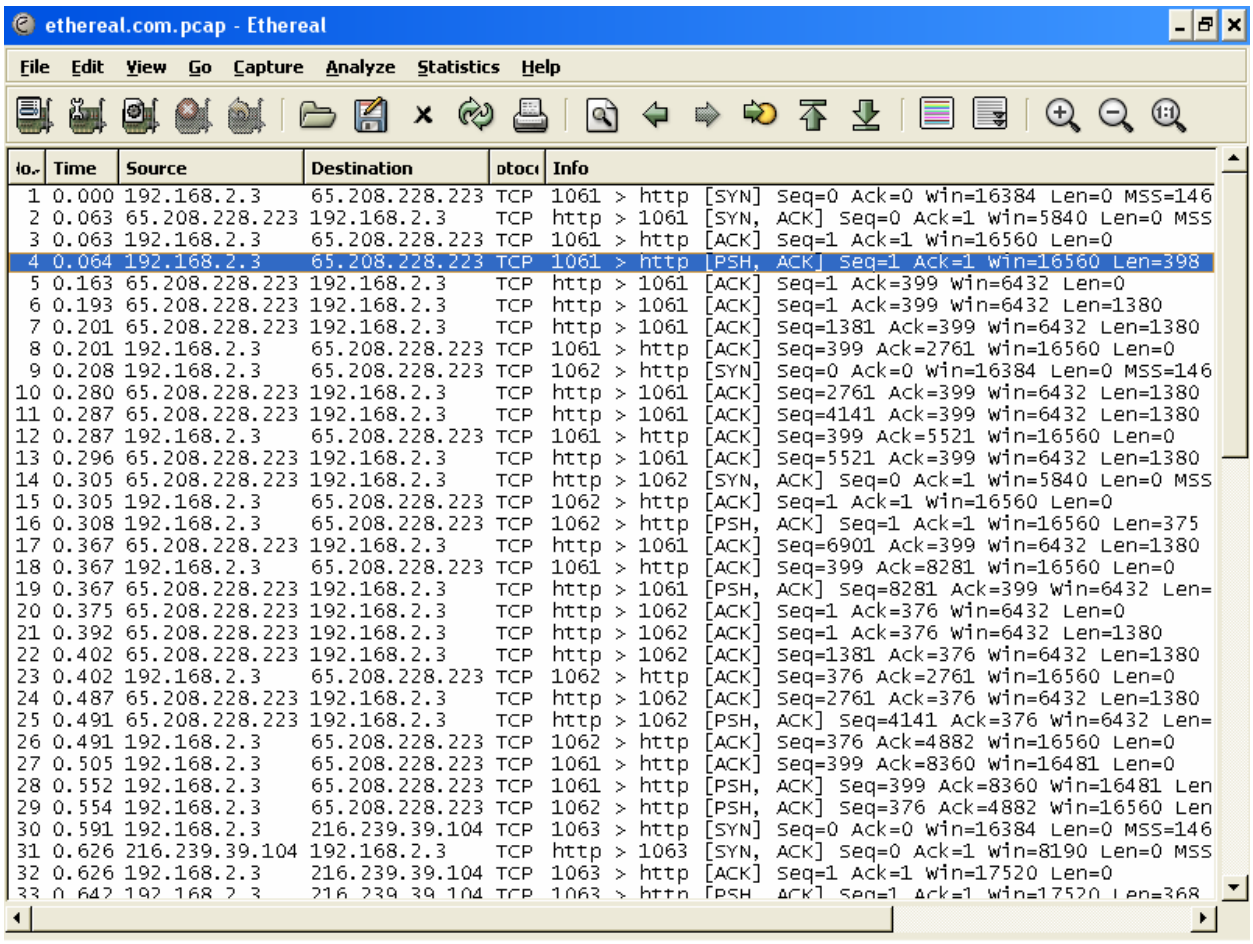
35. Захватите сетевой трафик функционирования команды Ping при проверке доступности сервера Google с параметром «r», равным 5. Сохраните данные в файле с именем «ping-r5». Ответьте на следующие вопросы:
    a. Поясните назначение параметра -r в команде Ping.
    b. Каким образом в кадрах передается информация о маршруте?
    c. Почему значение параметра -r в команде Ping не может быть больше 9?
    d. Каким образом ведут себя значения полей идентификатора и последовательного номера в заголовке ICMP захваченных кадров?
36. Захватите сетевой трафик функционирования команды Ping при проверке доступности сетевого узла вашего компьютерного класса с параметром «s», равным 4. Сохраните данные в файле с именем «ping-s4». Ответьте на следующие вопросы:
    a. Поясните назначение параметра -s в команде Ping.
    b. Каким образом в кадрах передается штамп времени?
    c. Почему значение параметра -s в команде Ping не может быть больше 4?
37. Захватите сетевой трафик функционирования команды Ping при проверке доступности сетевого узла вашего компьютерного класса с параметром «l», равным 3500, и «n», равным 1. Сохраните данные в файле с именем «ping3500». Ответьте на следующие вопросы:
    a. Сколько кадров передано и получено вашим узлом?
    b. Сколько IP-дейтаграмм передано и получено вашим узлом?
    c. Были ли IP-дейтаграммы подвергнуты фрагментации, какие поля заголовка IP указывают на это?
    d. Сколько фрагментов IP-дейтаграмм оказалось в буфере захвата?
    e. Какой размер исходной дейтаграммы, подвергнувшейся дефрагментации?
    f. Какие размеры разных фрагментов одной и той же дейтаграммы?
    g. Меняется ли идентификатор дейтаграммы в ее фрагментах, каково его значение?
    h. Какие поля заголовка IP предназначены для сборки исходной дейтаграммы из фрагментов в правильной последовательности?
    i. В каких фрагментах исходных дейтаграмм присутствует заголовок ICMP?
    j. Проанализируйте результаты и приведите схему обмена сообщениями ICMP между узлами.
38. С указанием значений всех необходимых полей заголовков покажите взаимосвязь кадров в рамках обмена «запрос — ответ» протокола ICMP.

1.12. Анализ протокола TCP

ВЫПОЛНИТЬ!

39. Захватите сетевой трафик при обращении к стартовой странице сервера www.ethereal.com. Для отображения в буфере кадров с протоколом TCP примените соответствующее выражение фильтрации.

В буфере захвата у вас находятся кадры, принадлежащие обмену клиента с сервером по протоколу HTTP, но в рамках текущего упражнения прикладной протокол нас не интересует, поэтому по аналогии с упражнением № 21 отключите анализ протокола HTTP. Фрагмент панелей со списком кадров после отключения анализа протокола FTP показан на рис. 1.9:



No.	Time	Source	Destination	Protocol	Info
1	0.000	192.168.2.3	65.208.228.223	TCP	1061 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=146
2	0.063	65.208.228.223	192.168.2.3	TCP	http > 1061 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
3	0.063	192.168.2.3	65.208.228.223	TCP	1061 > http [ACK] Seq=1 Ack=1 Win=16560 Len=0
4	0.064	192.168.2.3	65.208.228.223	TCP	1061 > http [PSH, ACK] Seq=1 Ack=1 Win=16560 Len=398
5	0.163	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=1 Ack=399 Win=6432 Len=0
6	0.193	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=1 Ack=399 Win=6432 Len=1380
7	0.201	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=1381 Ack=399 Win=6432 Len=1380
8	0.201	192.168.2.3	65.208.228.223	TCP	1061 > http [ACK] Seq=399 Ack=2761 Win=16560 Len=0
9	0.208	192.168.2.3	65.208.228.223	TCP	1062 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=146
10	0.280	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=2761 Ack=399 Win=6432 Len=1380
11	0.287	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=4141 Ack=399 Win=6432 Len=1380
12	0.287	192.168.2.3	65.208.228.223	TCP	1061 > http [ACK] Seq=399 Ack=5521 Win=16560 Len=0
13	0.296	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=5521 Ack=399 Win=6432 Len=1380
14	0.305	65.208.228.223	192.168.2.3	TCP	http > 1062 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
15	0.305	192.168.2.3	65.208.228.223	TCP	1062 > http [ACK] Seq=1 Ack=1 Win=16560 Len=0
16	0.308	192.168.2.3	65.208.228.223	TCP	1062 > http [PSH, ACK] Seq=1 Ack=1 Win=16560 Len=375
17	0.367	65.208.228.223	192.168.2.3	TCP	http > 1061 [ACK] Seq=6901 Ack=399 Win=6432 Len=1380
18	0.367	192.168.2.3	65.208.228.223	TCP	1061 > http [ACK] Seq=399 Ack=8281 Win=16560 Len=0
19	0.367	65.208.228.223	192.168.2.3	TCP	http > 1061 [PSH, ACK] Seq=8281 Ack=399 Win=6432 Len=
20	0.375	65.208.228.223	192.168.2.3	TCP	http > 1062 [ACK] Seq=1 Ack=376 Win=6432 Len=0
21	0.392	65.208.228.223	192.168.2.3	TCP	http > 1062 [ACK] Seq=1 Ack=376 Win=6432 Len=1380
22	0.402	65.208.228.223	192.168.2.3	TCP	http > 1062 [ACK] Seq=1381 Ack=376 Win=6432 Len=1380
23	0.402	192.168.2.3	65.208.228.223	TCP	1062 > http [ACK] Seq=376 Ack=2761 Win=16560 Len=0
24	0.487	65.208.228.223	192.168.2.3	TCP	http > 1062 [ACK] Seq=2761 Ack=376 Win=6432 Len=1380
25	0.491	65.208.228.223	192.168.2.3	TCP	http > 1062 [PSH, ACK] Seq=4141 Ack=376 Win=6432 Len=
26	0.491	192.168.2.3	65.208.228.223	TCP	1062 > http [ACK] Seq=376 Ack=4882 Win=16560 Len=0
27	0.505	192.168.2.3	65.208.228.223	TCP	1061 > http [ACK] Seq=399 Ack=8360 Win=16481 Len=0
28	0.552	192.168.2.3	65.208.228.223	TCP	1061 > http [PSH, ACK] Seq=399 Ack=8360 Win=16481 Len
29	0.554	192.168.2.3	65.208.228.223	TCP	1062 > http [PSH, ACK] Seq=376 Ack=4882 Win=16560 Len
30	0.591	192.168.2.3	216.239.39.104	TCP	1063 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=146
31	0.626	216.239.39.104	192.168.2.3	TCP	http > 1063 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS
32	0.626	192.168.2.3	216.239.39.104	TCP	1063 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
33	0.642	192.168.2.3	216.239.39.104	TCP	1063 > http [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=368

Рис. 1.9. Отображение информации о протоколе TCP

Обратите внимание, что теперь по каждому захваченному кадру приводится информация, касающаяся только протокола TCP. Например, для пакета № 4 (рис. 1.9) запись «1061> ftp» означает порты источника и назначения, «[PSH, ACK]» — установленные биты флагов, «Seq=1» — последовательный номер, «Ack=1» — номер подтверждения, «Win=16560» — размер приемного окна, «Len=398» — размер пересылаемого блока данных.

Каждая TCP-сессия (причем при обращении к одной странице сессий может быть несколько!) начинается с обмена тремя TCP-сегментами с установленными битами SYN, SYN-ACK и ACK. На рис. 1.9 можно видеть открытие трех сессий TCP (кадры с номерами 1, 2, 3; 9, 14, 15; 30, 31, 32 соответственно).

ВЫПОЛНИТЬ!

40. Определите количество сеансов TCP в буфере захваченных пакетов.

На рис. 1.9 также видно, что сеансы TCP начинаются с относительных последовательных номеров, равных нулю. Для того чтобы отобразить реальные последовательные номера, выбранные узлами при взаимодействии, необходимо выполнить команду меню ***Edit ⇒ Preferences***, в появившемся диалоговом окне (фрагмент диалогового окна см. на рис. 1.10) выбрать протокол TCP и убрать маркер в строке параметра «Relative sequence numbers and window scaling».

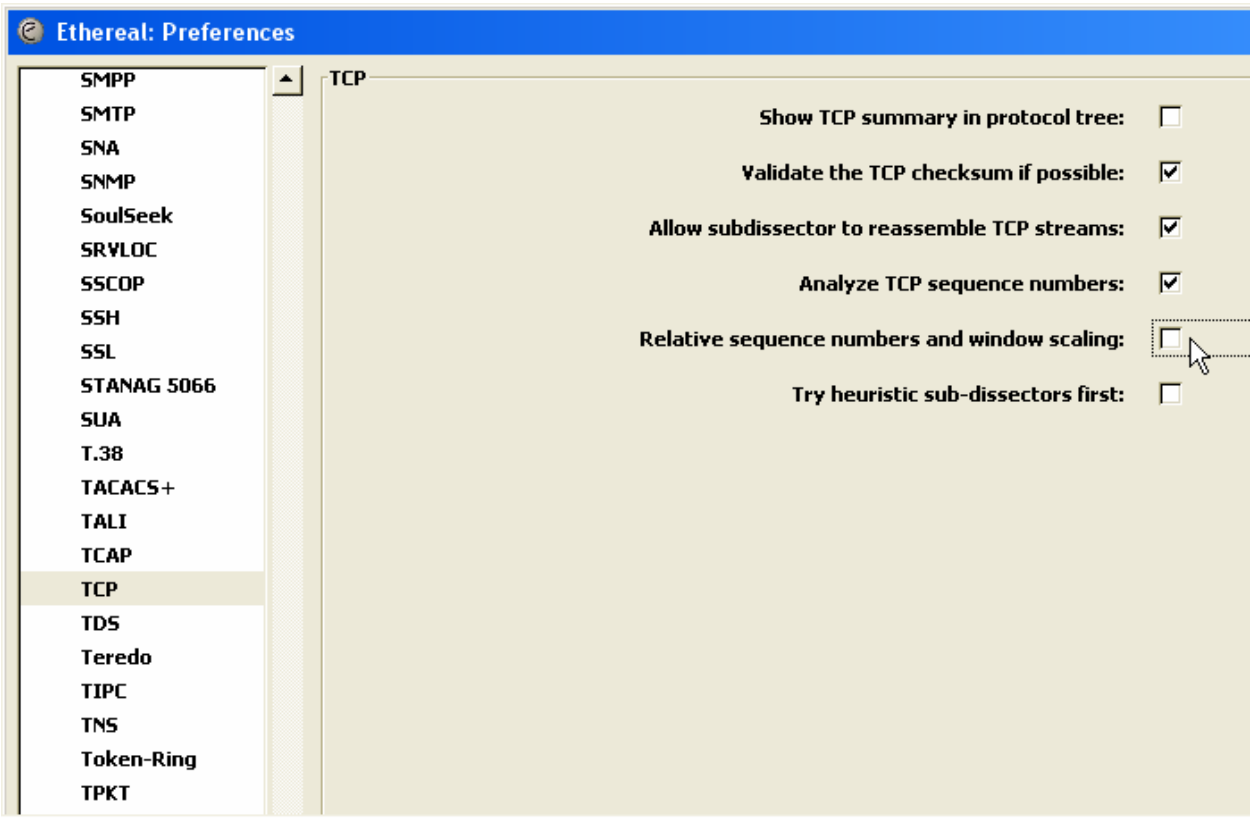


Рис. 1.10. Параметры анализа протокола TCP

ВЫПОЛНИТЬ!

41. Отобразите реальные последовательные номера в рамках сеансов TCP.
42. Проанализируйте третий кадр в рамках какого-либо сеанса TCP и ответьте на следующие вопросы:
    a. Какие порты используются клиентом и сервером?

b. Какой начальный последовательный номер выбран клиентом?
c. Присутствует ли в этом кадре поле подтверждения, каково его значение?
d. Какая длина заголовка TCP, присутствуют ли данные в этом кадре?
e. Какой бит флагов установлен и для чего он служит?
f. Какие дополнительные опции TCP передаются клиентом в этом кадре?
g. Сохраните кадр и выделите различным цветом поля заголовка TCP, пояснив их назначение.

Немаловажная возможность программы Ethereal по анализу TCP трафика состоит в том, что с помощью команды меню ***Statistics ⇒ Conversations*** можно быстро определить все сеансы, имеющиеся в буфере. В диалоговом окне для отображения сеансов TCP необходимо выбрать закладку TCP (рис. 1.11).

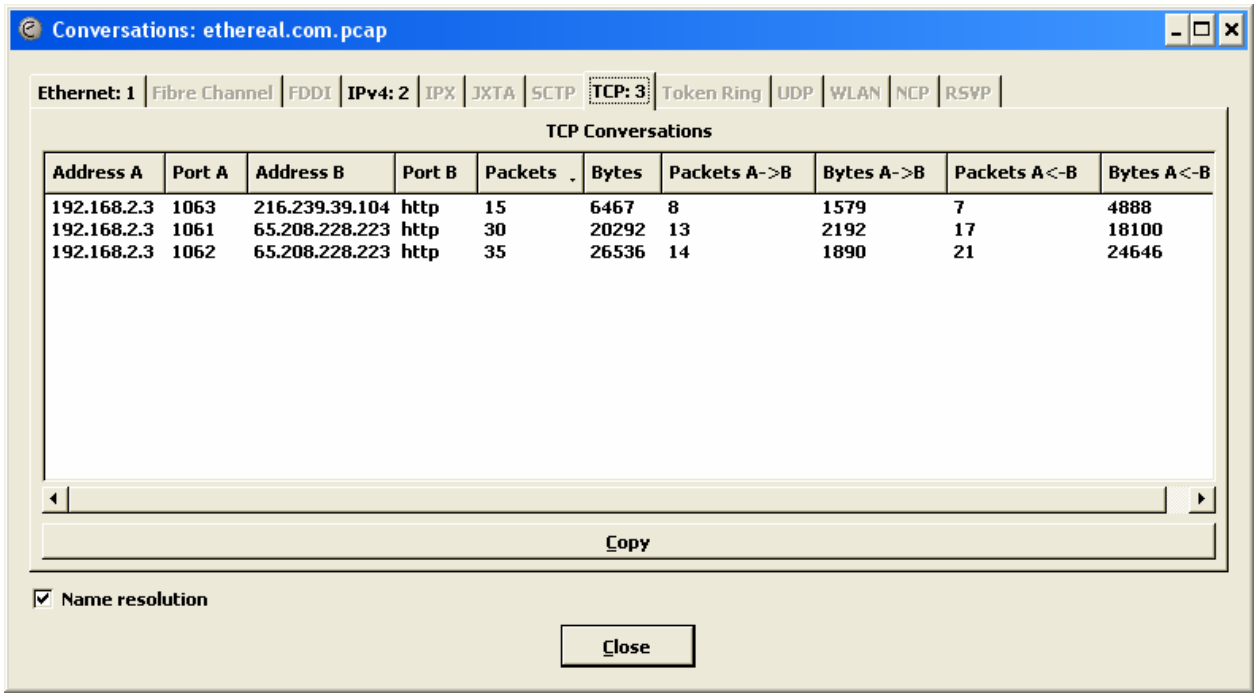


Рис. 1.11. Статистика по сеансам TCP

ВЫПОЛНИТЬ!

43. Отобразите статистику сеансов TCP.
44. Выберите первый сеанс и с помощью контекстного меню ***Apply as Filter ⇒ Selected ⇒ A<—>B*** отобразите в буфере кадры, принадлежащие этому сеансу.

Для того чтобы быстро просмотреть передаваемые данные в рамках того или иного сеанса, используют команду меню Analyze ⇒ Follow TCP Stream. После выполнения команды на экране появится диалоговое окно, в котором

разными цветами будут отображены как запросы клиента, так и ответы сервера (рис. 1.12).

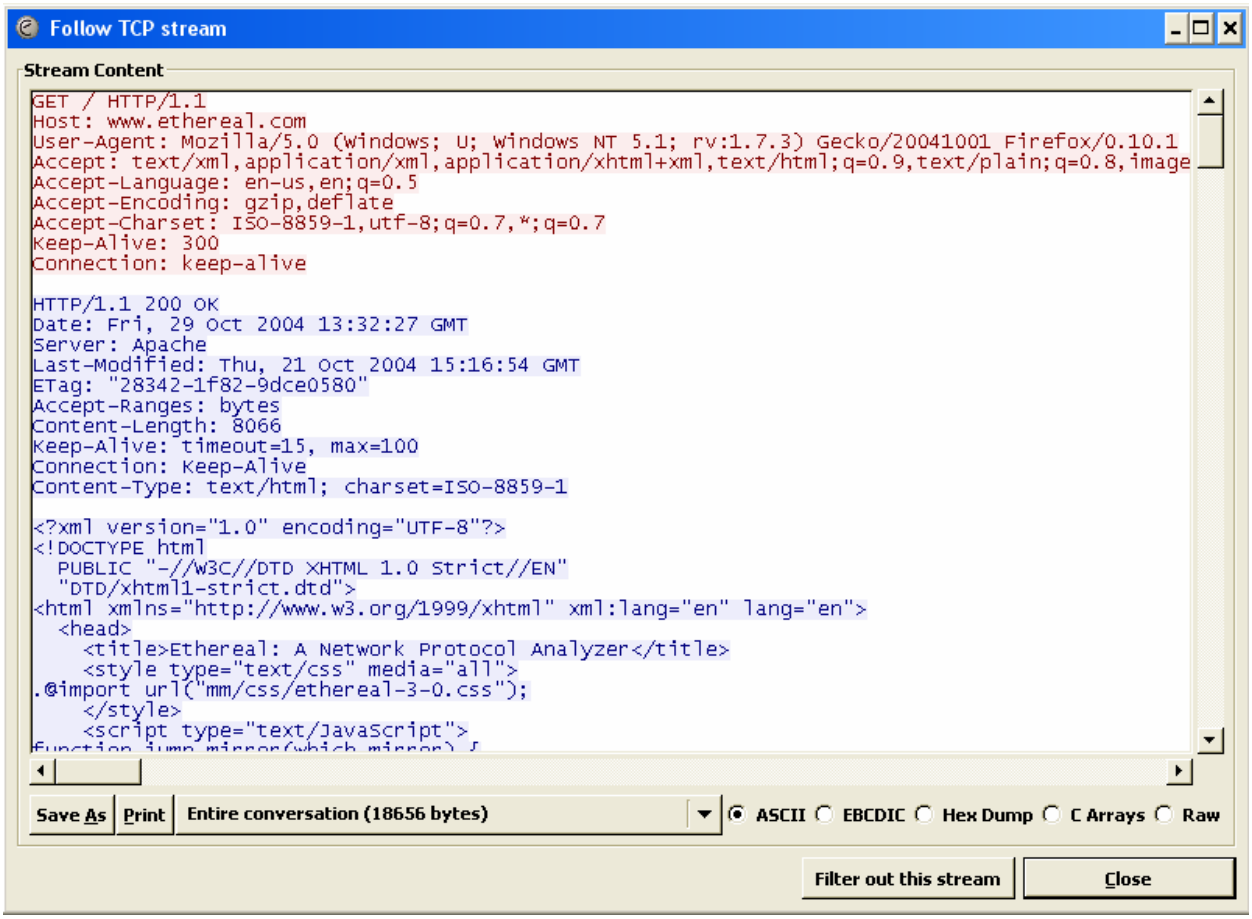


Рис. 1.12. Восстановленный сеанс TCP

Кнопка «Entire conversation» с раскрывающимся списком позволяет отобразить обе стороны, участвующие в обмене, или только одну из них. Диалоговое окно позволяет отобразить данные в различных форматах (ASCII, EBCDIC, Hex Dump, C Arrays, Raw) и сохранить их в файл. При обнаружении в сеансе кадров с каким-либо файлом можно отобразить лишь поток соответствующего направления, выбрать необходимый формат и сохранить его на диск.

ВЫПОЛНИТЬ!

45. Определите, что передавалось в рамках захваченных вами сеансов TCP.

2. ВЫЯВЛЕНИЕ СЕТЕВЫХ АТАК ПУТЕМ АНАЛИЗА ТРАФИКА

2.1. Этапы сетевой атаки

Стандартные сетевые атаки производятся в три этапа: сбор информации, выявление уязвимых мест атакуемой системы и реализация выбранной атаки.

Этап сбора информации заключается в изучении сетевой топологии атакуемой сети, определении типа и версии операционной системы атакуемого узла, выявлении доступных сетевых и иных сервисов, функционирующих на атакуемом узле.

Этап выявления уязвимых мест атакуемой системы осуществляется после или параллельно с этапом сбора информации. Его суть заключается в выяснении версий используемого на атакуемом узле сетевого ПО, выявлении его конфигурации и в анализе наличия уязвимостей в указанном ПО и его настройках.

И, наконец, этап реализации атаки — это либо отправка определенных последовательностей сетевых пакетов на определенные сетевые службы, приводящая к неработоспособности узла, либо выполнение каких-либо запросов к сетевым службам удаленного узла, результатом которых будет получение доступа к защищаемой информации.

В целом первые два этапа не могут быть классифицированы как преступление. Как указывается в [1], компьютерными сетевыми преступлениями являются предусмотренные уголовным законодательством общественно опасные деяния, совершенные на основе удаленного доступа к объекту посягательства с использованием глобальных компьютерных сетей в качестве основного средства достижения цели. Таким образом, компьютерным преступлением является лишь третий этап — реализация атаки.

Вместе с тем выполнение третьего этапа практически невозможно без проведения двух первых этапов атаки. Следовательно, защите должны подлежать и информация о сетевой топологии, и перечень доступных сервисов, и версии программного обеспечения, и т. п.

2.2. Исследование сетевой топологии

Задача изучения сетевой топологии заключается в выявлении сетевых узлов, присутствующих в заданном диапазоне адресов. При этом распространенные сетевые сканеры, такие как nmap, netcat, InterNetView, решают данную задачу чаще всего путем ICMP-сканирования.

Как известно, протокол ICMP используется для определения доступности сетевых узлов. Стандартной программой, применяющей протокол ICMP, является утилита ping. Функционирование утилиты ping сводится к отправке на тестируемый узел запроса и получению ответа. Отправляемый запрос на-

зывается ICMP-запросом (тип пакета ECHO_REQUEST), а получаемый ответ — ICMP-ответом (тип пакета ECHO_REPLY).

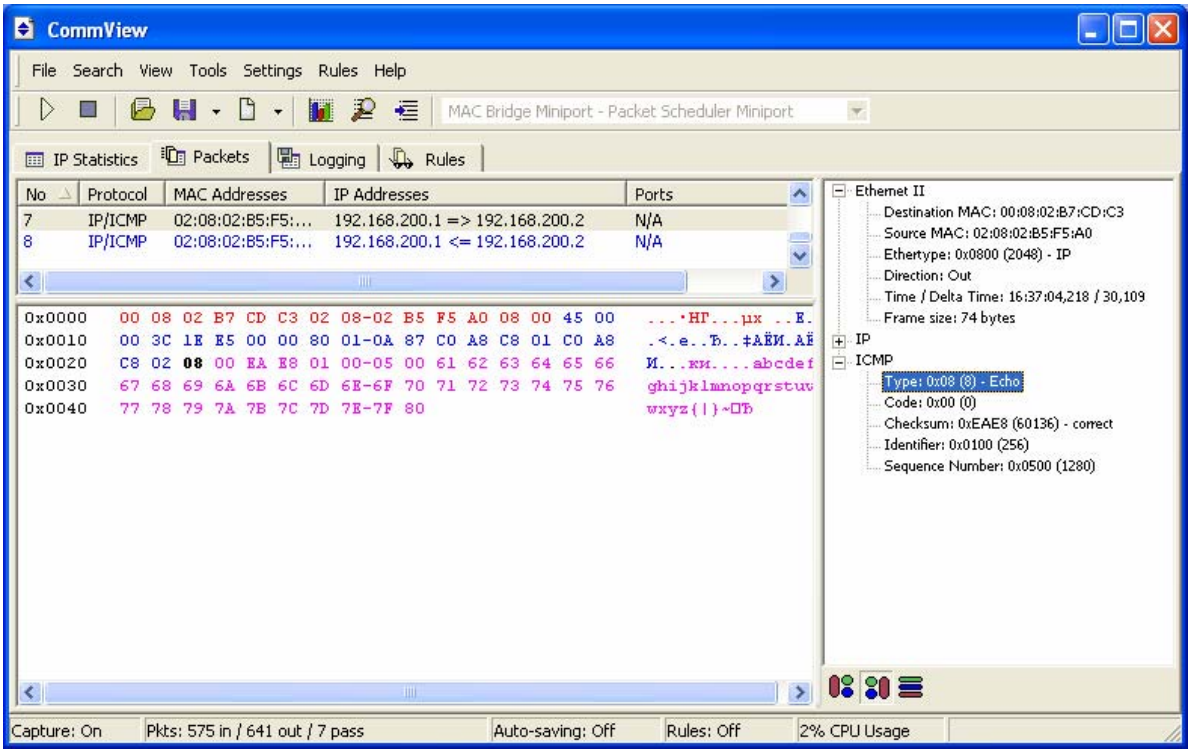


Рис. 2.1. Исходящий ICMP-запрос

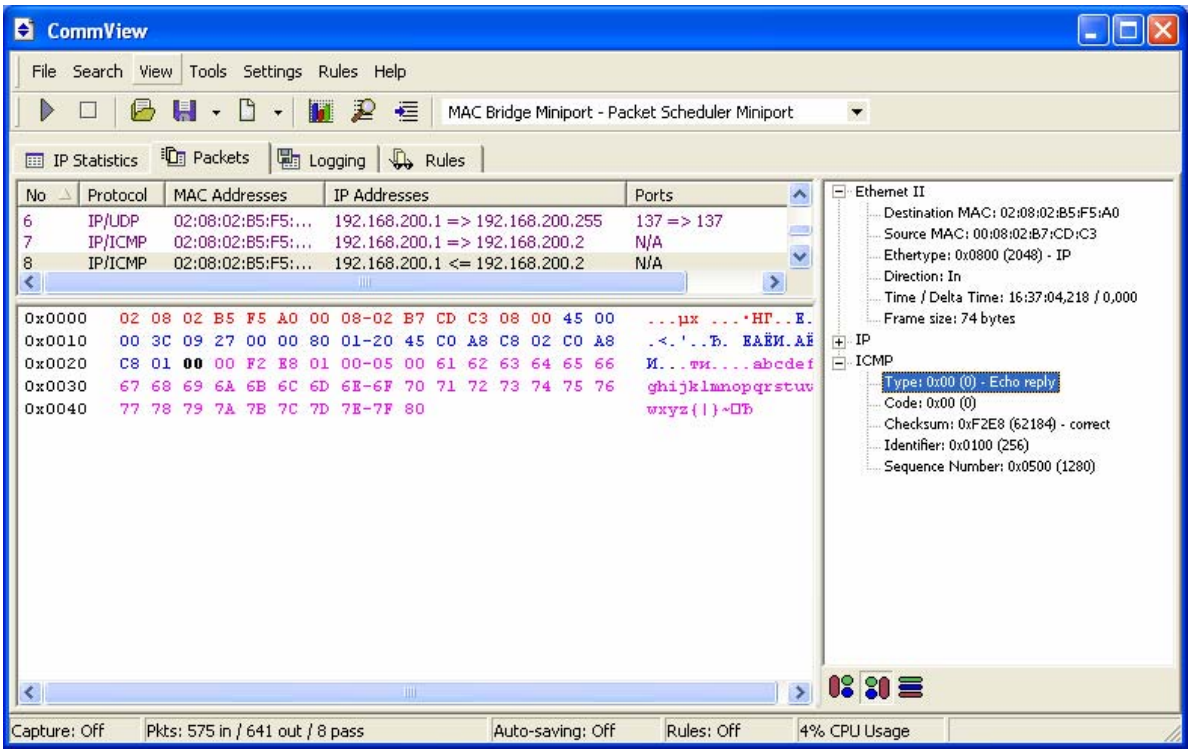


Рис. 2.2. Входящий ICMP-ответ

Так, если на компьютере под управлением, например, ОС Windows 2000 с IP-адресом 192.168.200.1 производится выполнение команды `ping 192.168.200.2` с целью тестирования доступности узла с IP-адресом 192.168.200.1, то в сети можно наблюдать четыре последовательно передаваемые пары сообщений: исходящий ICMP-запрос (тип ICMP-пакета — 0x08, Echo, рис. 2.1) и входящий ICMP-ответ (тип ICMP-пакета — 0x00, Echo-Reply, рис. 2.2).

Аналогичные пакеты имеют место при сканировании, которое производится, например, сканером InterNetView (рис. 2.3). Единственным отличием является то, что вместо четырех последовательных пар пакетов в трафике присутствует только одна пара: ICMP-запрос и ICMP-ответ.

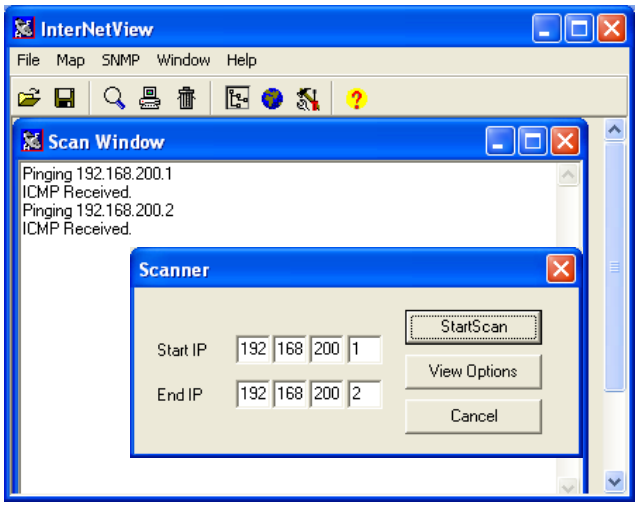


Рис. 2.3. Окно сканера InterNetView

В общем случае единичный входящий ICMP-запрос не является атакой — это лишь стандартное средство проверки доступности узла. Последовательное выполнение ICMP-запросов с перебором адресов из определенного диапазона уже можно рассматривать как атаку. Вместе с тем, если защищаемая сеть является «клиентской сетью», т. е. не содержит серверов, предоставляющих сетевые услуги, то входящие в эту сеть ICMP-запросы также должны рассматриваться как атака.

Для усложнения процесса выявления атаки перебор адресов может вестись не последовательно, а в псевдослучайном порядке.

Как известно, при выполнении стандартного ICMP-запроса в ОС Windows 2000 происходит обмен стандартной текстовой строкой длиной 32 байта. Обычно данная строка представляет собой 26 букв английского алфавита, дополненных шестью дополнительными символами. Вместе с тем объем передаваемых данных может быть существенно увеличен (до 65 535 байт) с целью передачи не стандартной последовательности, а некоторой специально подготовленной команды или текста. В этом случае проявлением атаки могут быть исходящие ICMP-ответы, в которых может быть передана защищаемая информация.

Кроме того, по получаемому ICMP-ответу, а именно по коду ICMP-пакета, злоумышленник может определить тип операционной системы тестируемого узла с целью конкретизации дальнейшей атаки.

Таким образом, в случае ICMP-пакетов атакой будем считать входящие ICMP-запросы и исходящие ICMP-ответы.

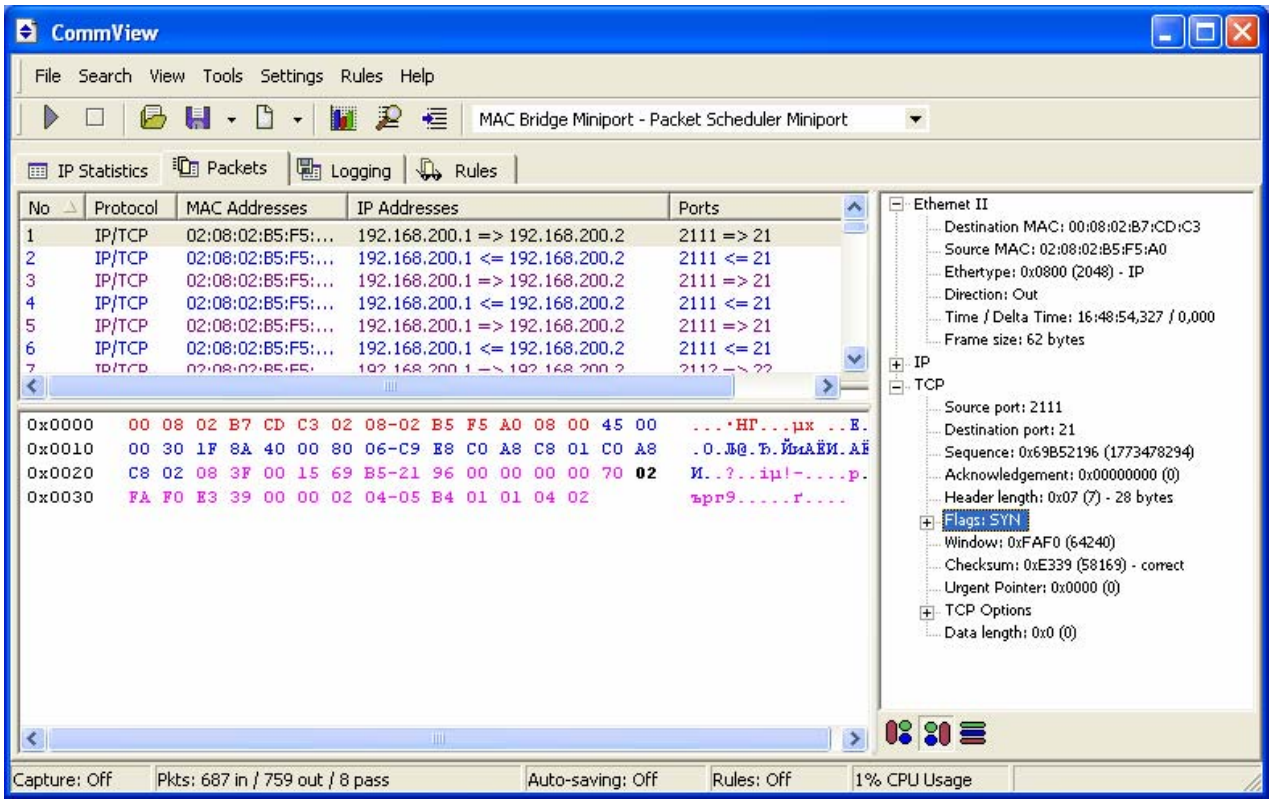


Рис. 2.4. Запрос установки TCP-соединения

ВЫПОЛНИТЬ!

1. Определите настройки протокола TCP/IP вашего компьютера, например, командой `ipconfig` из командной строки.
2. Установите и запустите средство анализа сетевого трафика. Осуществите захват сетевого трафика.
3. Выполните команду `ping *.*.*.*` для обнаружения в сети соседнего компьютера.
4. Осуществите просмотр трафика. В полученных сетевых пакетах убедитесь в наличии полей «источник», «приемник», «тип протокола».
5. Найдите пакет, источником которого является Ваш компьютер, тип протокола — ICMP, описание — Echo. Откройте подробное описание данного пакета. Найдите тип пакета и отправляемые данные. Сколько и каких символов отправляется на искомый компьютер?

6. Найдите ответный пакет (приемник — ваш компьютер, тип протокола — ICMP, описание — Echo Reply). Откройте подробное описание данного пакета. Сколько и каких символов отправляется в ответ?
7. Сколько раз осуществляется обмен ICMP-пакетами? Как представлены в IP-пакетах IP-адреса приемника и источника?

Вторым широко распространенным способом выявления сетевой топологии является TCP-сканирование, которое заключается в последовательной попытке установления сетевого соединения по определенному порту с перебором IP-адресов.

При установке TCP-соединения, как было указано ранее, первым пакетом, отправляемым на тестируемый узел, является пакет с установленным флагом SYN (рис. 2.4). В зависимости от того, присутствует ли в сети компьютер с указанным адресом, на котором включена тестируемая служба, возможны три ситуации. В том случае, если компьютер присутствует и на нем функционирует запрашиваемый порт, ответом будет пакет с установленными флагами ACK и SYN, указывающими на то, что по данному порту может быть установлено соединение (рис. 2.5). Анализируя данный ответ, атакующий не только может установить факт присутствия в сети узла, но и определить наличие на нем определенной сетевой службы.

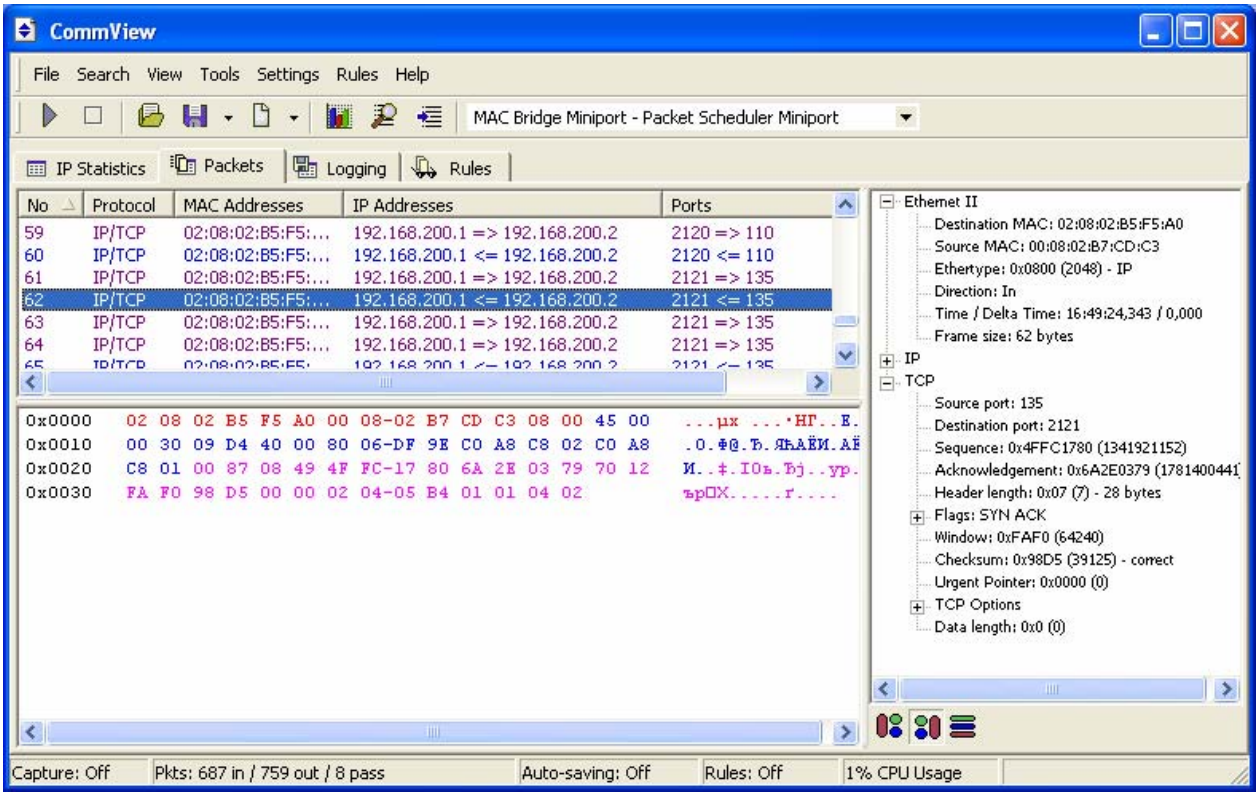


Рис. 2.5. Ответ о возможности установки TCP-соединения

В том случае, если компьютер присутствует, но запрашиваемый порт на нем не открыт, в ответ отправляется TCP-пакет с установленными флагами ACK и RST, указывающими на то, что по запрашиваемому порту соединение

установить нельзя (рис. 2.6). Получив подобный ответ, атакующий принимает решение о присутствии в сети узла с интересующим IP-адресом, но недоступности запрашиваемого порта.

И, наконец, если в сети нет искомого узла, то в ответ не будет получено ничего.

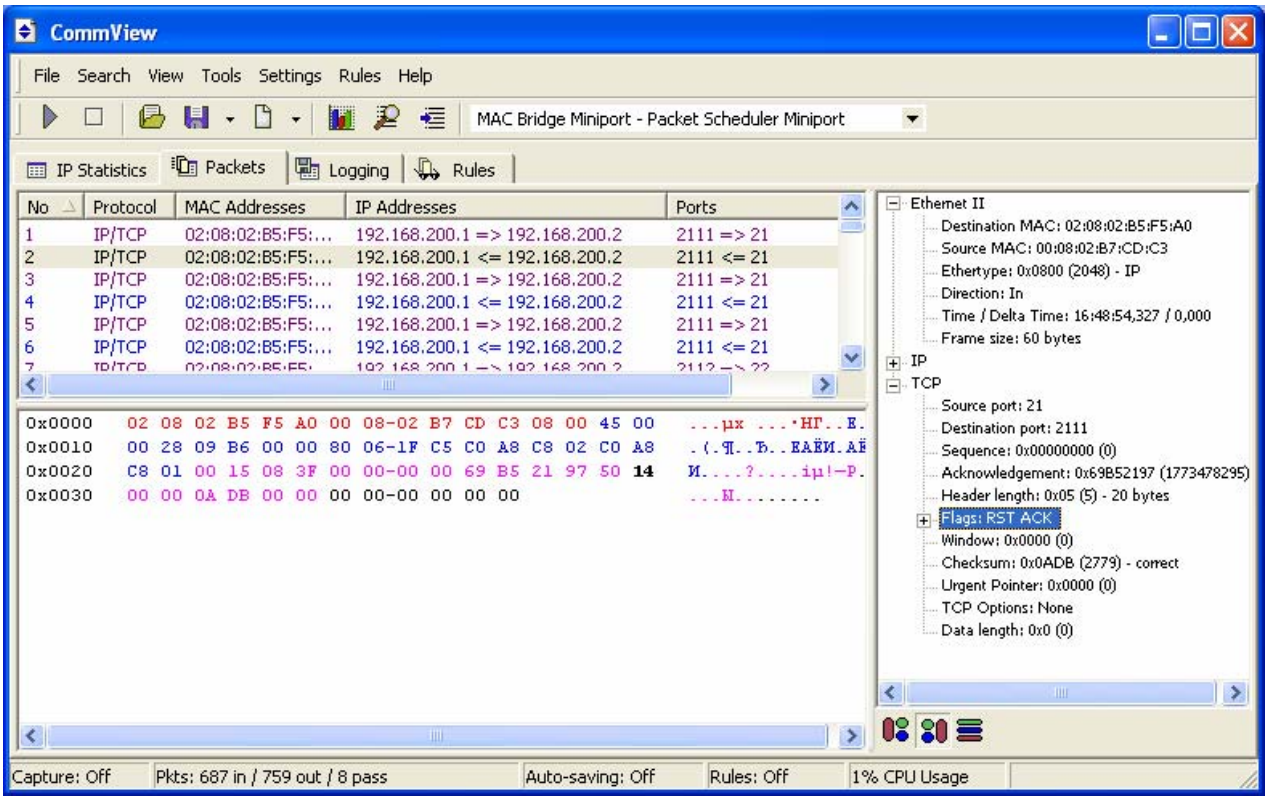


Рис. 2.6. Ответ о невозможности установки TCP-соединения

2.3. Обнаружение доступных сетевых служб

Выше была описана технология выявления сетевых узлов путем установки TCP-соединения. Аналогичная технология используется, когда заранее известно, что узел в сети присутствует, но необходимо получить информацию о доступных сетевых службах, т. е. выполнить сканирование портов сетевого узла. В этом случае последовательно осуществляются попытки подключения к сетевым портам в определенном диапазоне.

Чаще всего применяются не подряд все номера портов, а только те из них, которые наиболее интересны злоумышленникам с целью дальнейшего проникновения. В ряде случаев перечень номеров портов может быть сформирован злоумышленниками на основе полученной ранее по коду ICMP-ответа информации о типе операционной системы.

Одиночный запрос установки TCP-соединения по одному из портов может считаться атакой лишь в случае, когда защищаемая сеть является «клиентской». Однако если мы защищаем «клиентскую» сеть, которая не должна

предоставлять вовне каких-либо сетевых услуг, то и одиночные попытки установки соединения должны интерпретироваться как атака. Последовательные же попытки установить соединение с несколькими портами явно свидетельствует о начавшейся сетевой атаке.

Таким образом, в случае TCP-пакетов атакой будем считать все попытки установки TCP-соединения, инициируемые извне.

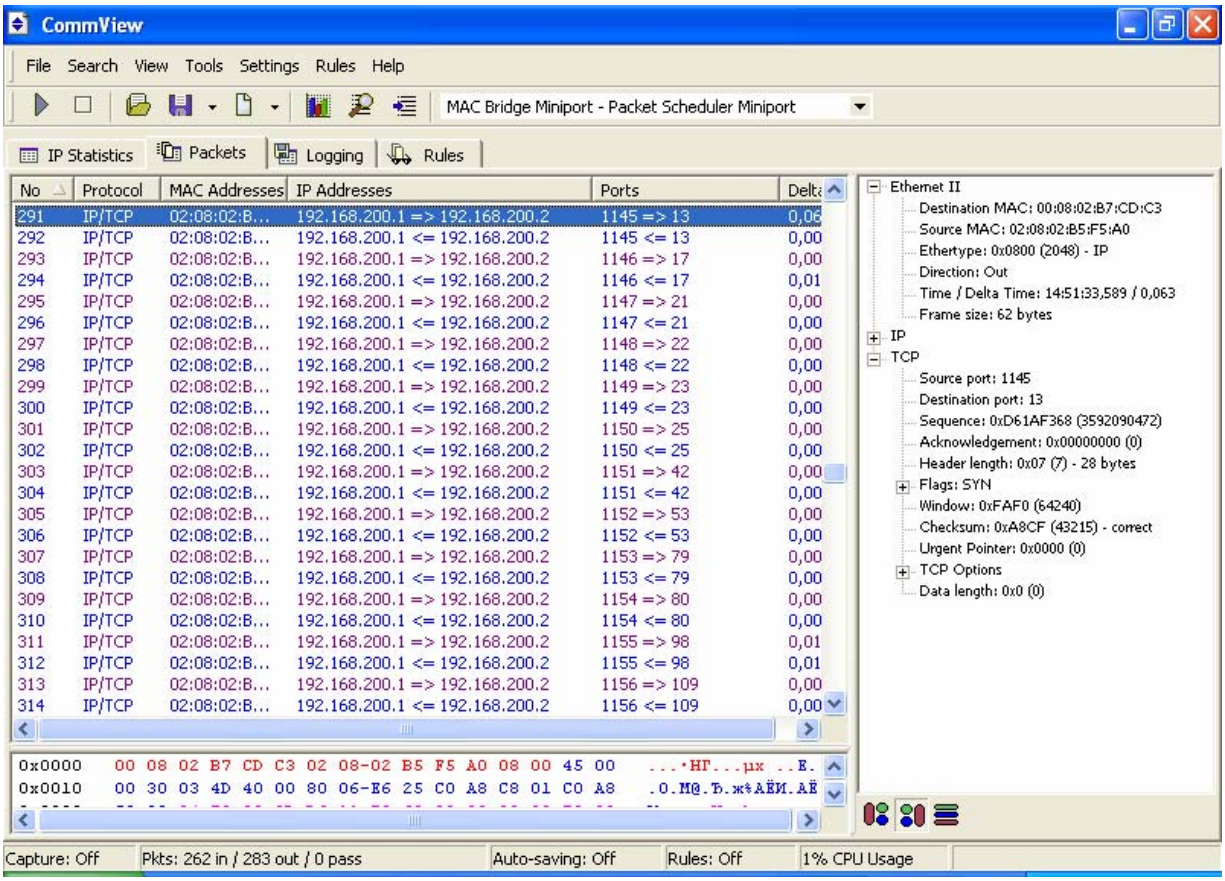


Рис. 2.7. Последовательные попытки установки TCP-соединений

Заметим, что мы рассмотрели только классический способ TCP-сканирования, известный как сканирование методом Connect(), когда устанавливается полное TCP-соединение. Вместе с тем известны более изощренные методы, позволяющие процесс сканирования осуществлять скрытно: SYN-сканирование, FIN-сканирование, ACK-сканирование, XMAS-сканирование, NULL-сканирование. Соответственно известны утилиты, позволяющие автоматизировать указанные методы. Коротко опишем эти методы.

Метод сканирования TCP-портов системным вызовом Connect() является основным для сканирования портов по протоколу TCP. Функция Connect() позволяет атакующему узлу соединиться с любым портом сервера. Если порт, указанный в качестве параметра функции, прослушивается сервером (т. е. порт открыт для соединения), то результатом выполнения функции будет установление соединения с сервером по указанному порту. В противном случае, если соединение не установлено, то порт с указанным номером является за-

крытым. Метод Connect() является легко обнаруживаемым благодаря наличию многочисленных попыток подключения с одного адреса и ошибок установления соединения (поскольку атакующий узел после соединения с сервером сразу обрывает его).

Метод сканирования TCP-портов флагом SYN известен еще как «сканирование с установлением наполовину открытого соединения» поскольку установление полного TCP-соединения не производится. Вместо этого атакующий отправляет на определенный порт сервера SYN-пакет, как бы намереваясь создать соединение, и ожидает ответ. Наличие в ответе флагов SYN|ACK означает, что порт открыт и прослушивается сервером. Получение в ответ TCP-пакета с флагом RST означает, что порт закрыт и не прослушивается. В случае приема SYN|ACK-пакета узел немедленно отправляет RST-пакет для сброса устанавливаемого сервером соединения.

Метод сканирования TCP-портов флагом FIN известен по-другому как «обратное стелс-сканирование с использованием флага FIN». Идея метода заключается в том, что согласно RFC 793 на прибывший FIN-пакет на закрытый порт сервер должен ответить RST-пакетом. FIN-пакеты на открытые порты игнорируются объектом сканирования.

Метод сканирования TCP-портов флагом ACK похож на FIN-сканирование, известен по-другому как «обратное стелс-сканирование с использованием флага ACK». Идея метода заключается в том, что согласно RFC 793 на прибывший ACK-пакет на закрытый порт сервер должен ответить RST-пакетом. ACK-пакеты на открытые порты игнорируются объектом сканирования.

Методы сканирования XMAS («Новогодняя елка») и NULL заключаются в отправке на сервер TCP-пакета с установленными всеми флагами (XMAS) либо со всеми сброшенными флагами (NULL). В соответствии с RFC 793 на прибывший пакет с данными значениями флагов на закрытый порт сервер должен ответить RST-пакетом. Такие пакеты на открытые порты игнорируются объектом сканирования.

Указанные выше методы сканирования позволяют злоумышленнику выяснить наличие открытых TCP-портов на атакуемом узле. Для обнаружения открытых UDP-портов применяется иной подход.

Выполнить анализ открытых UDP-портов злоумышленнику несколько сложнее, чем TCP-портов. Причина в том, что в отличие от протокола TCP, UDP является протоколом с негарантированной доставкой данных. Поэтому UDP-порт не посылает подтверждение приема запроса на установление соединения, и нет никакой гарантии, что отправленные UDP-порту данные успешно дойдут до него. Тем не менее большинство серверов в ответ на пакет, прибывший на закрытый UDP-порт, отправляют ICMP-сообщение «Порт недоступен» (Port Unreachable — PU). Таким образом, если в ответ на UDP-пакет пришло ICMP-сообщение PU, то сканируемый порт является закрытым, в противном случае (при отсутствии PU) порт открыт.

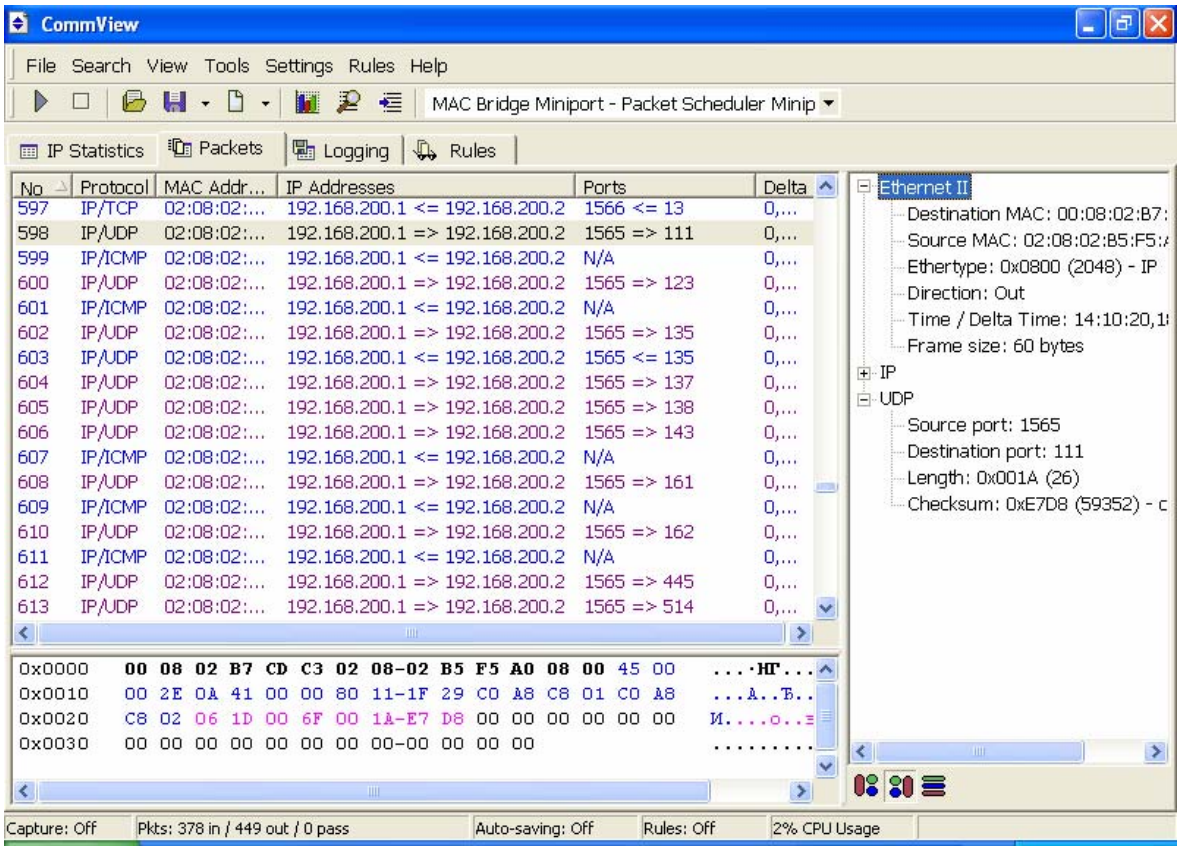


Рис. 2.8. Пример UDP-сканирования

Обычно сканеры работают следующим образом: на тестируемый порт отправляется UDP-пакет, состоящий, например, из 18 нулей (рис. 2.8). Если соответствующий порт открыт, то в ответ тестируемый компьютер может либо ничего не отправить, либо отправить ответный UDP-пакет. В этом случае сканер делает вывод о том, что порт открыт. Если же порт закрыт, то тестируемый компьютер должен ответить ICMP-сообщением с кодом 0x03 (Port Unreachable). Получив такое сообщение (рис. 2.9), сканер сделает вывод о закрытости данной службы.

Для UDP-протокола нет четкого понятия, кто является инициатором пакета. Так, исходящий UDP-запрос на любую службу, очевидно, предполагает входящий UDP-ответ. Вместе с тем любой UDP-пакет с равной вероятностью может быть и входящим UDP-ответом, и исходящим UDP-запросом. Следовательно, в отличие от протокола TCP здесь нельзя четко разделить входящие и исходящие UDP-пакеты.

Единственным критерием, который позволяет идентифицировать входящие UDP-пакеты как атаку, является последовательность пакетов, отправляемая на различные UDP-порты. При этом порт отправителя вероятнее всего будет в «клиентском» диапазоне — больше 1024.

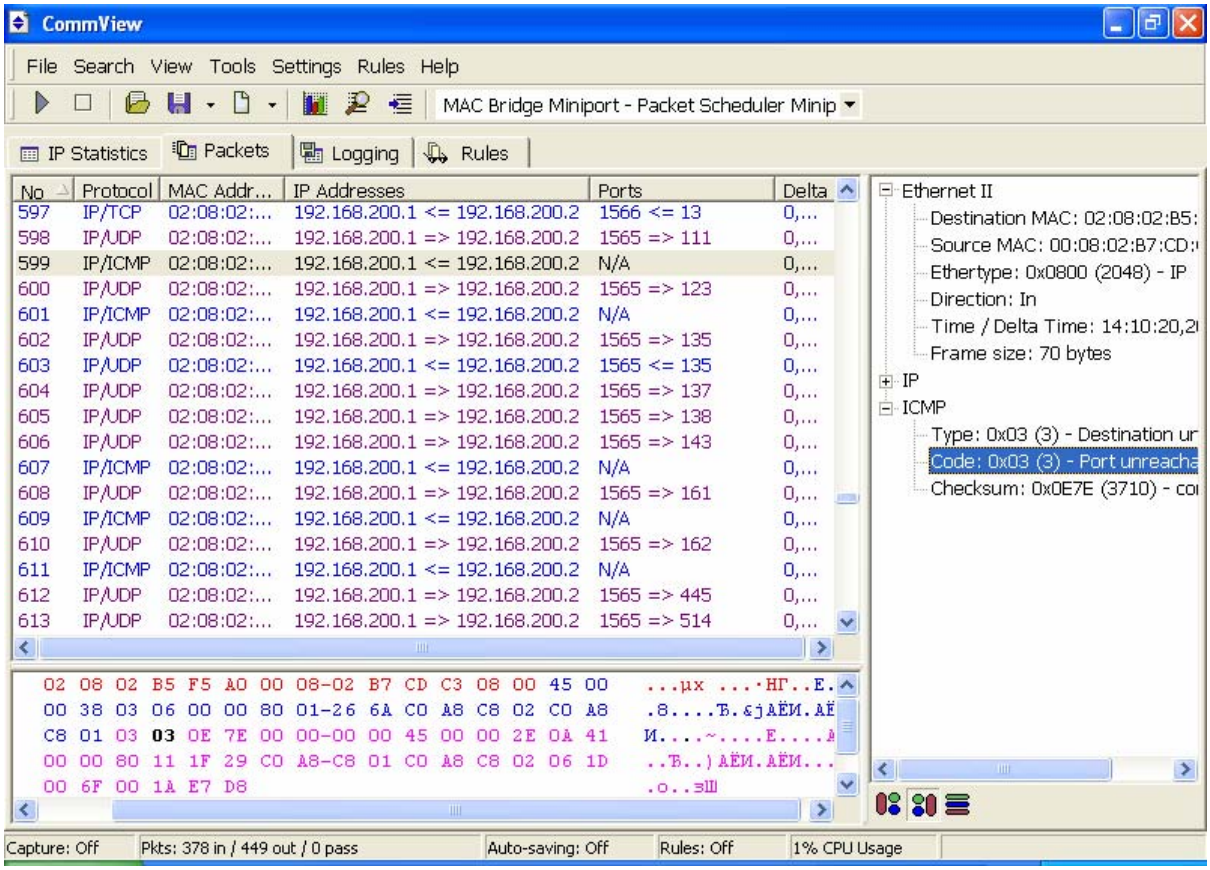


Рис. 2.9. Пример ICMP-пакета Port Unreachable

ВЫПОЛНИТЬ!

8. Запустите программу-сканер портов SAURON.
9. Осуществите захват сетевого трафика.
10. Выполните сканирование TCP- и UDP- портов соседнего компьютера в диапазоне от 130 до 140, последовательно выполняя сканирование различными методами: Connect(), SYN-сканированием, FIN-сканированием, ACK-сканированием, XMAS-сканированием, NULL-сканированием, UDP-сканированием.
11. Проанализируйте полученный трафик. Найдите отличительные признаки каждого метода сканирования. Сформулируйте признаки подобных атак. Сделайте вывод о возможности блокирования данных атак.

2.4. Выявление уязвимых мест атакуемой системы

Данный этап атаки производится чаще всего одновременно с выяснением открытых портов. Суть его заключается в определении типа и версии программного продукта, отвечающего за получение информации на открытом порту. Это может быть, например, операционная система в целом, web- , ftp- или иной сервер. Зная версию программного продукта, злоумышленник мо-

жет, воспользовавшись известными уязвимостями данной версии, осуществить целенаправленную атаку.

Так, например, выяснив наличие открытого порта 25, злоумышленник отправляет стандартный запрос на соединение с ним и в ответ получает версию программного продукта, реализующего SMTP-сервер (рис. 2.10).

Признаком атаки в данном случае является выполнение входящих запросов к внутренним сетевым службам, особенно к таким, которые редко используются для работы в Интернет.

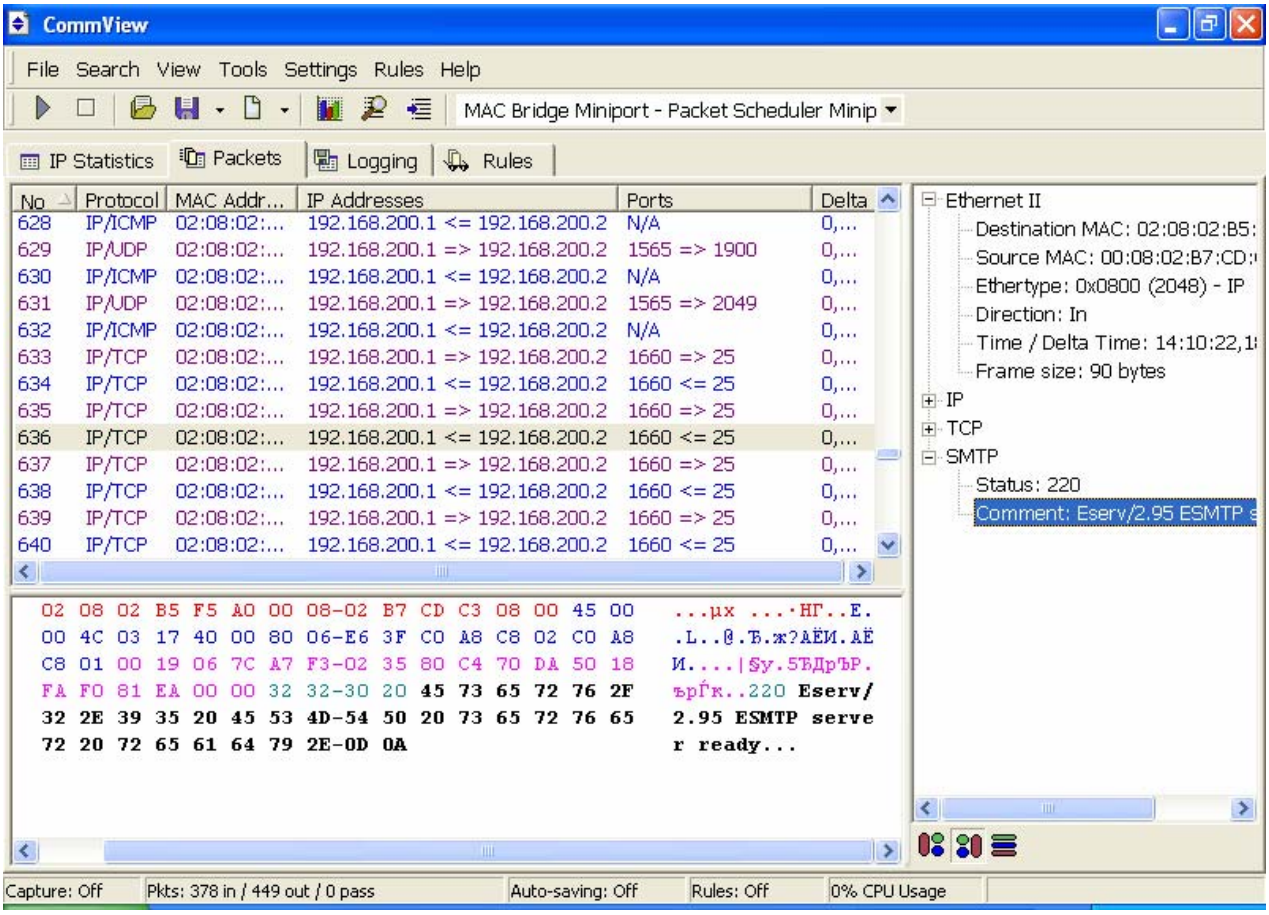


Рис. 2.10. Пример ответа SMTP-сервера

2.5. Реализации атак

В литературе содержится большое количество упоминаний реализаций атак на различные сетевые службы: «Ping Flood» (затопление ICMP-пакетами), «Ping of Death» (превышение максимально возможного размера IP-пакета), «SYN Flood» (затопление SYN-пакетами), «Teardrop», «UDP Bomb» и т. д. Так как целью пособия не является изучение всех алгоритмов атак, ограничимся только двумя атаками типа «отказ в обслуживании» (DoS), приводящими к «зависанию» либо сетевой службы, либо компьютера в целом. За-

метим, что описываемые атаки были актуальны для операционных систем предыдущего поколения.

Атака «Ping of Death» приводила к зависанию реализации стека протоколов TCP/IP в ОС Windows 95. Атака основана на отправке IP-пакета, длина которого превышает стандартную величину. Напомним, что максимальный размер IP-пакета составляет 65535 байт, из них 20 байт отводится на заголовок. Таким образом, максимальный размер данных, передаваемых в одном пакете, составляет 65515 байт. В реализации ряда ОС именно такой буфер и отводился для хранения получаемых данных, а размер буфера не контролировался. Если же приходил пакет большей длины, то получаемые данные затирали машинный код в оперативной памяти, находившийся после отведенного буфера. Для реализации атаки достаточно было использовать стандартную утилиту ping следующим образом:

```
ping -l 65527 -s 1 адрес_жертвы
```

В этом случае отправляемые данные составляют 65527 байт, заголовок ICMP — 8 байт, заголовок IP — 20 байт. Таким образом, отправлялся пакет длиной 65555 байт, что на 20 байт превышало максимальный размер IP-пакета.

Другая известная атака, достаточно долго приводившая в неработоспособное состояние сетевые узлы под управлением ОС Windows NT версии 4.0, получила название WinNuke. Для ее реализации в Интернет можно было найти программу с аналогичным названием и простым интерфейсом (рис. 2.11).

В процессе выполнения программа WinNuke устанавливает стандартное TCP-соединение с портом 139 атакуемого узла. Особенностью соединения является то, что атакующим для установки соединения используется TCP-порт с номером 40, в отличие от обычно используемых номеров портов больших, чем 1024. После установки соединения атакующий отправляет нестандартный для SMB-протокола пакет (рис. 2.12), в результате получения которого ОС Windows NT версии 4.0 «вылетала в синий экран смерти».

2.6. Выявление атаки на протокол SMB

Протокол SMB/CIFS (Server Message Blocks/Common Internet File System) предназначен для соединения компьютеров с ОС типа Windows 9* и Windows NT 5.* между собой или с сервером Samba (UNIX-сервером). Протокол SMB включает в себя все возможные операции (команды) для работы с файлами и принтерами (открытие, закрытие, создание и удаление файлов и директорий, чтение и запись в файл, поиск файлов, отправка на печать и отмена печати).

Рис. 2.11. Интерфейс программы WinNuke

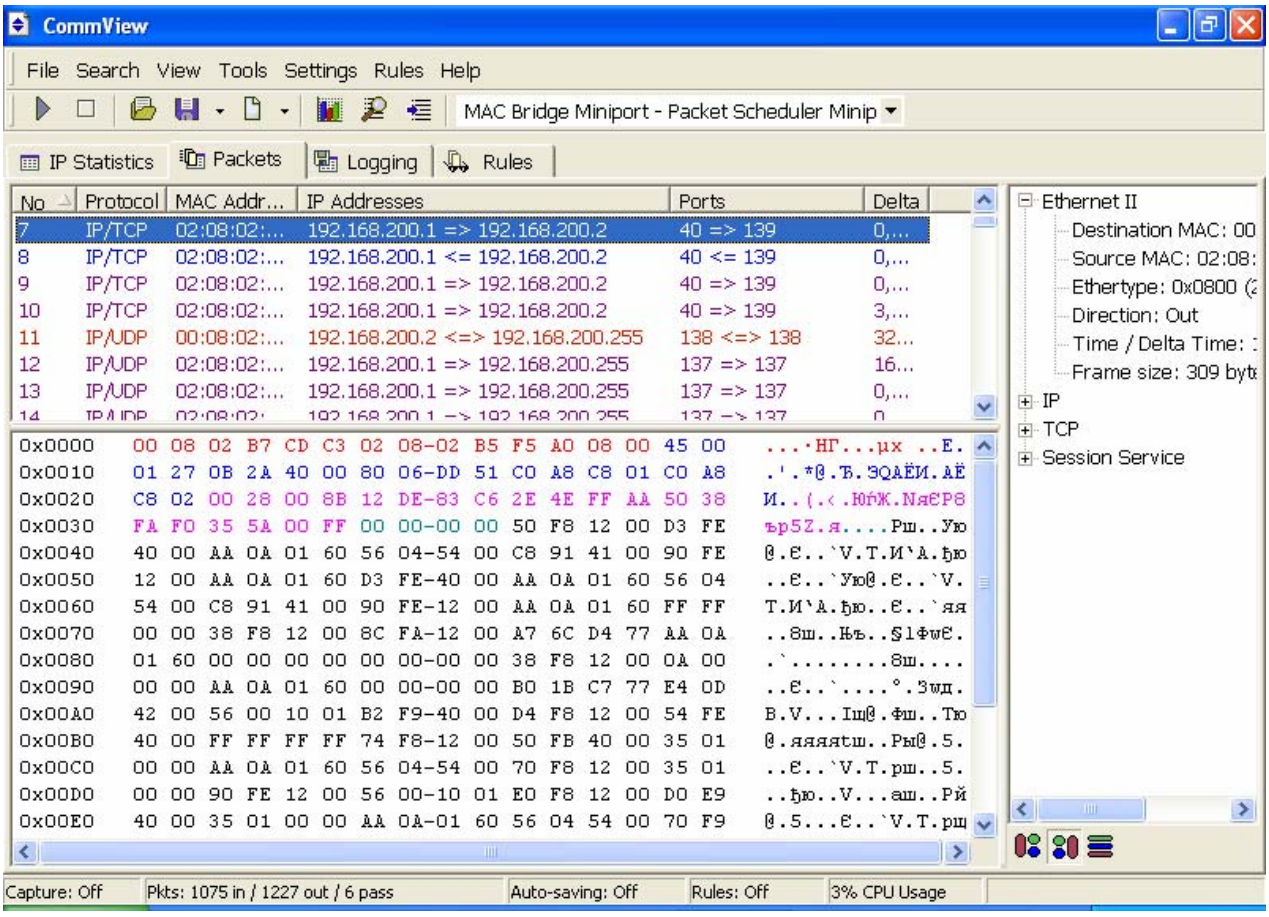


Рис. 2.12. Пример сетевого пакета программы WinNuke

SMB-сообщение состоит из двух частей: заголовка фиксированного размера и поля команды, размер которой меняется динамически в зависимости от состава сообщения. Протокол SMB имеет несколько версий и диалектов, каждая из последующих совместима с предыдущей (табл. 2.1).

Таблица 2.1

Диалекты протокола SMB

Название протокола	Обозначение
Core	PC NETWORK PROGRAM 1.0
Core Plus	MICROSOFT NETWORKS 1.03
LAN Manager 1.0	LANMAN1.0
LAN Manager 2.0	LM1.2X002
LAN Manager 2.1	LANMAN2.1
NT LAN Manager 1.0	NT LM 0.12
Samba's NT LM 0.12	Samba
Common Internet File System	CIFS 1.0

Среди особенностей каждой из версий следует отметить применяемый алгоритм аутентификации, в частности применение открытых или зашифрованных паролей.

Установка соединения между сервером (компьютером, предоставляющим доступ к ресурсу) и клиентом (компьютером, который желает воспользоваться данным ресурсом) происходит в 4 шага:

1. Установка виртуального соединения. Создается двунаправленный виртуальный канал между клиентом и сервером.

2. Выбор версии протокола (рис. 2.13). Клиент посылает запрос, содержащий список всех версий протоколов SMB, по которым он может создать соединение. Сервер отвечает номером записи в списке, предложенном клиентом (считая записи с 0), или значением 0xFF, если ни один из вариантов предложенных протоколов не подходит. Здесь же сервер передает требуемый режим безопасности, признак необходимости шифрования пароля и «вызов» (8 случайных байт), используя который клиент зашифрует пароль и передаст его серверу в виде «ответа» на следующем шаге.

3. Установка параметров сессии (рис. 2.14). Клиент посылает имя пользователя, пароль (если он существует) в виде «ответа», имя рабочей группы, а также полный путь к доступной директории на сервере (перечень доступных директорий сервер предоставляет клиенту ранее по иному запросу).

4. Получение доступа к ресурсу. Сервер выдает клиенту идентификатор (TID — уникальный идентификатор для ресурса, используемого клиентом), показывая тем самым, что пользователь прошел процедуру авторизации и ресурс готов к использованию. Сервер указывает тип службы доступа: A — для диска или файла; LPT1 — для вывода на печать; COM — для прямого соединения принтеров и модемов, IPC — для идентификации при доступе к ресурсу.

Рис. 2.13. Согласование SMB-диалекта

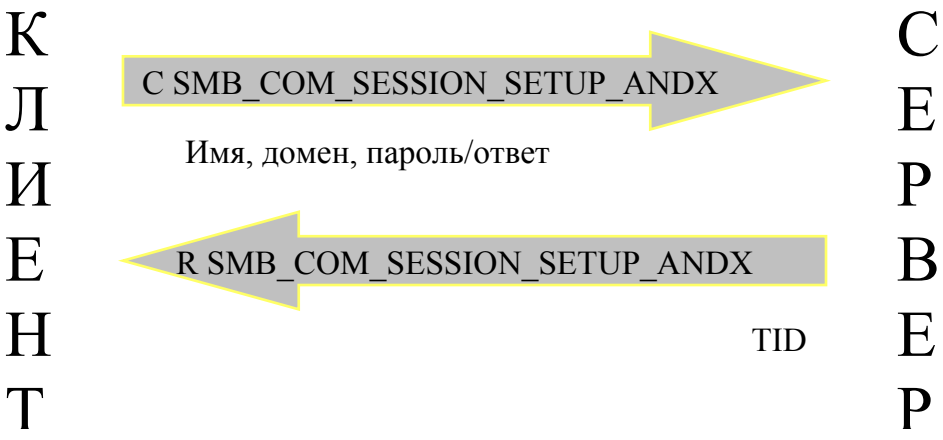


Рис. 2.14. Установка параметров сессии

ВЫПОЛНИТЬ!

12. Создайте на виртуальном компьютере, работающем под управлением ОС Windows 2000, каталоги и в нем небольшой текстовый файл. Предоставьте созданный каталог в сетевой доступ.
13. Включите захват трафика. В основной ОС откройте в сетевом окружении на виртуальном компьютере созданный текстовый файл. Для чего зарегистрируйтесь, введя имя и пароль пользователя, имеющегося на виртуальной ОС.
14. Просмотрите полученный трафик. Какой протокол используется для файлового обмена? Какие номера TCP-портов задействованы на приемнике и источнике?
15. Найдите пакеты этапа согласования SMB-диалекта. Какой диалект выбран сервером?
16. Найдите пакет, в котором передается имя пользователя. Передается ли пароль пользователя в открытом виде?

17. Найдите пакет, в котором передается текст открытого вами файла (описание пакета начинается с «R read & X»). Передается ли текст файла в зашифрованном виде?

Одной из основных широко известных уязвимостей [4], присутствующих в сетевых настройках по умолчанию в ОС Windows 2000 (исправлено в Windows XP), является возможность установления «нулевого сеанса» (анонимного подключения). Целью данной атаки являлось подключение к сетевым ресурсам ОС Windows 2000 без указания имени пользователя и пароля.

Для установления «нулевого сеанса» осуществляется подключение к ресурсу IPC$ удаленного компьютера без указания имени и пароля при помощи стандартной утилиты net:

```
net use \\*.*.*.*\IPC$ "" /user:"",
```

где `*.*.*.*` — IP-адрес компьютера, **`/user:""`** — имя пользователя (пусто); **`""`** — пароль пользователя (пусто).

ВЫПОЛНИТЬ!

18. Разорвите все установленные ранее соединения командой:

```
net use * /delete
```

19. Включите режим захвата трафика. Выполните анонимное подключение к виртуальному компьютеру. Просмотрите полученный трафик. Какие номера TCP-портов задействованы на приемнике и источнике? Какой SMB-диалект выбран сервером? Найдите пакет, в котором передается имя пользователя в виде пустой строки.
20. Сделайте вывод о возможности обнаружения и блокирования данной атаки.

3. ЗАЩИТА КОМПЬЮТЕРНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ МЕЖСЕТЕВЫХ ЭКРАНОВ

3.1. Понятие межсетевого экрана

В стратегии защиты от несанкционированного доступа к информационным ресурсам компьютерной сети особое внимание уделяется обеспечению безопасности ее границ. Целостность периметра компьютерной сети обеспечивается использованием тех или иных базовых технологий межсетевого экранирования в точке подключения защищаемой сети к внешней неконтролируемой сети. В качестве внешней сети чаще всего выступает глобальная сеть Интернет. Систему разграничения компьютерных сетей с различными политиками безопасности, реализующую правила информационного обмена между ними, называют межсетевым экраном (МЭ). В переводной литературе также встречаются термины firewall или брандмауэр.

Межсетевой экран — это локальное (однокомпонентное) или функционально-распределенное (многокомпонентное) программное (программно-аппаратное) средство (комплекс), реализующее контроль за информацией, поступающей в автоматизированную систему (АС) и/или исходящей из нее (рис. 3.1).

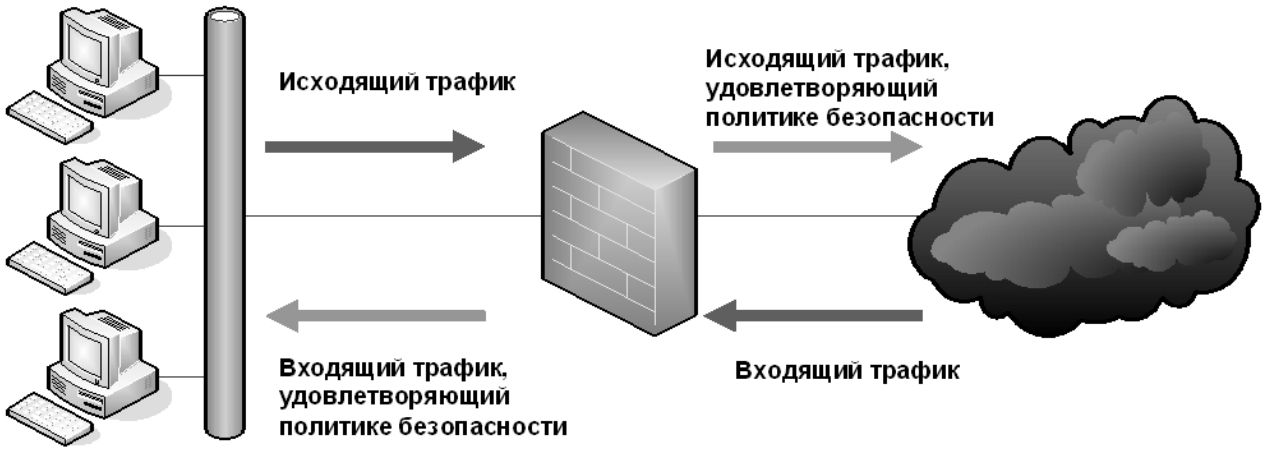


Рис. 3.1. Контроль периметра сети МЭ (защищаемая сеть слева)

МЭ повышает безопасность объектов внутренней сети за счет игнорирования несанкционированных запросов из внешней среды. Это уменьшает уязвимость внутренних объектов, так как сторонний нарушитель должен преодолеть некоторый защитный барьер, в котором механизмы обеспечения безопасности сконфигурированы особо тщательно. Кроме того, экранирующая система, в отличие от универсальной, может и должна быть устроена более простым и, следовательно, более безопасным образом, на ней должны присутствовать только те компоненты, которые необходимы для выполнения

функций экранирования. Кроме того, экранирование позволяет контролировать информационные потоки, исходящие во внешнюю среду, что способствует поддержанию во внутренней области режима конфиденциальности. Кроме функций разграничения доступа, МЭ может обеспечивать выполнение дополнительных функций безопасности (аутентификацию, контроль целостности, фильтрацию содержимого, обнаружение атак, регистрацию событий).

МЭ не является симметричным устройством, для него определены понятия «внутри» и «снаружи» (входящий и исходящий трафики). При этом задача экранирования формулируется как защита внутренней области от неконтролируемой и потенциально враждебной внешней.

3.2. Компоненты межсетевого экрана

В общем случае алгоритм функционирования МЭ сводится к выполнению двух групп функций, одна из которых ограничивает перемещение данных (фильтрация информационных потоков), а вторая, наоборот, ему способствует (посредничество в межсетевом взаимодействии). Следует отметить, что выполнение МЭ указанных групп функций может осуществляться на разных уровнях модели OSI. Принято считать, что чем выше уровень модели OSI, на котором МЭ обрабатывает пакеты, тем выше обеспечиваемый им уровень защиты.

Как отмечено выше, МЭ может обеспечивать защиту АС за счет фильтрации проходящих через него сетевых пакетов, то есть посредством анализа содержимого пакета по совокупности критериев на основе заданных правил и принятия решения о его дальнейшем распространении в (из) АС. Таким образом, МЭ реализует разграничение доступа субъектов из одной АС к объектам другой АС. Каждое правило запрещает или разрешает передачу информации определенного типа между субъектами и объектами. Как следствие, субъекты одной АС получают доступ только к разрешенным информационным объектам другой АС. Интерпретация набора правил выполняется последовательностью фильтров, которые разрешают или запрещают передачу данных (пакетов) на следующий фильтр. МЭ или один из его компонентов, функционирующий вышеописанным образом, называют пакетным фильтром.

Пакетный фильтр функционирует на сетевом уровне модели OSI (рис. 3.2). Значимой для функционирования пакетного фильтра информацией является:

- IP-адрес отправителя;
- IP-адрес получателя;
- тип протокола (TCP, UDP, ICMP);
- порт отправителя (для TCP, UDP);
- порт получателя (для TCP, UDP);
- тип сообщения (для ICMP); а иногда и другая информация (например, время суток, день недели и т.д.).

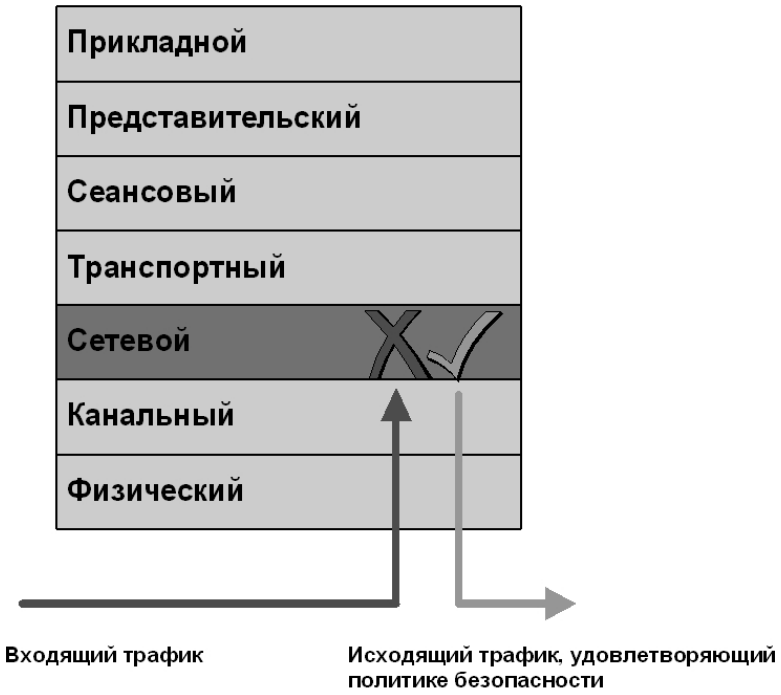


Рис. 3.2. Место пакетного фильтра в модели OSI

В англоязычной литературе рассмотренный компонент МЭ чаще всего обозначают термином «stateless packet filter» или просто «packet filter». Данные системы просты в использовании, дешевы, оказывают минимальное влияние на производительность АС. Основным недостатком является их уязвимость при атаке, называемой IP-спуфинг — фальсификации адресов отправителя сообщений. Кроме того, они сложны при конфигурировании: для их установки требуется знание сетевых, транспортных и прикладных протоколов.

Другой вариант алгоритма функционирования МЭ предполагает, что защита АС обеспечивается с помощью экранирующего агента, который проверяет допустимость полученного запроса субъекта к объекту, при положительном результате этой проверки устанавливает свое соединение с объектом, а затем обеспечивает пересылку информации между субъектом и объектом взаимодействия, осуществляя контроль и/или регистрацию. В то же время в случае «прозрачных» агентов субъекту кажется, что он непосредственно взаимодействует с объектом. Использование экранирующих агентов позволяет обеспечить дополнительную защитную функцию — сокрытие истинного субъекта взаимодействия.

Выделяют два вида экранирующих агентов в зависимости от того, на каком уровне модели OSI они выполняют свои функции (рис. 3.3): экранирующий транспорт и экранирующий шлюз.

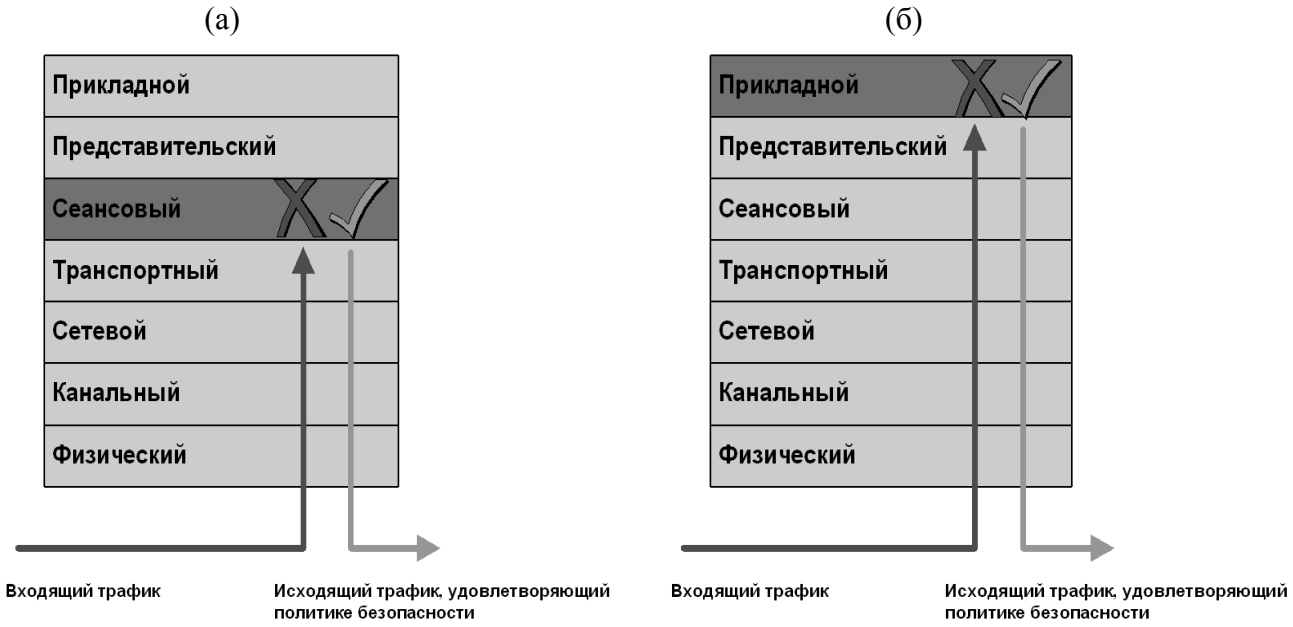


Рис. 3.3. Место экранирующего агента в модели OSI:
(а) — экранирующий транспорт; (б) — экранирующий шлюз

Экранирующий транспорт или шлюз сеансового уровня (в англоязычной литературе используется термин «circuit-level gateway») контролирует допустимость устанавливаемого соединения, участвует в формировании канала передачи данных и не позволяет проходить пакетам, не относящимся к разрешенным сеансам связи. Функционирование данного компонента связано лишь с сессиями протокола TCP. Так как при посредничестве шлюз сеансового уровня анализирует информацию, содержащуюся лишь в заголовках протокола TCP без какого-либо предположения об используемом прикладном протоколе, то существует уязвимость, заключающаяся в том, что в рамках разрешенного установленного соединения приложение может осуществлять передачу произвольных неконтролируемых данных. Как правило, вышеописанный компонент используется лишь в сочетании с другими, а не отдельно.

Более надежную защиту обеспечивает экранирующий шлюз или шлюз прикладного уровня (в англоязычной литературе используется термин «application-level gateway» или «application proxy»), так как он проверяет содержимое каждого проходящего через шлюз пакета на прикладном уровне, где для анализа доступны служебные поля заголовка прикладного протокола и информация пользователя. Прикладной шлюз представляет собой программу-посредник (в англоязычной литературе используется термин «proxy server»), разработанную для конкретного сервиса сети Интернет. Следовательно, при внедрении сервисов, основанных на новых прикладных протоколах, появляется необходимость в разработке новых программ-посредников.

Дальнейшее развитие различных технологий межсетевого экранирования и их взаимопроникновение привело к появлению гибридных компонентов МЭ, сочетающих в себе достоинства всех трех ранее рассмотренных компо-

нентов и лишенных некоторых их недостатков. Такие системы, чаще всего называемые МЭ экспертного уровня (в англоязычной литературе используются термины «stateful inspection firewall» или «deep packet inspection firewall»), функционируют на всех уровнях модели OSI: от сетевого до прикладного включительно (рис. 3.4). Они обладают высокими показателями по производительности функционирования (пакетный фильтр) и по обеспечиваемому уровню безопасности (шлюз прикладного уровня).

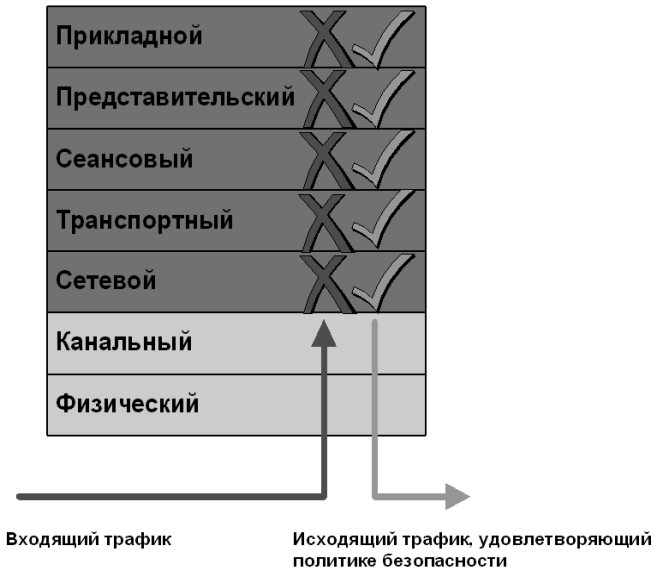


Рис. 3.4. Место МЭ экспертного уровня в модели OSI

Первые реализации таких компонентов, называемые пакетными фильтрами с динамической фильтрацией (dynamic packet filter), не функционировали на уровнях выше сеансового. Их отличие от простого пакетного фильтра состояло в том, что последний принимает решение о фильтрации трафика на основе анализа информации, содержащейся только в текущем пакете без какой-либо логической связи с предыдущими обработанными пакетами, в то время как при динамической фильтрации учитывается контекст установленных или устанавливаемых соединений.

Инспекционный модуль более поздних реализаций МЭ экспертного уровня имеет доступ ко всему содержимому пакета и может анализировать служебные поля заголовков протоколов всех уровней модели OSI (в том числе прикладного) и пользовательские данные. В дополнение к этому инспекционный модуль заносит в динамически создаваемую таблицу состояния связей всю информацию о сетевых соединениях, но, в отличие от шлюза сеансового уровня, создает записи виртуальных соединений как для протокола TCP, так и для протокола UDP. Инспекционный модуль МЭ экспертного уровня загружается в ядро операционной системы и располагается между канальным и сетевым уровнями модели OSI, что обеспечивает обработку всего входящего и исходящего трафика на всех сетевых интерфейсах системы.

Особенность функционирования МЭ экспертного уровня состоит в том, что он не оказывает посреднических услуг сетевого взаимодействия на сеансовом и прикладном уровнях модели OSI. Вместо этого он использует специфические технологии распознавания допустимых соединений (в том числе с динамически назначаемыми номерами портов) и улучшенные алгоритмы обработки данных уровня приложения.

3.3. Политика межсетевого экранирования

При настройке политики межсетевого экранирования рассматривают два аспекта сетевой безопасности: политику доступа к сетевым ресурсам и политику реализации собственно МЭ. Политика доступа к сетевым ресурсам отражает общие требования по безопасности той или иной организации, и при ее разработке должны быть сформулированы правила доступа пользователей к различным сервисам, используемым в организации. Указанные правила описывают, какой внутренний (внешний) пользователь (группа пользователей), когда, с какого внутреннего (внешнего) узла сети и каким сервисом может воспользоваться с уточнением в случае необходимости способов аутентификации пользователей и адресов целевых серверов.

Политика реализации МЭ определяет, каким образом применяется политика доступа к сетевым ресурсам, и в ряде случаев зависит от используемых сервисов и выбранных средств построения экрана. Как правило, при выборе политики реализации МЭ останавливаются на одной из двух базовых стратегий:

- разрешать все, что явно не запрещено;
- запрещать все, что явно не разрешено.

Хотя может показаться, что эти две стратегии очень просты и почти не отличаются друг от друга, на самом деле это не так. При выборе первой стратегии МЭ по умолчанию разрешает все сервисы, которые не указаны как запрещенные. В этом случае для обеспечения безопасности сети придется создавать правила, которые учитывали бы все возможные запреты. Это не только приведет к необходимости описания большого количества правил, но и заставит пересматривать их при появлении каждого нового протокола или сервиса, которые существующими правилами не охватываются.

Вторая стратегия строже и безопаснее. Намного проще управлять МЭ, запретив весь трафик по умолчанию и задав правила, разрешающие прохождение через границу сети только необходимых протоколов и сервисов. Запрет всего трафика по умолчанию обеспечивается вводом правила «Запрещено все» в последней строке таблицы фильтрации. Однако в ряде случаев, в частности при использовании простого пакетного фильтра, описание правил допустимых сервисов также сопряжено с трудоемким процессом, требующим досконального знания алгоритмов функционирования протоколов в рамках того или иного сервиса.

3.4. Архитектура МЭ

После определения требований по безопасности защищаемой сети и разработки политики доступа к сетевым ресурсам возникает задача проектирования МЭ. В зависимости от требований к межсетевому обмену организации и степени обеспечиваемой защищенности периметра ее сети в МЭ может входить от одного до нескольких рассмотренных компонентов. Состав и способ их взаимного расположения определяет архитектуру МЭ. Существуют следующие базовые схемы построения МЭ (могут быть модифицированы в другие варианты конфигурации) на основе:

- фильтрующего маршрутизатора;
- двудомного узла (узла с двумя сетевыми интерфейсами);
- экранирующего узла;
- экранирующей сети.

МЭ на основе фильтрующего маршрутизатора представляет собой аппаратный или программный маршрутизатор на периметре защищаемой сети, в котором определен набор правил, устанавливающих разрешенные сетевые сервисы (рис. 3.5). Каждый сетевой пакет перед принятием решения о его маршрутизации проверяется на принадлежность к разрешенному типу трафика. Достоинства и недостатки данной схемы МЭ определяются возможностями функционирующего на маршрутизаторе пакетного фильтра.

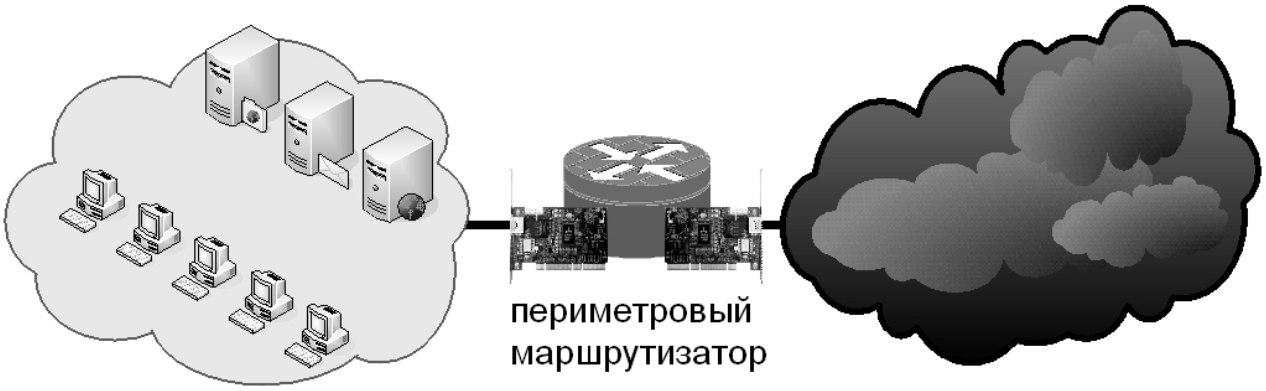


Рис. 3.5. МЭ на основе фильтрующего маршрутизатора

МЭ на основе двудомного узла представляет собой компьютер с двумя сетевыми интерфейсами, один из которых подключен к защищаемой внутренней сети, а второй — к внешней (рис. 3.6). Стандартная служба маршрутизации сетевых пакетов в ОС двудомного узла отключается для того, чтобы непосредственное взаимодействие между узлами внутренней и внешней сети было невозможным. Межсетевое взаимодействие в рамках разрешенных сервисов обеспечивается прокси-сервером, функционирующим на двудомном узле. Схема по сравнению с предыдущей характеризуется большей степенью безопасности, но предоставляемый пользователям сети набор сервисов ограничен и определяется ПО прокси-сервера.

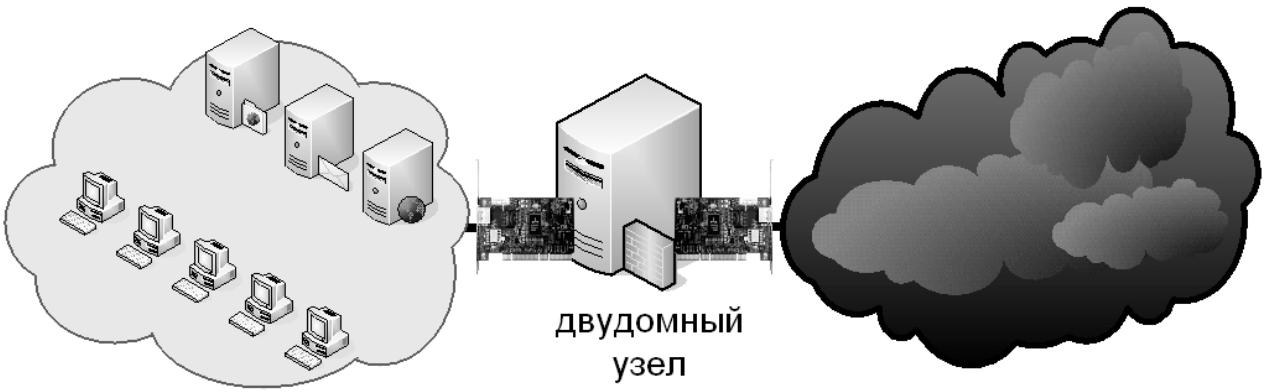


Рис. 3.6. МЭ на основе двудомного узла

МЭ на основе экранирующего узла представляет собой комбинацию предыдущих схем: в состав его входят фильтрующий маршрутизатор на периметре и прокси-сервер, функционирующий на узле-бастионе с одним интерфейсом, во внутренней сети (рис. 3.7). Пакетный фильтр на маршрутизаторе конфигурируется таким образом, что разрешенный входящий и выходящий сетевой трафик обязательно проходит через узел-бастион. Схема характеризуется большей гибкостью по сравнению со схемой МЭ на основе двудомного узла, так как сервис, не поддерживаемый прокси-сервером, может быть разрешен напрямую через маршрутизатор.

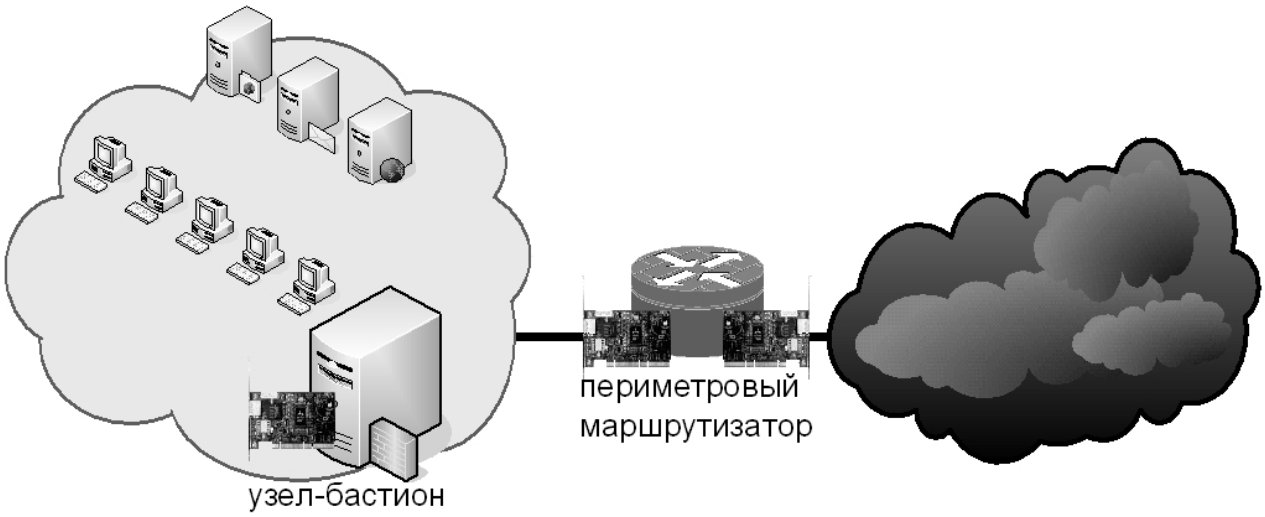


Рис. 3.7. МЭ на основе экранирующего узла

Схема МЭ на основе экранирующей сети представляет собой развитие предыдущей схемы и отличается от нее наличием дополнительного маршрутизатора (рис. 3.8). Между внешним и внутренним фильтрующими маршрутизаторами создается «менее защищаемая» сеть, называемая периметровой сетью или демилитаризованной зоной (DMZ), которая «экранирует» защищаемую сеть от внешнего мира. Как правило, в периметровой сети устанавливаются узлы с прокси-сервером и серверами открытых сервисов.

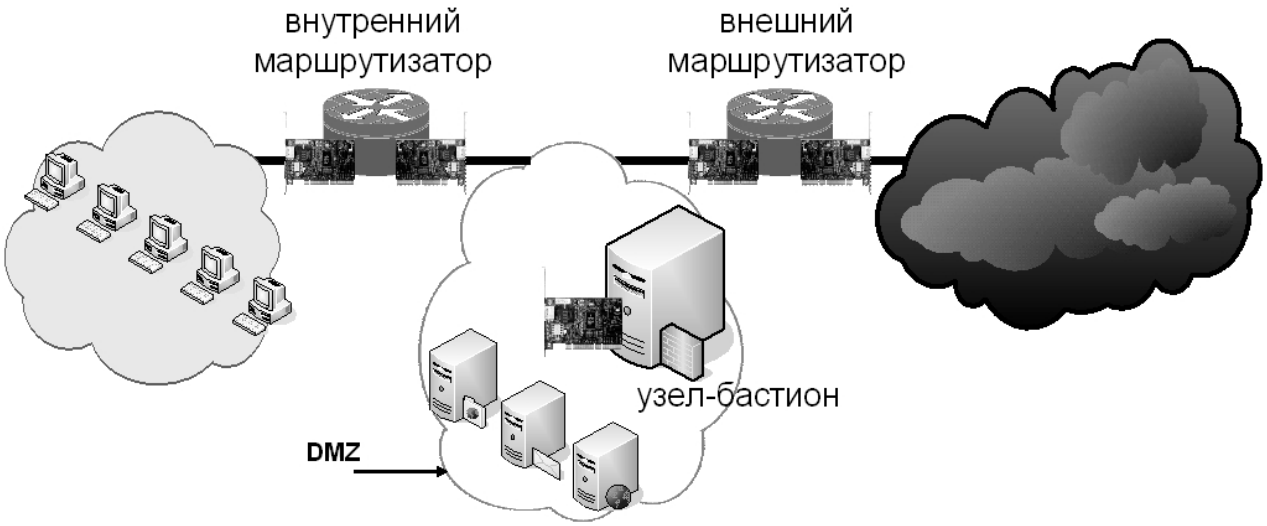


Рис. 3.8. МЭ на основе экранирующей сети

3.5. Пример реализации политики МЭ

Для иллюстрации возможностей технологий межсетевого экранирования рассмотрим два различных варианта решения следующей задачи. Пусть согласно политике безопасности некоторой организации для пользователей защищаемой сети необходимо обеспечить только сервис электронной почты, т. е. МЭ должен обеспечивать прохождение только почтового трафика между любым внутренним клиентом и определенным почтовым сервером во внешней сети по протоколам SMTP и POP3.

Первый вариант решения задачи — использование схемы МЭ на основе двудомного узла, соединяющего внутреннюю и внешнюю сети. На этом узле, выполняющем функции прокси-сервера, отключается служба маршрутизации пакетов и устанавливается шлюз прикладного уровня, обеспечивающий функционирование только протоколов электронной почты. Программы почтовых клиентов на узлах внутренней сети настраиваются на работу с внешним почтовым сервером через данный прокси-сервер, при этом в их настройках указывается адрес внутреннего сетевого интерфейса двудомного узла. Схема информационного обмена между клиентом и сервером изображена на рис. 3.9. Сверху и снизу на рисунке стрелками показаны схемы обмена пакетами клиента и сервера при отправке почтовых сообщений и выемке почтовой корреспонденции соответственно. На рисунке цифрами изображены номера портов источника и назначения при использовании протоколов SMTP и POP3 (порты 25 и 110 прокси-сервера в реальной ситуации могут быть другими). Так как служба маршрутизации на двудомном узле отключена, то кроме пакетов, изображенных на схеме информационного обмена, никакие другие сетевые пакеты через него проходить не будут.

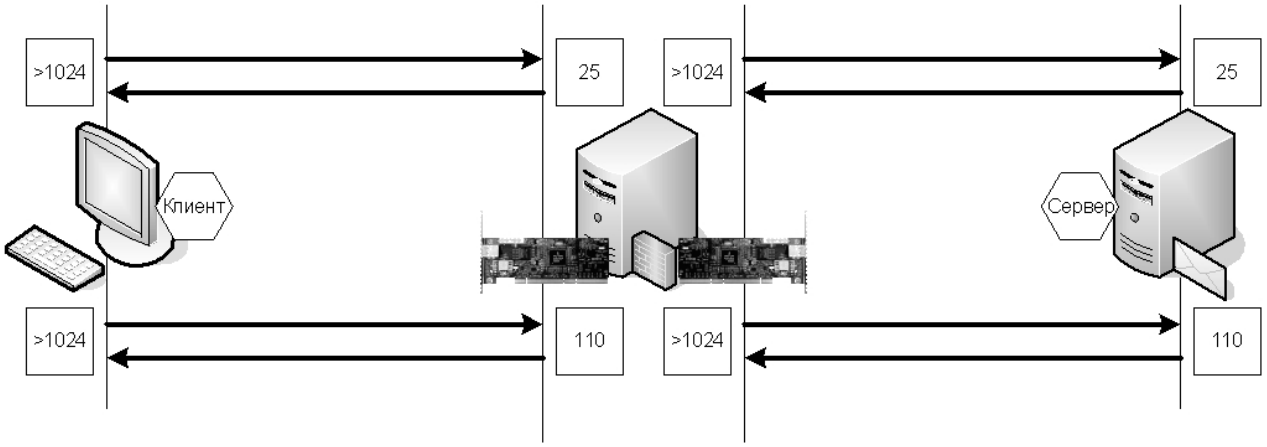


Рис. 3.9. Схема информационного обмена при использовании двудомного узла

Заметим, что некоторые прокси-серверы позволяют определить списки:

− пользователей (адресов внутренних узлов), которым (с которых) разрешается использование сервиса электронной почты;

− адресов внешних серверов, к которым разрешено подключение внутренних клиентов.

Второй вариант решения задачи — использование схемы МЭ на основе фильтрующего маршрутизатора. Соответствующая поставленной задаче схема информационного обмена между клиентом и сервером изображена на рис. 3.10.

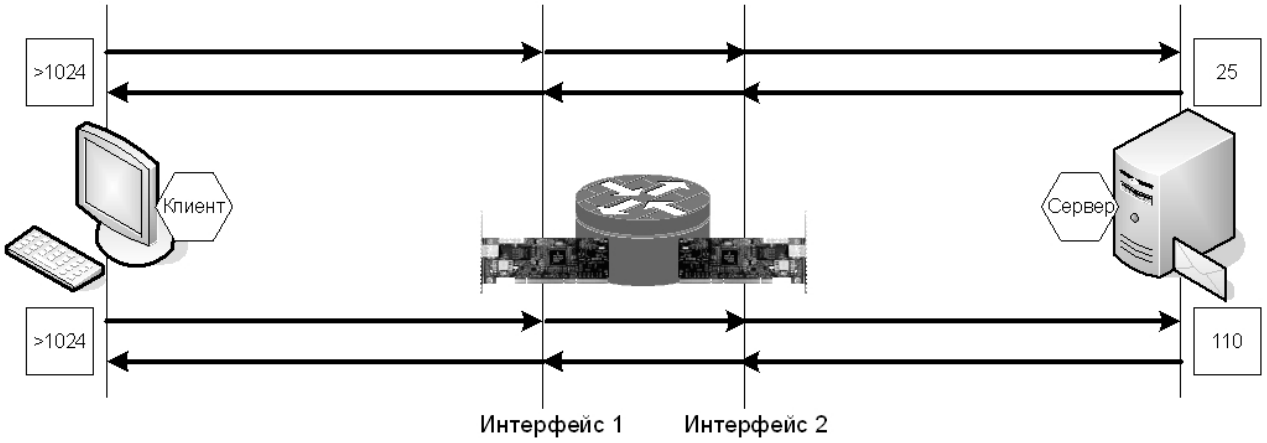


Рис. 3.10. Схема информационного обмена
при использовании фильтрующего маршрутизатора

Для пакетного фильтра определяются правила фильтрации пакетов, проходящих через сетевые интерфейсы 1 и 2, подключенные к внутренней и внешней сети соответственно. При описании правил фильтрации составляется таблица, обычно содержащая следующие поля: «Номер правила» (нумерация ведется буквами латинского алфавита), «Направление» (вход или выход), «Протокол» (тип пакета), «Адрес источника», «Порт источника», «Адрес получателя», «Порт получателя», «Действие» (разрешить, запретить, игнориро-

вать). Для ICMP-пакетов вместо полей «Порт источника/назначения» используется поле «Тип ICMP-сообщения».

В случае использования статического пакетного фильтра для каждого сетевого интерфейса разрабатывается своя таблица правил фильтрации (табл. 3.1 и табл. 3.2). Слова «Клиенты» и «Сервер» в реальной таблице фильтрации заменяются диапазоном IP-адресов клиентов и IP-адресом внешнего почтового сервера соответственно. В столбце «Направление» указывается направление сетевого пакета по отношению к МЭ. Так, направление пакетов, поступающих в МЭ, обозначается словом «Вход», а для пакетов, исходящих из МЭ, применяется слово «Выход». Символ «*» обозначает «любой».

Таблица 3.1.

Правила фильтрации пакетов для интерфейса 1

Правило	Направление	Протокол	Адрес источника	Порт источника	Адрес назначения	Порт назначения	Действие
A	Вход	TCP	Клиенты	>1024	Сервер	25	Разрешить
B	Выход	TCP	Сервер	25	Клиенты	>1024	Разрешить
C	Вход	TCP	Клиенты	>1024	Сервер	110	Разрешить
D	Выход	TCP	Сервер	110	Клиенты	>1024	Разрешить
E	*	*	*	*	*	*	Запретить

Таблица 3.2.

Правила фильтрации пакетов для интерфейса 2

Правило	Направление	Протокол	Адрес источника	Порт источника	Адрес назначения	Порт назначения	Действие
A	Выход	TCP	Клиенты	>1024	Сервер	25	Разрешить
B	Вход	TCP	Сервер	25	Клиенты	>1024	Разрешить
C	Выход	TCP	Клиенты	>1024	Сервер	110	Разрешить
D	Вход	TCP	Сервер	110	Клиенты	>1024	Разрешить
E	*	*	*	*	*	*	Запретить

В простейшем случае правила фильтрации для разных сетевых интерфейсов (табл. 3.1 и табл. 3.2) отличаются лишь значением поля «Направление». Правила *A* и *B* обеих таблиц разрешают отправку клиентами почтовых сообщений, правила *C* и *D* разрешают выемку клиентами почтовой корреспонденции, правила *E* запрещают прохождение любого трафика через сетевые интерфейсы.

В случае использования в фильтрующем маршрутизаторе программного обеспечения динамического фильтра таблицы фильтрации пакетов для каждого сетевого интерфейса заменяются одной, относящейся к МЭ в целом (табл. 3.3).

Таблица 3.3.

Правила фильтрации пакетов динамического фильтра

Правило	Адрес источника	Адрес назначения	Сервис	Действие
A	Клиенты	Сервер	SMTP (порт 25)	Разрешить
B	Клиенты	Сервер	POP3 (порт 110)	Разрешить
C	*	*	*	Запретить

Правила *A* и *B* табл. 3.3 разрешают прохождение запросов на установление соединения от внутренних клиентов к почтовому серверу, поэтому все последующие пакеты, принадлежащие разрешенному соединению, будут беспрепятственно проходить через динамический фильтр. Правило *C* запрещает прохождение любого трафика через сетевые интерфейсы маршрутизатора.

3.6. Сетевая среда лабораторной работы

Сетевая среда лабораторной работы, эмулируемая в программе VMware Workstation, представлена на рис. 3.11. Компьютер рабочего места (на рисунке — «PC») и виртуальный компьютер «IN» являются внутренними узлами защищаемой сети 192.168.20.0/24 (VMnet1 Host only). Два виртуальных компьютера «OUT1» и «OUT2» представляют «неизвестную» внешнюю сеть 10.0.0.0/8 (VMnet2). Виртуальный компьютер «FW-W98» представляет собой узел с программным обеспечением МЭ и используется для защиты периметра внутренней сети. Все IP-адреса сетевых интерфейсов виртуальных узлов назначены вручную, поэтому необходимо отключить DHCP-сервер VMware Workstation и настроить стек TCP/IP интерфейса «VMnet1 Host only» рабочего места. Для настройки IP-адресов и проверки доступности сетевых узлов используйте информацию, приведенную на рис. 3.11.

На виртуальных компьютерах «IN», «OUT1» и «OUT2» установлены ОС Windows 98 и ПО, необходимое для выполнения заданий: программа E-Serv (серверы HTTP, FTP и электронной почты), клиентские приложения Internet Explorer и Outlook Express. На виртуальном компьютере «FW-W98» установлены ОС Windows 98, простейший прокси-сервер AnalogX Proxy Server (также установлен на «IN» для реализации схемы на основе экранирующего узла), статический пакетный фильтр WinRoute Pro и анализатор сетевого трафика Ethereal. Захват и анализ сетевого трафика можно использовать в процессе настройки и диагностики МЭ, а также для изучения информационного обмена между узлами в рамках того или иного сетевого сервиса.

Для создания сетевой среды лабораторной работы в приложении VMware Workstation необходимо последовательно открыть файлы виртуальных машин с именами «IN», «FW-W98», «OUT1», «OUT2» и включить их. После загрузки программ автозапуска на компьютерах «IN», «OUT1», «OUT2» можно свернуть окна приложения E-Serv.

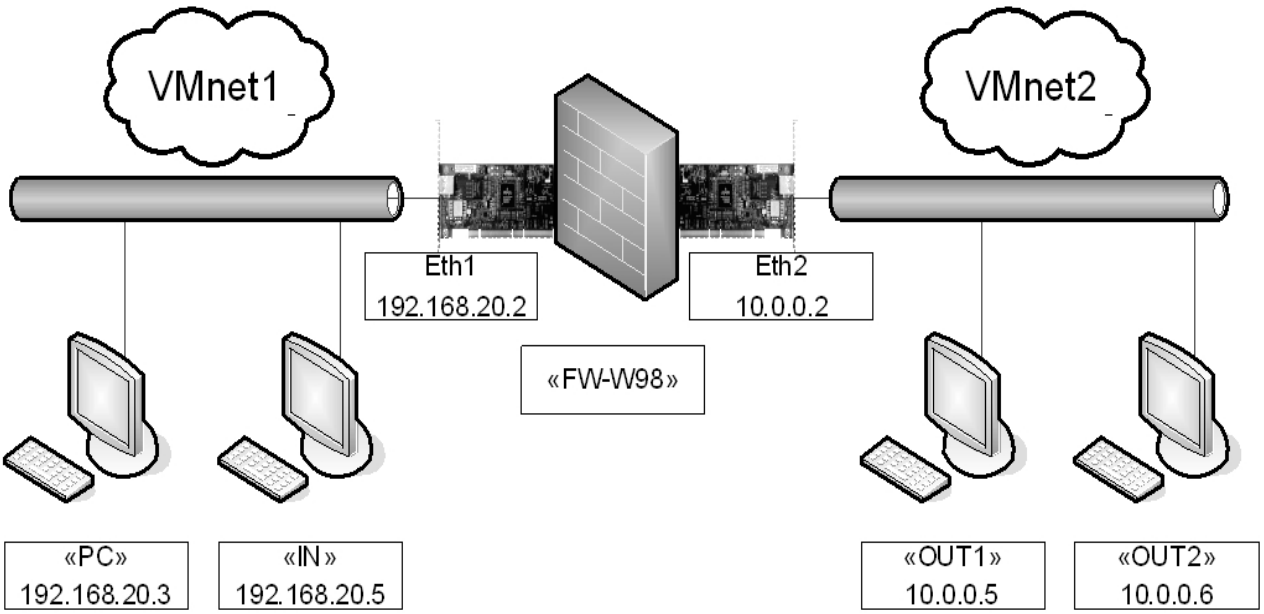


Рис. 3.11. Сетевая среда лабораторной работы

3.7. Применение МЭ на основе двудомного узла

Пусть для защиты внутренней сети используется схема МЭ на основе двудомного узла и необходимо предоставить клиентам внутренней сети доступ только к web-сервису через прокси-сервер, функционирующий на данном двудомном узле.

В качестве прокси-сервера при решении задачи используется программа AnalogX Proxy Server. Конфигурирование параметров производится в диалоговом окне, которое вызывается с помощью контекстного меню «Configure» иконки программы на панели задач. Диалоговое окно настройки параметров функционирования прокси-сервера показано на рис. 3.12.

Для поддержки программой сетевого сервиса используется соответствующая кнопка на панели «Services». Строка ввода «Proxy Binding» определяет привязку сервера к тому или иному сетевому интерфейсу узла, на котором он функционирует. Значение «disabled» указывает на то, что сервер будет обслуживать запросы клиентов, поступающие на все интерфейсы (Open state). Если задать в этом поле определенный IP-адрес (как правило, принадлежащий внутренней сети), то сервер будет обслуживать запросы клиентов, поступающие только на этот интерфейс (Closed state). Кнопка «Logging» включает режим регистрации системных событий (файл «proxy.log» в каталоге программы). По умолчанию все клиентские запросы по протоколу HTTP прокси-сервер ожидает на порт 6588.

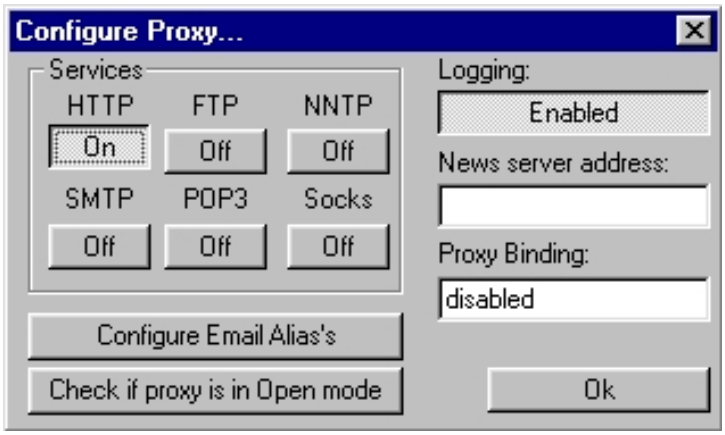


Рис. 3.12. Окно настройки программы AnalogX Proxy Server

Настройка программы Internet Explorer (рис. 3.13) узла-клиента на работу через прокси-сервер осуществляется с помощью закладки «Подключения» свойств обозревателя.

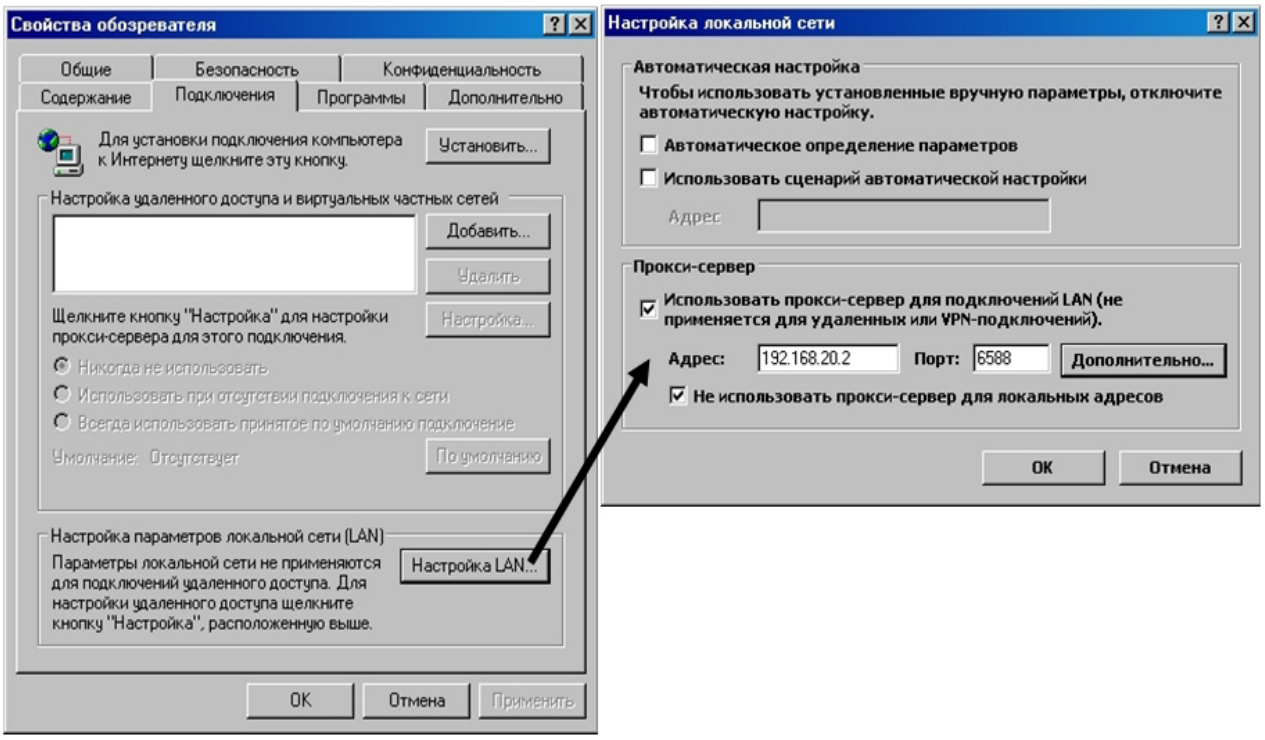


Рис. 3.13. Настройка программы Internet Explorer на работу через прокси-сервер

ВЫПОЛНИТЬ!

1. Убедиться в доступности узлов «OUT1» и «OUT2» с узла «IN» и наоборот.
2. Отключить маршрутизацию на «FW-W98», выбрав пункт «Stop WinRoute Engine» контекстного меню иконки программы WinRoute на панели задач (изображение иконки должно смениться на). Убедиться в недоступности узлов «OUT1» и «OUT2» с узла «IN» и наоборот.

3. Запустить на «FW-W98» прокси-сервер и настроить его на обслуживание запросов HTTP (рис. 3.12).
4. На компьютерах «IN» и «OUT1» настроить программу Internet Explorer на работу через прокси-сервер (рис. 3.13) и убедиться в возможности работы с web-серверами на этих узлах (обратиться с «IN» к серверу на «OUT1» и наоборот). Это должно быть возможным, так как прокси-сервер, функционирует в режиме «Open state».
5. Перевести прокси-сервер в режим «Closed state», указав в строке ввода «Proxy Binding» адрес интерфейса «FW-W98», принадлежащий внутренней сети.
6. Проверить возможность обращения клиента «IN» к серверу «OUT1» и недоступность сервера «IN» для клиента «OUT1».

Задача решена: защищаемая сеть извне недоступна. Все клиенты внутренней сети, настроенные на работу через прокси-сервер обеспечены web-сервисом. Никакой другой трафик кроме web-сервиса между сетями проходить не будет.

7. Вернуть настройки приложения AnalogX Proxy Server в исходное состояние и закрыть его. На компьютерах «IN» и «OUT1» вернуть настройки приложения Internet Explorer в исходное состояние.

3.8. Применение МЭ на основе фильтрующего маршрутизатора

Пусть для защиты внутренней сети используется схема МЭ на основе фильтрующего маршрутизатора со статическим фильтром и необходимо предоставить всем клиентам внутренней сети только лишь web-сервис (рис. 3.14).

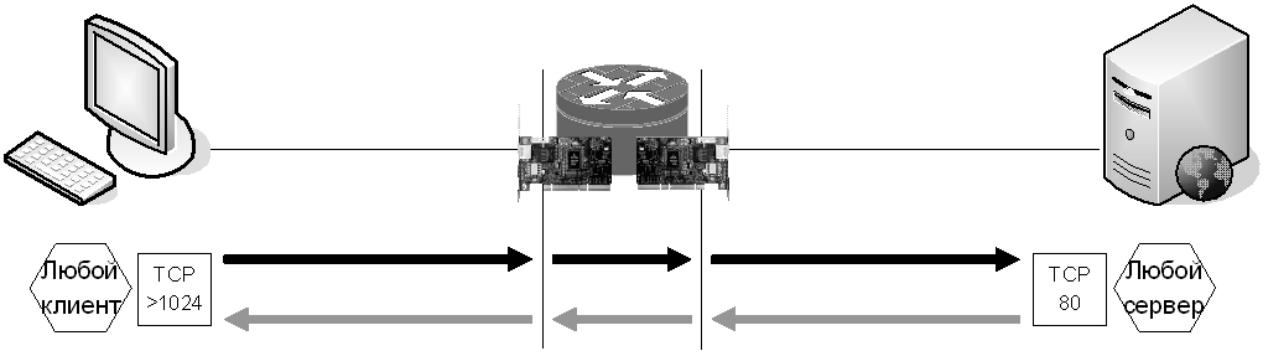


Рис. 3.14. Схема информационного обмена по условию задачи

В качестве статического фильтра пакетов используется программа WinRoute на узле «FW-W98». Запуск программы происходит автоматически при загрузке ОС. Конфигурирование параметров функционирования фильтра, т. е. определение правил фильтрации, производится в диалоговом окне «Packet Filter», которое изображено на рис. 3.15.

На вкладках «Incoming» и «Outgoing» диалогового окна добавляются новые (Add…), редактируются (Edit…) и удаляются (Remove) имеющиеся правила фильтрации каждого сетевого интерфейса, присутствующего в систе-

ме для входящих и исходящих сетевых пакетов соответственно. Изменить порядок применения правил к обрабатываемым пакетам можно с помощью кнопок «Вверх» и «Вниз», находящихся в правой части диалогового окна. Для вступления правил фильтрации в силу следует использовать кнопку «Применить». В системе, как изображено на рис. 3.15, присутствуют два сетевых адаптера: первый, называемый «In AMD PCNET Family Ethernet Adapter», подключен к внутренней сети, второй, называемый «Out AMD PCNET Family Ethernet Adapter», подключен к внешней сети.

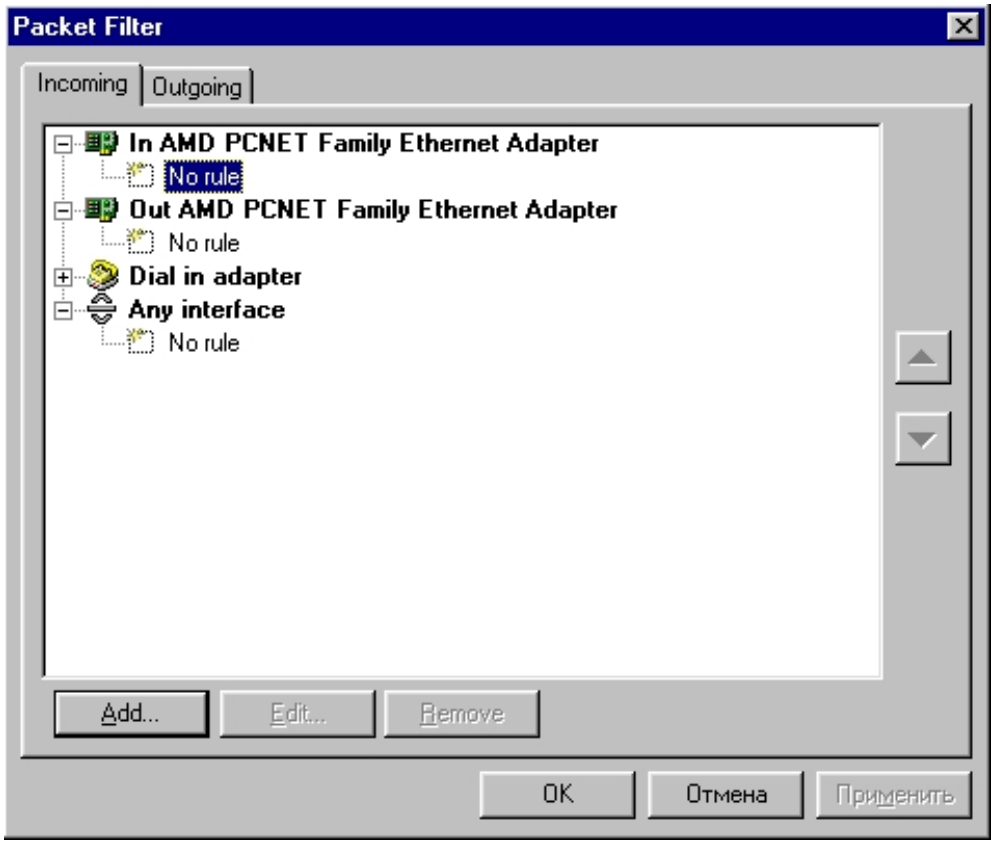


Рис. 3.15. Окно настройки фильтра пакетов

Следует учесть, что в статическом фильтре для каждого направления сетевого взаимодействия в рамках того или иного сервиса правила фильтрации, разрешающие прохождение сетевых пакетов через маршрутизатор, необходимо определять, как минимум, два раза. Например, при взаимодействии клиента с сервером прохождение пакетов клиента внутренней сети к внешнему серверу определяется разрешающими правилами в последовательности: входящее для внутреннего сетевого адаптера, затем исходящее для внешнего сетевого адаптера, а прохождение обратных пакетов — в последовательности: входящее для внешнего сетевого адаптера, затем исходящее для внутреннего сетевого адаптера.

Необходимость задания разрешающих правил одновременно для двух сетевых интерфейсов поясним следующим примером. Пусть по условиям некоторой задачи необходимо предоставить клиентам внутренней сети какой-

либо сервис. Решить поставленную задачу можно несколькими способами, определяя правила фильтрации пакетов для одного из интерфейсов или для двух одновременно, но только последний из них надлежащим образом обеспечит защиту самого МЭ от атак из внешней и внутренней сети (рис. 3.16).

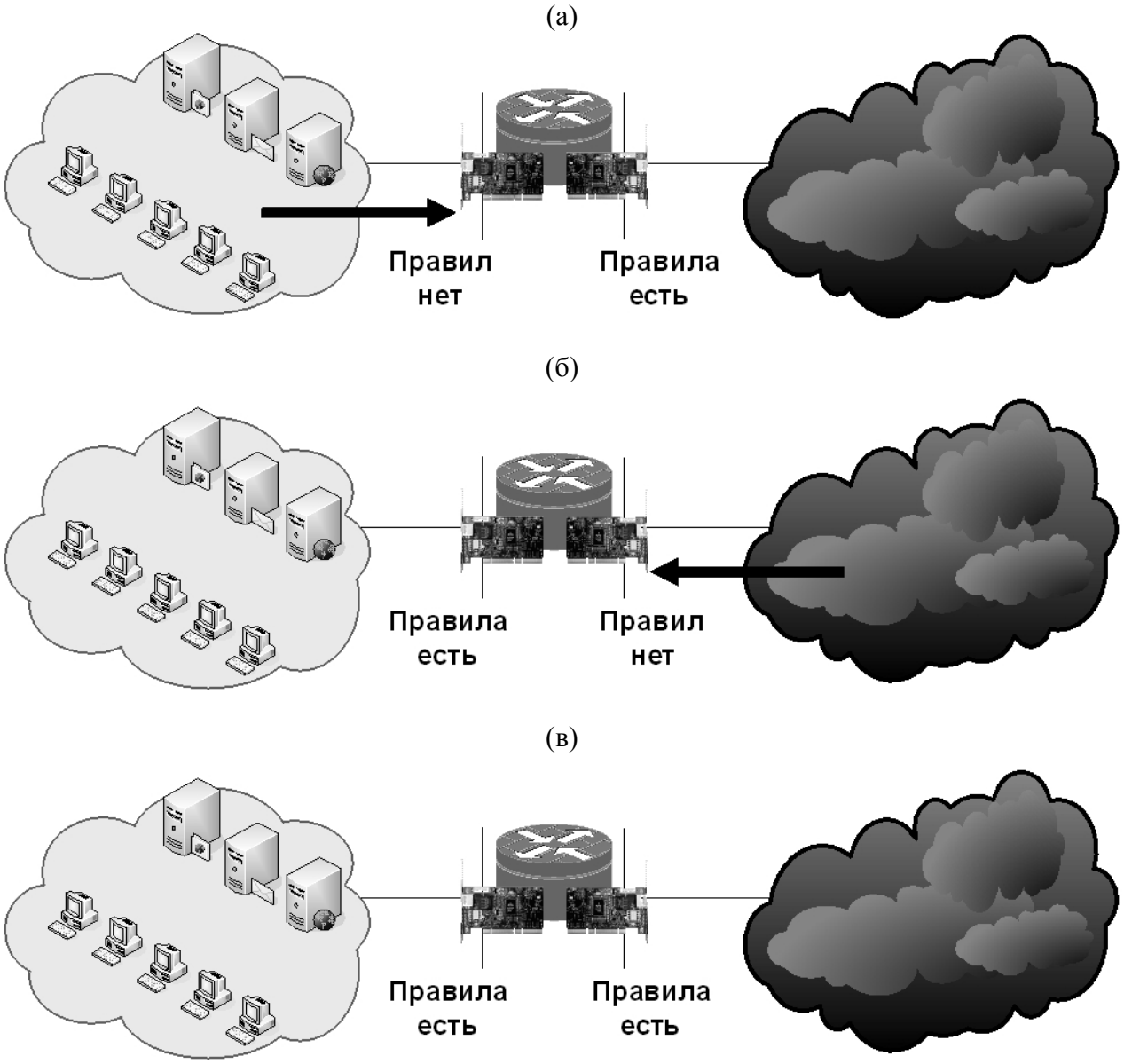


Рис. 3.16. Варианты определения правил фильтрации для интерфейсов МЭ: (а) — явная угроза со стороны внутренней сети; (б) — явная угроза со стороны внешней сети; (в) — явная угроза со стороны сетей отсутствует

Создание или редактирование правил фильтрации производится в диалоговом окне, изображенном на рис. 3.17. В общем случае для правила определяются: протокол (Protocol), адреса и порты отправителя (Source) и получателя (Destination), а также действие (Action), которое выполняется над пакетом, удовлетворяющим заданным критериям фильтрации. В зависимости от типа протокола некоторые поля могут отсутствовать, а присутствовать дополнительные критерии фильтрации. Например, для протокола IP не имеют

смысла значения портов отправителя и получателя, а для протокола ICMP имеется возможность фильтрации по типу сообщения. Возможное действие над пакетом определяется на панели «Action» следующим образом: разрешить (Permit), отбросить (Drop), запретить с извещением отправителя об ошибке (Deny). На панели «Log Packet» определяется режим регистрации событий при обработке пакета.

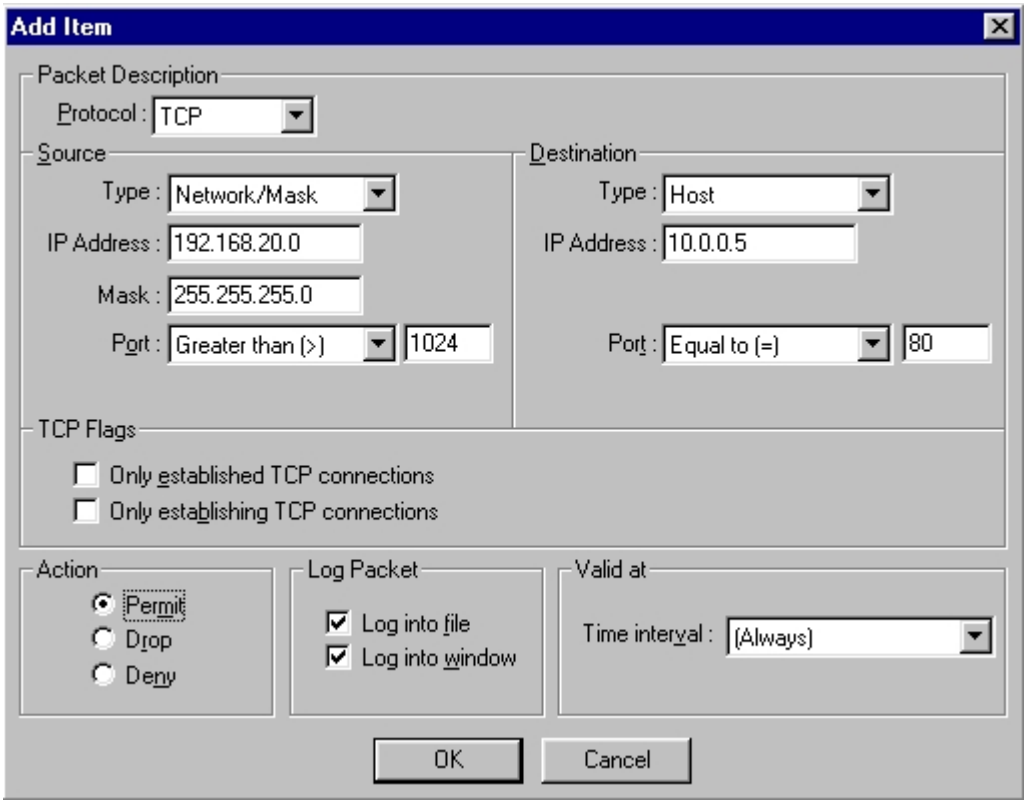


Рис. 3.17. Окно определения правила фильтрации

На рис. 3.17 изображены критерии фильтрации для правила, разрешающего прохождение пакетов любого клиента внутренней сети к внешнему web-серверу с адресом 10.0.0.5 в любое время суток.

ВЫПОЛНИТЬ!

8. Запустить приложение администрирования WinRoute с помощью пункта «WinRoute Administration…» контекстного меню иконки программы на панели задач, подключившись к «LocalHost» как пользователь Admin с пустым паролем.
9. Через меню ***Settings ⇒ Advanced ⇒ Packet Filter…*** вызвать диалоговое окно управления таблицей фильтрации пакетов.
10. Создать необходимые правила для разрешения web-сервиса внутренним пользователям и правило, запрещающее все остальное (для этой цели можно использовать последнюю строку «Any interface»).
11. Проверить правильность функционирования МЭ.

Для приобретения навыков администрирования пакетного фильтра самостоятельно выполните следующие задачи.

Задача 1. Необходимо ограничить пользователя компьютера «PC» в использовании web-сервиса так, чтобы он мог работать только с сервером «OUT2» (рис. 3.18). Обратите внимание на правильную последовательность определения правил фильтрации.

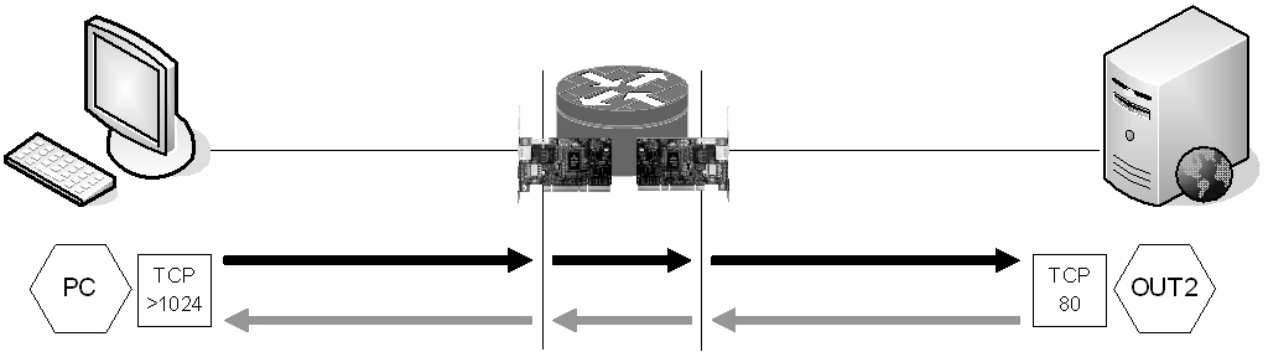


Рис. 3.18. Схема информационного обмена по условию задачи № 1

Задача 2. При начальных условиях предыдущей задачи необходимо предоставить внутренним пользователям возможность использования команды Ping для проверки доступности внешних узлов, но исключить такую возможность для внешних узлов при сканировании узлов защищаемой сети (рис. 3.19).

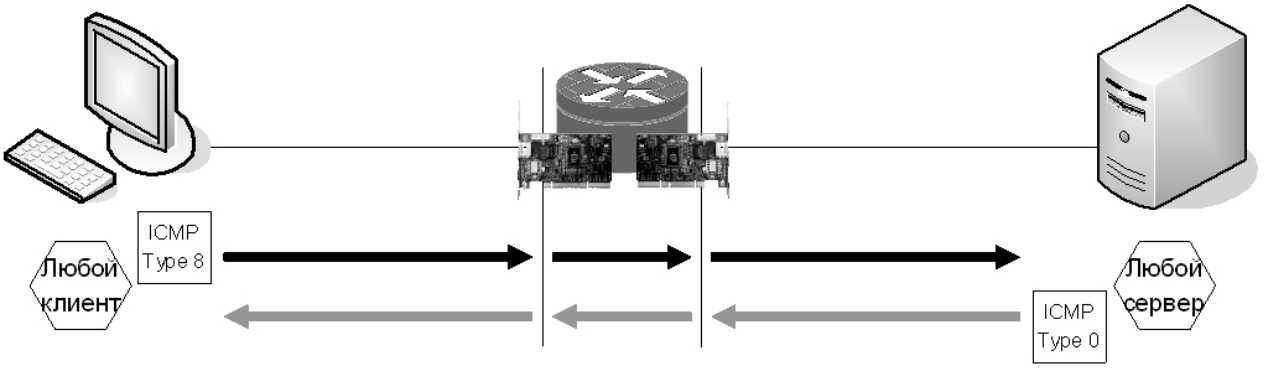


Рис. 3.19. Схема информационного обмена по условию задачи № 2

Задача 3. При начальных условиях предыдущей задачи необходимо предоставить всем клиентам внутренней сети FTP-сервис (рис. 3.20). Учтите, что на рис. 3.20 показан типовой обмен клиента и FTP-сервера, а программа E-Serv, установленная на виртуальных компьютерах, инициализирует передачу данных не с порта 20, а с порта >1024.

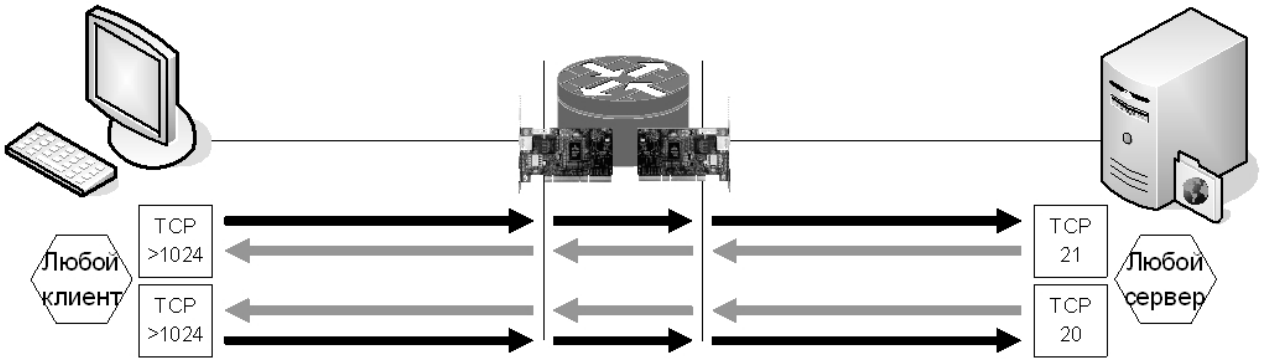


Рис. 3.20. Схема информационного обмена по условию задачи № 3

Задача 4. При начальных условиях предыдущей задачи необходимо ограничить пользователя компьютера «IN» в использовании FTP-сервиса так, чтобы он мог работать только с сервером «OUT2» (рис. 3.21).

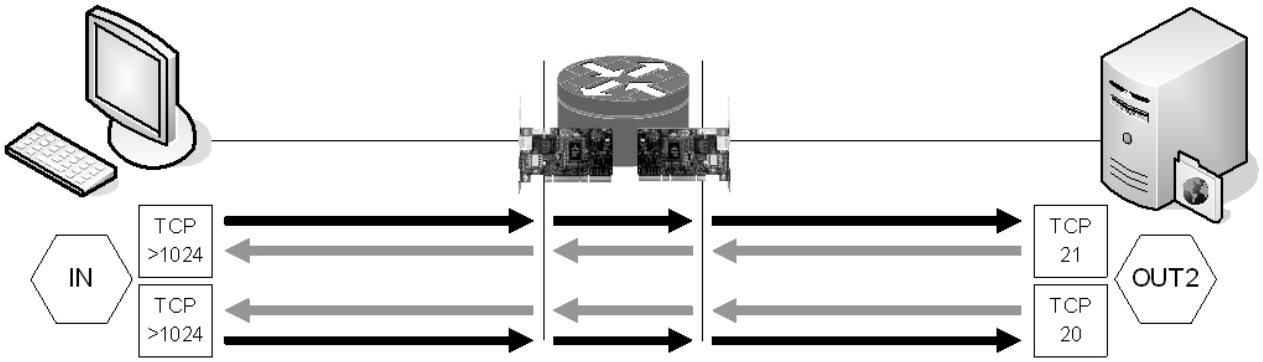


Рис. 3.21. Схема информационного обмена по условию задачи № 4

Задача 5. При начальных условиях предыдущей задачи необходимо предоставить пользователю компьютера «IN» сервис электронной почты (рис. 3.22). На компьютере «IN» приложение Outlook Express настроено на почтовый ящик с адресом «U1@mail.ru» на сервере «OUT1». Выемка почтовой корреспонденции осуществляется по протоколу POP3. Произведите отправку электронного сообщения от имени пользователя U1 самому себе.

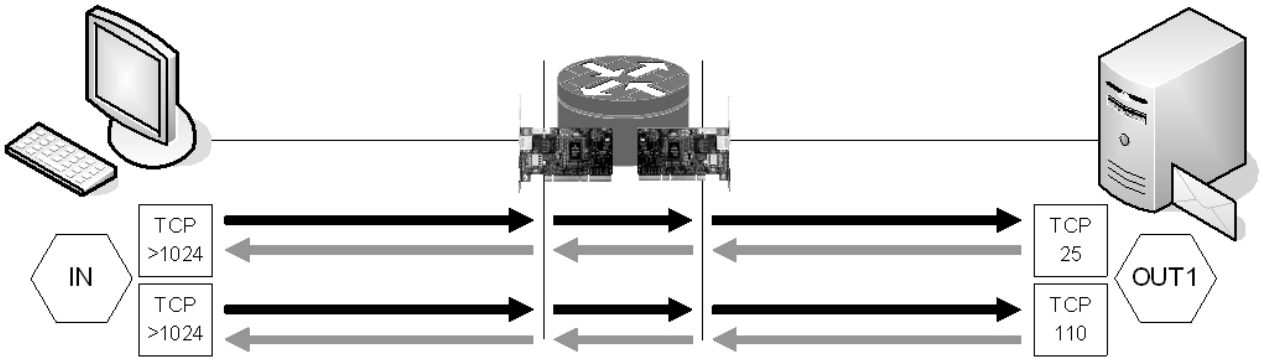


Рис. 3.22. Схема информационного обмена по условию задачи № 5

Задача 6. При начальных условиях предыдущей задачи необходимо предоставить пользователю внешнего узла «OUT1» возможность работы с внутренним web-сервером «IN» (рис. 3.23).

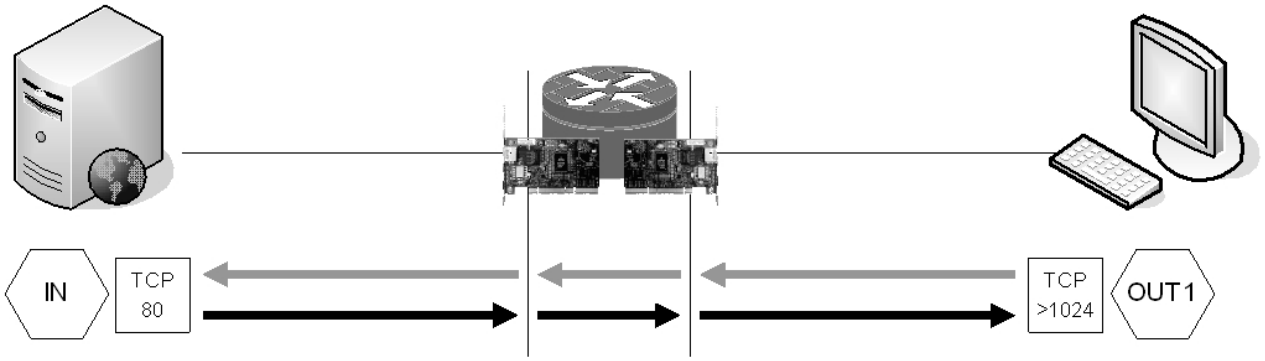


Рис. 3.23. Схема информационного обмена по условию задачи № 6

Задача 7. При начальных условиях предыдущей задачи необходимо предоставить пользователю внешнего узла «OUT2» возможность работы с внутренним FTP сервером «IN» (рис. 3.24).

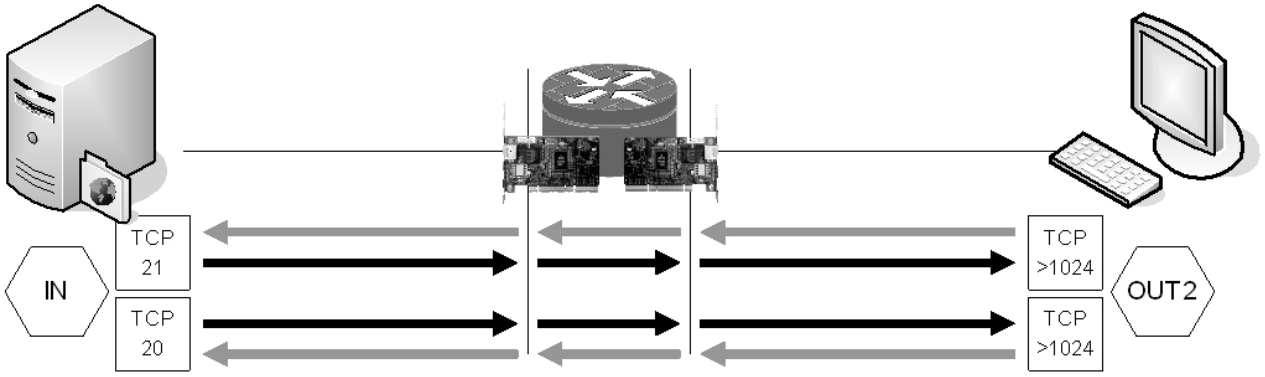


Рис. 3.24. Схема информационного обмена по условию задачи № 7

3.9. Применение МЭ на основе экранирующего узла

При построении схемы МЭ на основе экранирующего узла для защиты сети используют внутренний узел бастионного типа, на котором запущена программа прокси-сервера, выполняющая поддержку необходимых сервисов. На остальных узлах внутренней сети клиентские приложения настраиваются на работу через прокси-сервер. На фильтрующем маршрутизаторе, находящемся на периметре сети, определяются правила фильтрации сетевых пакетов таким образом, чтобы из внутренней сети во внешнюю разрешался только трафик с узла бастионного типа, а весь допустимый входящий в защищаемую сеть трафик направлялся только на узел бастионного типа. Такая схема построения МЭ похожа на схему МЭ на основе двудомного узла, но обладает большей гибкостью по сравнению с ней, так как для некоторых узлов защи-

щаемой сети допустимы исключения в виде предоставления тех или иных сервисов напрямую через фильтрующий маршрутизатор.

Пусть для защиты внутренней сети используется схема МЭ на основе экранирующего узла. Необходимо предоставить клиентам внутренней сети только лишь web-сервис. В качестве экранирующего шлюза используйте виртуальный компьютер «IN», в составе которого имеется приложение AnalogX Proxy Server, а в качестве клиента — компьютер рабочего места (рис. 3.25).

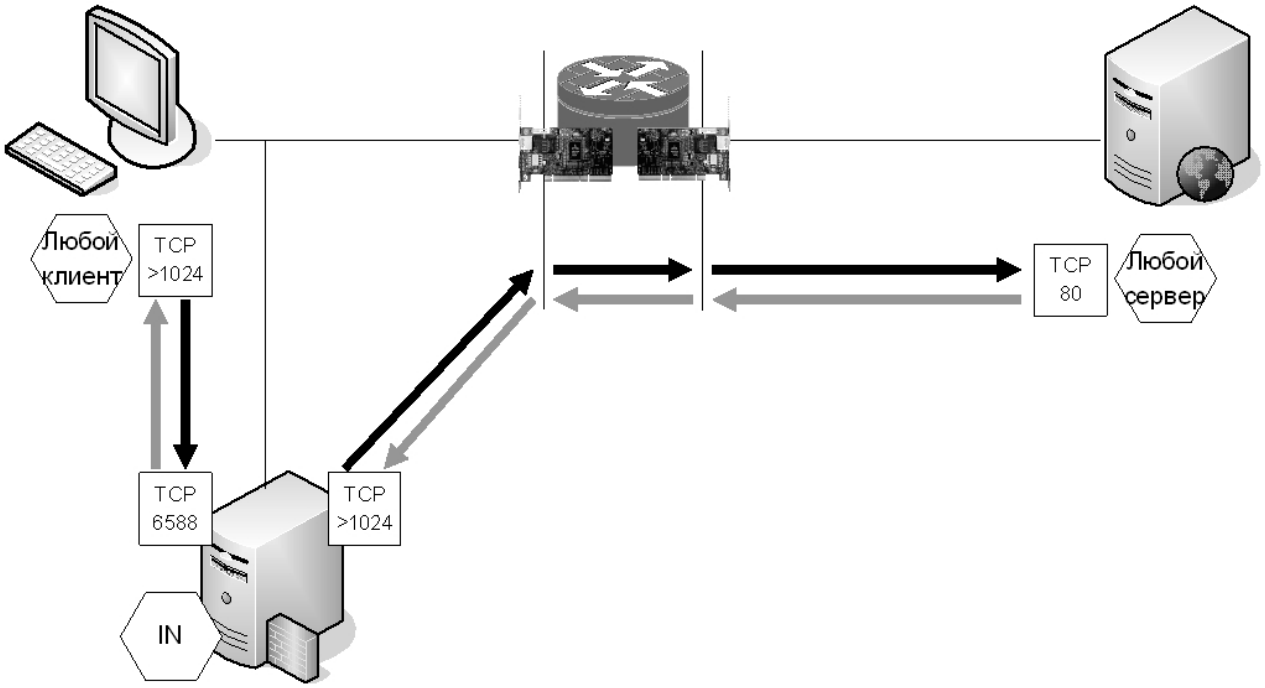


Рис. 3.25. Схема информационного обмена по условию задачи

ВЫПОЛНИТЬ!

12. На узле «IN» запустить прокси-сервер и установить необходимые параметры его функционирования.
13. На узле «FW-W98» для пакетного фильтра WinRoute создать необходимые правила для разрешения web-сервиса и правило, запрещающее все остальное.
14. Настроить программу Internet Explorer на узле «PC» для работы через прокси-сервер и убедиться в возможности работы с внутренним и внешними web-серверами.
15. Проверить правильность функционирования МЭ.
16. Вернуть настройки программного обеспечения узлов «PC», «IN», «FW-W98» в исходное состояние.

3.10. Применение технологии трансляции сетевых адресов

Трансляция сетевых адресов (NAT) — технология, которая позволяет маршрутизатору выполнять функцию прокси-сервера по сокрытию информации об узлах внутренней сети. В целях сокрытия информации о внутренней сети, маршрутизатор с NAT функционирует следующим образом:

- при передаче запросов клиентов защищаемой сети во внешнюю сеть заменяет их IP-адреса на IP-адрес своего внешнего интерфейса (может использоваться и диапазон IP-адресов);
- при возврате ответов серверов клиентам производит обратную замену: свой адрес в поле получателя меняет на адрес клиента, отправившего исходный запрос (рис. 3.26).

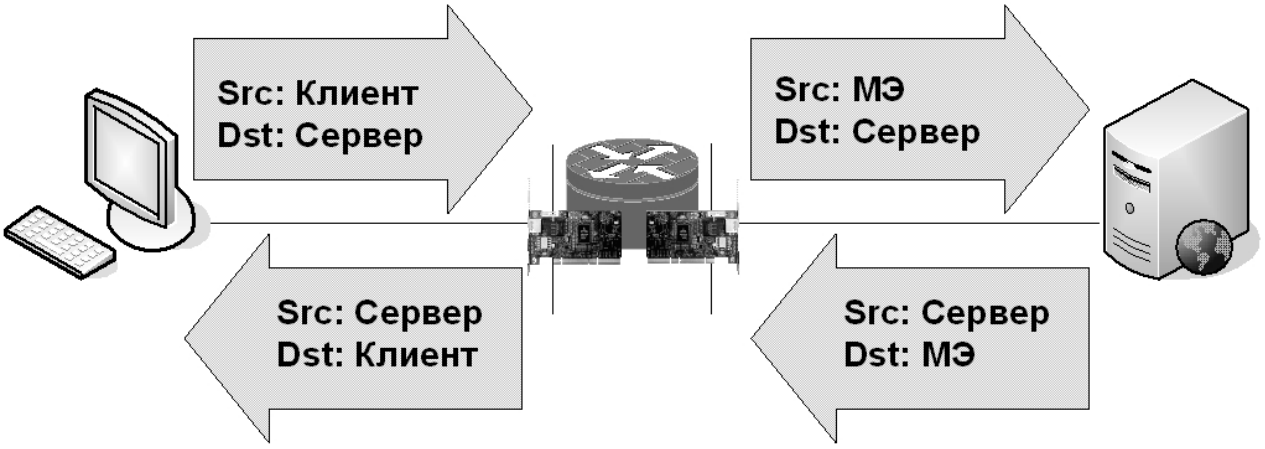


Рис. 3.26. Технология NAT

Преимущество использования трансляции сетевых адресов состоит в том, что при подключении внутренней сети к сети Интернет технология NAT позволяет существенно увеличить адресное пространство за счет использования IP-адресов из диапазона частных сетей, не обрабатываемых маршрутизаторами Интернет.

Существует несколько методов реализации NAT. Одни трансляторы адресов осуществляют это посредством статического присваивания адресов (static address assignment), при этом адрес клиента внутренней сети связывается с фиксированным внешним IP-адресом. Другие трансляторы, функционирующие по принципу динамического присваивания адресов (dynamic address assignment), выделяют клиентам внутренней сети внешний IP-адрес по мере поступления запросов. После освобождения клиентом внешнего IP-адреса он возвращается маршрутизатором в список свободных адресов и может быть предоставлен другому клиенту.

Концепция трансляции сетевых адресов, о которой шла речь до сих пор, обычно называется базовой трансляцией адресов (basic NAT). Ее реализация требует наличия нескольких внешних IP-адресов для обеспечения одновременной работы нескольких клиентов внутренней сети. Это означает, что чис-

ло внешних IP-адресов маршрутизатора с NAT должно быть равно максимально возможному числу активных исходящих соединений. Чтобы расширить число возможных исходящих соединений и при этом не увеличивать количество отведенных маршрутизатору внешних адресов в новой форме NAT, которая называется трансляцией портов сетевых адресов (NAPT), используется замена одновременно и IP-адреса и номера порта отправителя. Таким образом, один IP-адрес можно распределить между множеством клиентов внутреннней сети просто за счет изменения номера порта отправителя. Иногда для обозначения NAPT употребляются термины «PAT» (трансляция адресов портов) и «Overloading NAT».

Пусть для защиты внутренней сети используется схема МЭ на основе фильтрующего маршрутизатора с включенной функцией NAT. В учебных целях для пакетного фильтра никакие правила фильтрации не определяются.

ВЫПОЛНИТЬ!

17. На маршрутизаторе «FW-W98» включить функцию NAT для внешнего сетевого интерфейса МЭ. В программе WinRoute функция NAT включается посредством диалогового окна, которое вызывается командой меню ***Settings ⇒ Interface Table…*** (рис. 3.27).

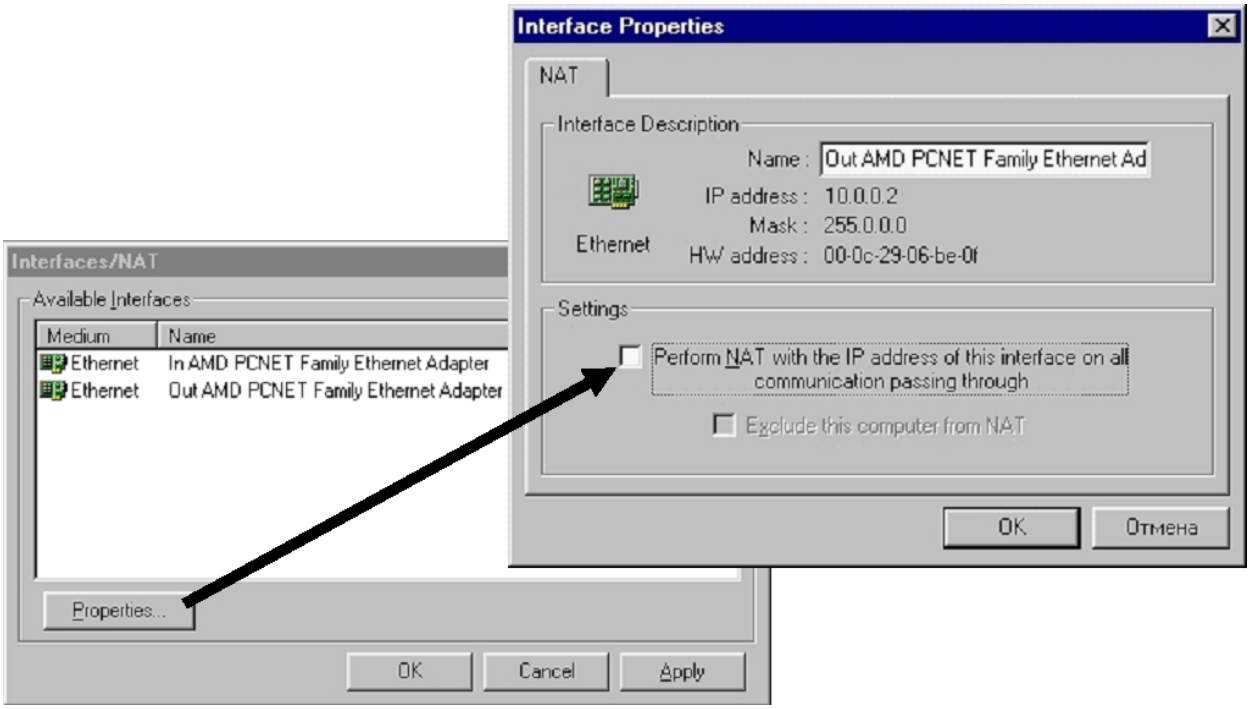


Рис. 3.27. Включение функции NAT в WinRoute

18. Проверить, что для внутренних клиентов существует возможность доступа к внешним серверам «OUT1» и «OUT2».
19. Проверить, что для внешних клиентов отсутствует возможность доступа к внутреннему серверу «IN».

20. Одновременно на двух клиентах внутренней сети запустить команду Ping с параметром -t к одному и тому же внешнему узлу. Объяснить, на какой информации основывается решение МЭ по распределению обратного трафика между клиентами при выполнении функции NAT.

Технология, называемая векторизацией адресов («address vectoring») или перенаправлением портов («port mapping»), по сути, является обратной NAT и служит для обеспечения возможности доступа извне к некоторым узлам защищаемой с помощью NAT сети. МЭ с включенной функцией перенаправления портов принимает запрос на соединение от внешнего клиента и в случае допустимости поступившего запроса переадресовывает его во внутреннюю сеть на указанный в таблице перенаправления узел, причем порт назначения внутреннего узла не обязательно должен совпадать с портом назначения в запросе внешнего узла (рис. 3.28).

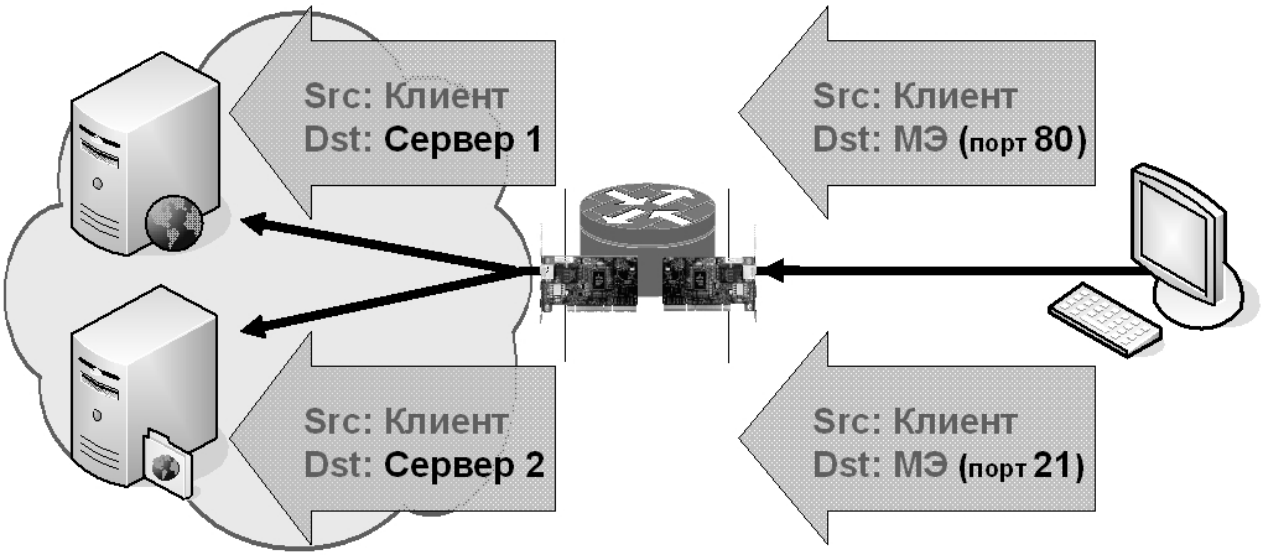


Рис. 3.28. Технология векторизации адресов

Пусть при начальных условиях предыдущей задачи необходимо дополнительно предоставить доступ внешних клиентов «OUT1» и «OUT2» к серверам Web и FTP на внутреннем узле «IN» соответственно.

В программе WinRoute функция векторизации адресов включается после добавления записей в таблицу переназначения портов посредством диалогового окна, которое вызывается командой меню ***Settings ⇒ Advanced ⇒ Port Mapping...*** (рис. 3.29). Прослушиваемый IP-адрес (Listen IP) можно оставить в значении по умолчанию, а для указания узлов, которым разрешен доступ во внутреннюю сеть (поле «Allow access only from»), необходимо предварительно создать адресную группу.

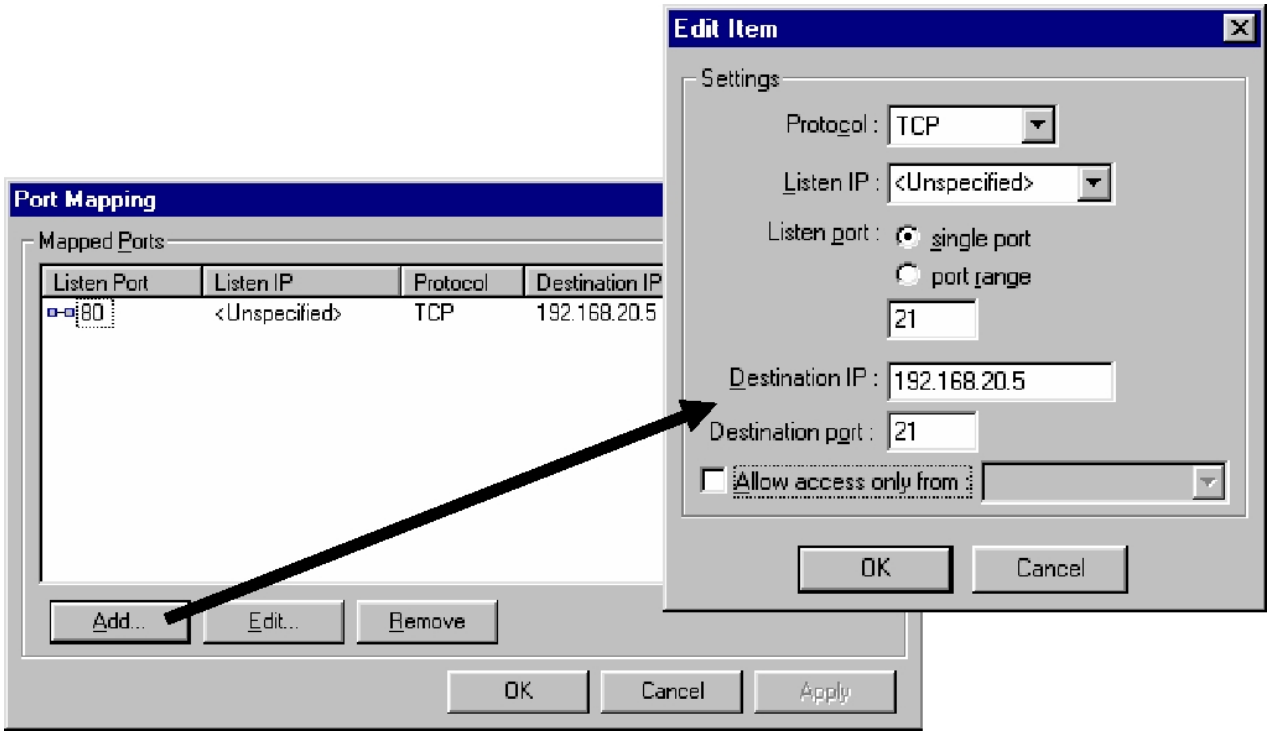


Рис. 3.29. Создание таблицы векторизации адресов

ВЫПОЛНИТЬ!

21. Создать адресную группу для web-сервиса, включив в нее узел «OUT1».
22. Создать адресную группу для FTP-сервиса, включив в нее узел «OUT2».

В программе WinRoute для создания адресных групп используется пункт меню ***Settings ⇒ Advanced ⇒ Address Groups…***

23. Создать соответствующие задаче записи таблицы переназначения портов.
24. Проверить, что для клиента «OUT1» существует возможность доступа к web-серверу на внутреннем узле «IN».
25. Проверить, что для клиента «OUT2» существует возможность доступа к FTP-серверу на внутреннем узле «IN».

4. СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК

Система обнаружения атак — это программный или программно-аппаратный комплекс, предназначенный для выявления и по возможности предупреждения действий, угрожающих безопасности информационной системы.

Первые прототипы СОА появились в начале 1980-х годов и были ориентированы в первую очередь на защиту автономных ЭВМ, не объединенных в сеть. Обнаружение атак производилось путем анализа журналов регистрации событий постфактум. Современные системы в основном ориентированы на защиту от угроз, направленных из сети, поэтому их архитектура существенным образом поменялась. Вместе с тем основные подходы к обнаружению атак остались прежними. Рассмотрим классификацию и принципы работы СОА более подробно.

4.1. Сигнатурный анализ и обнаружение аномалий

Основные подходы к обнаружению атак практически не изменились за последнюю четверть века, и, несмотря на громкие заявления разработчиков, можно с уверенностью утверждать, что концептуально обнаружение атак базируется либо на методах *сигнатурного анализа*, либо на методах *обнаружения аномалий*. Возможно также совместное использование указанных выше методов.

Сигнатурный анализ основан на предположении, что сценарий атаки известен и попытка ее реализации может быть обнаружена в журналах регистрации событий или путем анализа сетевого трафика. В идеале администратор информационной системы должен устранить все известные ему уязвимости. На практике, однако, данное требование может оказаться невыполнимым, так как в результате может существенным образом пострадать функциональность ИС. Не исключено также, что людские и материальные затраты, необходимые для устранения этих уязвимостей, могут превысить стоимость информации, обрабатываемой системой. Системы обнаружения атак, использующие методы сигнатурного анализа, предназначены для решения обозначенной проблемы, так как в большинстве случаев позволяют не только обнаружить, но и предотвратить реализацию атаки на начальной стадии ее выполнения.

Процесс обнаружения атак в данных системах сводится к поиску *заранее известной* последовательности событий или строки символов в упорядоченном во времени потоке информации. Механизм поиска определяется способом описания атаки.

Наиболее простым является описание атаки при помощи набора правил (условий). Применительно к анализу сетевых пакетов эти правила могут включать определенные значения отдельных полей заголовка пакета (IP-адрес и порт источника или получателя, установленные флаги, размер пакета

и т. д.). При анализе журналов регистрации событий правила могут ограничивать время регистрации пользователя в системе, количество попыток неправильного ввода пароля в течение короткого промежутка времени, а также наличие изменений в критических файлах системы. Таким образом, описание атаки отражает, во-первых, характер передаваемой информации и, во-вторых, совокупность реакций системы на реализацию атаки.

Если описать текущее состояние информационной системы совокупностью пар атрибут-значение, а события представить как действия, связанные с изменением этих атрибутов, то для описания атаки может использоваться теория конечных автоматов. В этом случае реализации каждой атаки соответствует последовательность переходов из «исходного» состояния системы в ее «конечное» состояние, характеризующее реализацию данной атаки. Условия и направление перехода определяются набором правил, как было описано выше. Такой подход к описанию сценария атаки является более точным, чем описание при помощи набора правил, так как позволяет учитывать динамику развития атаки и выявлять попытки реализации атак, скрытых в интенсивном потоке событий, сгенерированных злоумышленником для прикрытия своих истинных намерений.

Применение методов сигнатурного анализа требует от разработчика СОА выбора или создания специального языка, позволяющего описывать регистрируемые системой события, а также устанавливать соответствия между ними. Универсальность и полнота этого языка являются определяющими факторами эффективной работы системы обнаружения, так как в конечном счете на этом языке будут сформулированы правила, по которым выявляется атака.

Реагирование на попытку реализации атаки может включать как простое извещение администратора информационной системы, так и более активные меры: разрыв установленного соединения, отключение уязвимой службы, перепрограммирование межсетевого экрана на отклонение пакетов, полученных от выявленного системой злоумышленника, а также другие меры, препятствующие «успешному» завершению атаки.

Все СОА, использующие метод обнаружения атак по сигнатуре, имеют в своем составе базу данных известных атак (их сигнатур). Очевидно, что к принципиальным недостаткам рассматриваемого класса СОА относится невозможность обнаружения атак, сигнатуры которых отсутствуют в базе данных. Поэтому, чтобы обеспечить эффективную работу системы обнаружения, эта база должна регулярно обновляться. Обычно возможность обновления, в том числе автоматического, предусмотрена разработчиками системы. Преимуществами систем, использующих сигнатурный анализ, являются низкая вероятность «ложной тревоги» (ошибочного обнаружения атаки при ее фактическом отсутствии), а также относительная простота настройки.

Подход к обнаружению атак, основанный на попытке обнаружения аномального поведения системы, также был впервые предложен в 1980-х годах. Основной предпосылкой применения указанного подхода является то, что

в процессе «штатного» функционирования информационная система находится в некотором равновесном состоянии. Попытка реализации атаки ведет к выходу системы из этого состояния, причем факт выхода может быть зафиксирован. При создании СОА, работающих по принципу обнаружения аномалий, должны быть решены три задачи. Во-первых, необходимо разработать способ описания состояния информационной системы. Это нетривиальная задача, так как должна быть учтена как статическая, так и динамическая составляющие. Например, должны быть описаны типичные для системы потоки данных, передаваемых по сети. Во-вторых, необходимо разработать алгоритмы, при помощи которых будет автоматически (или с вмешательством администратора) составляться описание реальной работающей системы — ее «профиль». Это нужно для того, чтобы «научить» СОА различать штатный режим работы информационной системы. В-третьих, необходимо выбрать математические методы, которые будут использоваться для обнаружения аномалий в процессе функционирования системы. Другими словами, должны быть определены механизмы принятия решения о попытке атаки защищаемой системы. Очевидно, что используемые механизмы принятия решения в первую очередь зависят от того, как была описана система.

Рассмотрим простой пример. Пусть одним из параметров информационной системы является количество отклоненных входящих TCP-соединений, а точнее — среднее значение и дисперсия указанного параметра. В случае штатной работы системы количество отклоняемых соединений должно быть незначительным. Если злоумышленник начнет исследовать уязвимость системы при помощи сканирования портов, то есть попытается реализовать атаку «сканирование портов», количество отклоненных TCP-соединений резко возрастет. Такой скачок может быть обнаружен различными способами. Во-первых, может быть применен статистический критерий равенства средних значений двух случайных величин. Для его использования, правда, необходимо сделать два достаточно неоднозначных предположения: о нормальности распределений случайных величин и о равенстве их дисперсий. Во-вторых, что представляется более уместным, могут быть применены математические методы, известные под общим названием «методов обнаружения разладки». Эти методы специально разрабатывались для решения подобного класса задач, первоначально связанных с контролем систем слежения и управления. В-третьих, могут применяться математические методы распознавания образов. Известны также разработки, использующие нейросетевые методы анализа, однако о практическом внедрении этих методов в коммерческих программных продуктах до сих пор не сообщается.

Основным преимуществом использования подхода, основанного на обнаружении аномального поведения системы, является теоретическая возможность обнаружения новых, не описанных ранее, атак. Данная возможность основана на предположении, что атака по определению представляет собой набор действий, нехарактерных для штатного режима работы системы. На-

сколько эффективно будут выявляться новые атаки, определяется опять же способом описания состояния системы и количеством анализируемых параметров. Большинство математических методов обнаружения, используемых в рассматриваемом классе СОА, не являются детерминированными. Это означает, что все решения принимаются на основе статистического анализа и, следовательно, могут содержать ошибки. Возможны два класса ошибок: «пропуск атаки» и «ложная тревога». Вероятность пропуска определяется характером атаки и степенью адекватности описания текущего состояния системы. «Ложная тревога» может иметь место в том случае, если в информационной системе наблюдается нетипичная активность, являющаяся следствием действий законных пользователей (или процессов). Например, если на всех компьютерах локальной сети, имеющей подключение к Интернет, будет установлена антивирусная программа, запрограммированная на обновление антивирусных баз в одно и то же время каждые два дня, то одновременная попытка всех компьютеров подключиться к одному серверу Интернет будет интерпретироваться СОА как вирусная атака. Поэтому именно высокую вероятность «ложной тревоги» обычно называют главным недостатком систем обнаружения атак, основанных на обнаружении аномальной активности. Еще одним недостатком принято считать сложность настройки («обучения») системы, так как этот процесс требует от администратора глубоких знаний базовых принципов взаимодействия элементов информационной системы. В связи с этим полноценных коммерческих продуктов, использующих принципы обнаружения аномальной активности, на рынке практически нет, хотя разработки этих систем непрерывно ведутся ввиду их перспективности.

4.2. Обнаружение в реальном времени и отложенный анализ

По типу обрабатываемых данных системы обнаружения атак подразделяются на «системы реального времени» и «системы отложенной обработки». Системы отложенной обработки анализируют содержимое журналов регистрации событий или массив предварительно записанного трафика, а системы реального времени — входящий поток событий от программных датчиков. Очевидно, что адекватное реагирование на попытку реализации атаки, включая ее предотвращение, возможно только при использовании систем реального времени. В то же время это не означает, что СОА реального времени «лучше», чем системы отложенной обработки. Так, СОА реального времени, не имеющая функций по предотвращению атак, заведомо менее эффективна, чем аналогичная система с отложенной обработкой, поскольку в системе реального времени одним из основных критериев эффективности является простота используемых алгоритмов, а не их оптимальность с позиций надежности обнаружения атак. Поэтому выбор того или иного типа СОА должен делаться исходя из анализа задач, которые ставятся перед системой обнаружения.

Апостериорный анализ может использоваться со следующей целью:

− расследование информационных преступлений и инцидентов;
− выявление атак, не являющихся информационным преступлением (сбор информации об инфраструктуре сети, сканирование портов и пр.);
− сбор информации об уязвимостях информационной системы с целью их устранения;
− анализ активности отдельных пользователей;
− минимизация системных требований к СОА.

Основной целью использования систем обнаружения атак реального времени является быстрое реагирование на попытки реализации атак, в том числе пресечение этих попыток. В связи с этим типичными процедурами для данных систем является анализ и фильтрация трафика на сетевом и транспортном уровнях модели OSI. Чтобы сократить производительные затраты, часто рассматриваются лишь заголовки пакетов, а их содержимое «отбрасывается». Это, очевидно, значительно сокращает перечень обнаруживаемых атак.

4.3. Локальные и сетевые системы обнаружения атак

СОА, использующие информацию, получаемую от персонального компьютера, на который они установлены, обычно называют *локальными* (*host-based*). В противоположность им системы обнаружения, ориентированные на анализ всего доступного сетевого трафика, называют *сетевыми* (*network-based*).

Рассмотрим, какая информация может использоваться локальными СОА для выявления попыток атаки.

− Отслеживание попыток входящих и исходящих TCP- и UDP-соединений. В результате могут обнаруживаться и пресекаться попытки несанкционированных подключений к отдельным портам, а также попытки сканирования портов.
− Анализ входящего и исходящего сетевого трафика на предмет наличия «подозрительных» пакетов. «Досмотру» могут подлежать как поля заголовков пакетов, так и их содержимое.
− Отслеживание попыток регистрации на локальной ЭВМ. В случае интерактивной регистрации может накладываться ограничение на время регистрации, а в случае регистрации по сети можно ограничить перечень сетевых адресов, с которых разрешается вход в систему. Отдельное внимание уделяется множественным неудачным попыткам регистрации, которые могут иметь место в случае атаки «подбор пароля методом перебора».
− Отслеживание активности пользователей, наделенных повышенными полномочиями в системе (суперпользователь root в UNIX-системах, пользователи группы Администраторы в ОС Windows). Так как в широком классе случаев атака направлена на получение полномочий суперпользователя, могут

отслеживаться попытки регистрации этого пользователя в системе в неразрешенное время, попытки сетевой регистрации суперпользователя и т. д.

− Проверка целостности отдельных файлов или ключей реестра (для Windows-систем). Несанкционированные действия взломщика могут заключаться в попытках внесения изменений в базу данных пользователей или в модификации отдельных настроек системы. Контроль целостности критичных областей данных позволит СОА обнаружить попытку реализации атаки.

Сетевые СОА собирают и анализируют все доступные им сетевые пакеты на предмет наличия «подозрительного» содержимого или несанкционированных потоков информации от одного узла сети к другому. В связи с этим точка подключения СОА должна обеспечивать максимальный охват трафика, циркулирующего в сегменте сети. Обычно такие системы подключаются к специальному порту коммутатора либо устанавливаются непосредственно на маршрутизаторе сети. СОА данного класса гораздо эффективнее, чем локальные, способны обнаруживать факт сканирования портов, а также выявлять попытки атак на «отказ в обслуживании». Кроме того, если система обнаружения установлена на шлюзе, обеспечивающем доступ из локальной сети в Интернет, то путем фильтрации нежелательных пакетов может обеспечиваться защита этой локальной сети от внешних атак. Получается, что СОА выполняет в этом случае функции межсетевого экрана (либо управляет им). Таким образом, сетевые системы обнаружения атак находят свое применение в информационных системах, где установка специализированного программного обеспечения на компьютеры пользователей затруднительна, и там, где требуется изолировать сетевой сегмент от внешней угрозы. Необходимо отметить, что анализ интенсивного потока данных требует существенных вычислительных затрат, поэтому аппаратные требования к узлу, на котором устанавливается такая СОА, могут быть очень высокими. Наиболее критичной эта проблема становится при попытке защиты сети, содержащей несколько сотен компьютеров и имеющей выход в Интернет. К этому классу относятся большинство сетей крупных предприятий.

4.4. Распределенные системы обнаружения атак

Отдельным классом систем обнаружения атак являются распределенные системы. Их основным отличием является перераспределение функций сбора данных между несколькими «агентами» — программными датчиками, установленными на узлах информационной системы. Агенты могут собирать информацию непосредственно с компьютера, на который они установлены, или анализировать данные, передаваемые по сети. Наиболее принципиальным моментом при внедрении распределенных СОА является организация информационного обмена между отдельными агентами системы. Конечной целью этого обмена является принятие решения о факте атаки.

Преимуществом использования распределенных систем является возможность собирать и анализировать значительно больший объем информации, что позволяет обнаруживать широкий спектр атак. В этом отношении наиболее эффективным является решение, объединяющее методологию локальных и сетевых СОА в единое целое. С другой стороны, распределенные системы обладают рядом недостатков, наиболее существенным из которых является меньшая защищенность их компонентов. Во-первых, сбор данных с нескольких узлов создает дополнительную нагрузку на сеть, которая, в случае полномасштабной атаки, может превысить ее пропускную способность. Во-вторых, обмен информацией по сети подразумевает наличие открытых портов, потенциально доступных для атаки на отказ в обслуживании (переполнение очереди входящих TCP-соединений). В-третьих, на узлах информационной системы возможно внедрение вредоносного программного обеспечения, которое будет блокировать работу агента или пытаться подделывать информацию, им передаваемую.

Отдельной задачей, возникающей при проектировании распределенных СОА, является выбор идеологии процедуры принятия решения. Возможны несколько вариантов процедуры, отличающиеся степенью централизации. Наиболее простым является вариант процедуры с предельной степенью централизации, когда агенты занимаются лишь сбором данных и передачей их центральному узлу СОА — модулю принятия решения. Этот модуль анализирует поступающую информацию и выносит решение о факте атаки. Данный вариант характеризуется большим объемом передаваемых по сети данных, что повышает вероятность обнаружения факта работы СОА злоумышленником и делает систему уязвимой к атакам на отказ в обслуживании. Наиболее очевидным решением по использованию такого типа СОА является их установка в рамках подсетей небольшого размера, что позволит обойти проблему перегрузки сети пакетами, сгенерированными системой. Увеличение масштабов сети требует многоступенчатого подхода к принятию решения. Он заключается в том, что выделяются промежуточные модули принятия решения, которые собирают данные только с ограниченного числа «своих» агентов и передают на верхний уровень гораздо меньший объем информации, дополненный промежуточным решением. При любом варианте реализации процедуры принятия решения очевидно, что первоочередной становится задача обеспечения защищенности самой СОА.

4.5. Система обнаружения атак Snort

4.5.1. Общие сведения

Snort — это бесплатная система обнаружения атак с открытым исходным кодом, разработанная Мартином Рошем (Martin Roesch). Доступны версии программы, работающие под управлением операционных систем Win-

dows NT, Linux, BSD, Mac OS X, а также некоторых других. В соответствии с предложенной выше классификацией, Snort является сетевой СОА, основанной на сигнатурном анализе. Сигнатуры атак описываются при помощи *правил* — специальных синтаксических конструкций, позволяющих выявлять интересующую администратора информацию в полях заголовков и содержимом передаваемых по сети пакетов. Кроме того, в Snort реализовано несколько препроцессоров, выполняющих более сложные операции по анализу трафика, такие, например, как дефрагментация IP-пакетов, отслеживание TCP-соединений и выявление попыток сканирования портов.

4.5.2. Установка и запуск программы

Для обеспечения возможности перехвата сетевых пакетов программой Snort необходима предварительная установка и запуск службы WinPcap (WinPcap_3_1.exe).

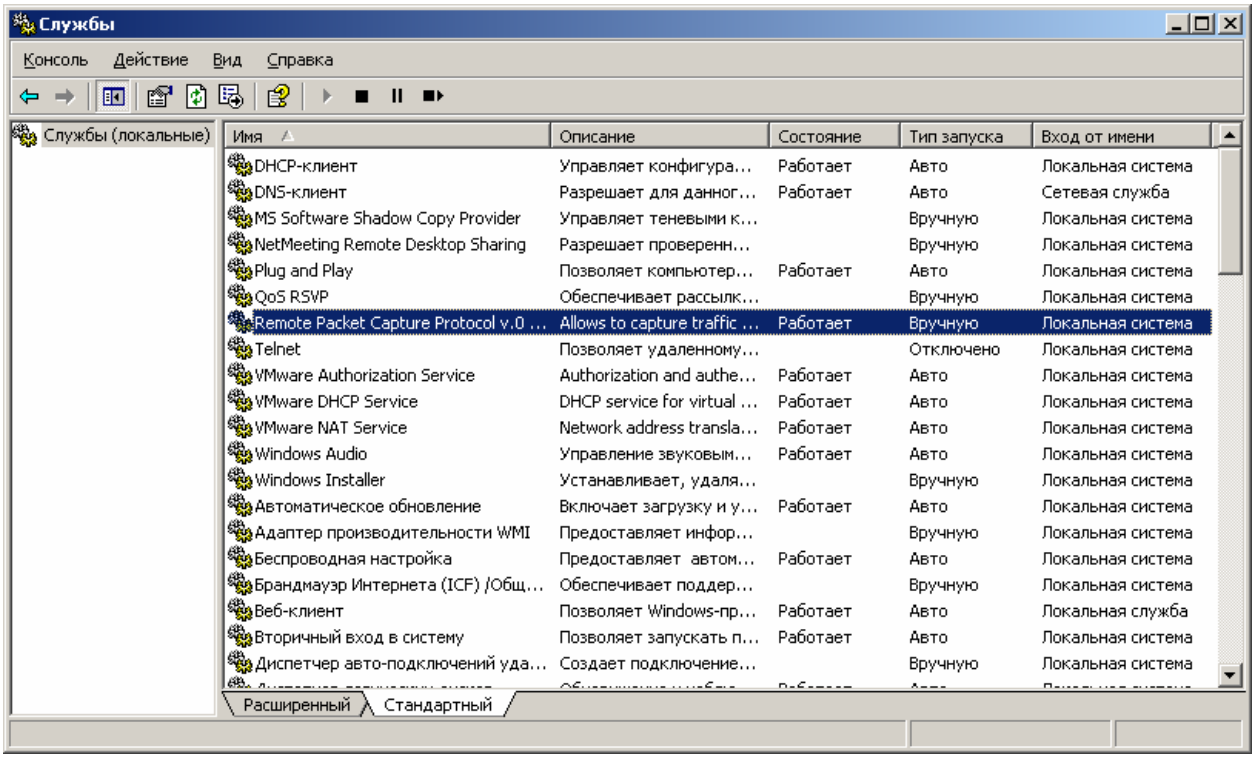


Рис. 4.1. Служба WinPcap в перечне служб

Для установки СОА Snort версии 2.4.3 необходимо запустить файл «Snort_243_Installer.exe» и ответить на интересующие программу-инсталлятор вопросы. По умолчанию Snort устанавливается в каталог «C:\snort», а его исполняемый файл располагается в каталоге «C:\snort\bin». Запуск СОА Snort осуществляется из командной строки, в табл. 4.1 приведен неполный список параметров, с которыми может производиться запуск. Чтобы вывести этот список на экран во время работы, необходимо выполнить команду

```
snort -?
```

Таблица 4.1

Список параметров СОА Snort

Параметр	**Описание**
`-c <rules>`	Использовать файл правил `<rules>`.
`-E`	Добавлять предупреждения (alerts) в журнал регистрации событий Windows NT (не создавая log-файла).
`-h <hn>`	Задать домашнюю сеть (home network) `<hn>`.
`-i <if>`	Подключиться к сетевому интерфейсу номер `<if>` (список интерфейсов можно получить при помощи команды `snort -W`).
`-I`	Добавлять к предупреждению наименование интерфейса.
`-K <mode>`	Режим регистрации предупреждений: `pcap` (по умолчанию) — в двоичном формате, `ascii` — в текстовом формате, `none` — без регистрации.
`-l <ld>`	Сохранять результаты регистрации в каталог `<ld>`.
`-L <file>`	Сохранять результаты регистрации в указанный файл формата tcpdump (будет располагаться в каталоге, предварительно указанном параметром `-l`).
`-n <cnt>`	Завершить работу программы после получения `<cnt>` пакетов.
`-N`	Не сохранять в файлах регистрации содержимого пакетов (сохраняется лишь текст предупреждений).
`-p`	Анализировать только пакеты, отправленные на локальный адрес, и широковещательные пакеты.
`-r <tf>`	Прочитать и обработать файл `<tf>`, записанный в формате tcpdump.
`-S <n=v>`	Установить значение переменной `n` файла правил, равной `v`.
`-U`	Записывать временные метки в универсальном скоординированном времени (UTC).
`-v`	Отображать на экране заголовки всех перехваченных пакетов.
`-V`	Показать версию Snort.
`-W`	Показать доступные сетевые интерфейсы.
`-X`	Сохранять в файле журнала регистрации событий содержимое перехваченных пакетов в «сыром» виде, начиная с уровня link модели OSI.

Параметр	Описание
`-y`	Добавить месяц, день и год в отображаемые и сохраняемые временные отметки.
`-?`	Показать помощь по параметрам командной строки.

Для проверки работоспособности СОА Snort рекомендуется выполнить следующие действия.

ВЫПОЛНИТЬ!

1. Установить СОА Snort.
2. Вывести на экран список доступных сетевых интерфейсов командой

   ```
   snort –W
   ```

3. Запустить Snort на выбранном интерфейсе в режиме анализатора пакетов с выводом информации на экран, указав программе завершить работу после приема третьего пакета:

   ```
   snort –v –i 1 –n 3
   ```

4. Выполнить любые действия, которые приведут к отправке или приему сетевых пакетов (например, отправить эхо-запрос на любой IP-адрес командой ping). Убедиться, что пакеты перехватываются и отображаются на экране.

4.5.3. Описание языка правил

Рассмотрим краткое описание языка правил, на котором задаются сигнатуры атак, обнаруживаемых СОА Snort. Полное описание языка правил содержится в файле документации «c:\snort\doc\snort_manual.pdf»

Правила записываются в одну строку, если возникает необходимость перенести текст правила на следующую строку, необходимо добавить в конце строки символ обратной косой черты «\».

Правила состоят из двух частей: заголовка и набора атрибутов. Заголовок, в свою очередь, состоит из:

1. Указания действия, которое необходимо выполнить (`alert`, `log`, `pass` и др.).
2. Протокола (`tcp`, `udp`, `icmp`, `ip`).
3. IP-адреса и маски подсети источника и приемника информации, а также информации о портах источника и приемника.

Действие `alert` заключается в генерации предупреждающего события и сохранении содержимого пакета для дальнейшего анализа. Действие `log` предполагает сохранение пакета без генерации предупреждения. Действие

pass означает пропуск пакета (его игнорирование). Существует также ряд более сложных действий, которые здесь не рассматриваются.

Текст атрибутов располагается в скобках, каждая пара атрибут — значение имеет вид *<атрибут>: <значение>;*. Значения строковых атрибутов записываются в кавычках.

Рассмотрим пример простого правила (одна строка):

```
alert <протокол> <адрес_подсети1>[/маска_подсети1]
<порт1> <направление> <адрес_подсети2>[/маска_подсети2]
<порт2> ([msg:"Текст сообщения";] [другие_атрибуты])
```

где

- alert — действие, которое необходимо выполнить при обнаружении пакета, удовлетворяющего данному правилу, и которое заключается в генерации «предупреждения» — записи в журнале регистрации
- <протокол> — наименование протокола (tcp, udp, icmp, ip)
- <адрес_подсети>[/маска_подсети] — IP-адрес и маска подсети, либо IP-адрес узла участника обмена в формате: 192.168.247.0/24, либо 192.168.247.1
- <порт> — номер порта либо диапазон портов в формате 1:1024 для обозначения диапазона портов от 1 до 1024, 1024: — с номерами больше или равными 1024, или :1024 — меньше или равными 1024 соответственно
- <направление> — обозначение направления в виде ->, <- или <>

Вместо IP-адресов и номеров портов могут использоваться псевдонимы any, являющиеся заменителем любого значения.

Атрибуты являются наиболее значимой частью правил, так как позволяют искать интересующую информацию в полях заголовков и содержимом пакетов. Существует четыре категории атрибутов:

- meta-data — предоставляют информацию о правиле, но не влияют на процесс обнаружения;
- payload — атрибуты данного типа предназначены для поиска информации в «полезной нагрузке» (содержимом) пакета;
- non-payload — предназначены для поиска информации в заголовках пакетов;
- post-detection — определяют поведение системы после обнаружения пакета, удовлетворяющего правилу.

В табл. 4.2 приведен неполный список атрибутов, которые могут быть использованы при написании правил.

Если известно местонахождение интересующей информации в пакете, то целесообразно ограничить область поиска при помощи модификаторов offset и depth, так как это существенно сократит время, затрачиваемое на анализ пакета.

Таблица 4.2

Список атрибутов COA Snort

Атрибут	Описание
meta-data	
`msg: "<текст>";`	Сообщение, которое добавляется в журнал регистрации при активации правила.
`sid: <идентификатор>;`	Уникальный номер, используемый для идентификации правил. Идентификаторы от 100 до 1 000 000 используются для правил, включенных в дистрибутив Snort. Для локальных правил следует использовать значения больше 1 000 000.
`rev: <номер_редакции>;`	Целое число, служащее для обозначения номера редакции правила.
`classtype: <имя_класса>;`	Используется для обозначения класса атаки. Полный список классов приведен в документации по Snort.
`priority: <приоритет>;`	Целое число, используемое для переопределения приоритета, задаваемого указанным ранее классом атаки, или для назначения приоритета новому правилу. Наивысший приоритет — 1, типичное значение атрибута составляет от 1 до 4.
payload	
`content: [!]"<строка>";`	Позволяет искать заданную подстроку в содержимом полезной нагрузки пакета. Восклицательный знак означает отсутствие указанной информации в пакете. По умолчанию данный атрибут является чувствительным к регистру. Для обозначения двоичных данных следует использовать шестнадцатеричные значения, отделенные вертикальными чертами: `|00 5C|`. Атрибут `content` имеет несколько модификаторов, которые могут располагаться следом за ним.
`nocase;`	Модифицирует стоящий ранее атрибут `content`, делая его нечувствительным к регистру.
`depth: <число_байт>;`	Значение атрибута (в байтах) определяет, как далеко в полезной нагрузке пакета должен производиться поиск.

Атрибут	Описание
`offset: <число_байт>;`	Значение атрибута определяет, сколько байтов полезной нагрузки следует пропустить. Поиск будет вестись, начиная с `число_байт+1`-го байта.
`distance: <число_байт>;`	Атрибут похож на `depth`, но указывает, сколько байт необходимо пропустить после предыдущей совпавшей подстроки перед продолжением поиска.
`within: <число_байт>;`	Атрибут указывает системе искать совпадения лишь в первых `число_байт`, начиная с конца предыдущей совпавшей подстроки.
non-payload	
`dsize: <размер>;`	Сравнить размер полезной нагрузки с заданным. Возможно указание диапазонов значений с использованием знаков >, < и <>. Например: `>128`, `300<>500` (от 300 до 500).
`flags: [!|*|+]<флаги>`	Проверить, установлены ли указанные флаги в принятом TCP-пакете. Флаги записываются подряд без пробелов и обозначаются следующим образом: `F` — FIN (LSB в байте флагов) `S` — SYN `R` — RST `P` — PSH `A` — ACK `U` — URG `1` — Резерв 1 (MSB в байте флагов) `2` — Резерв 2 `0` — флаги не установлены Могут быть дополнительно использованы следующие модификаторы: ! — указанные флаги не установлены * — установлен хотя бы один из указанных + — установлены указанные и любые другие
`itype: <тип>;`	Сравнить тип ICMP сообщения с указанным. Возможно указание диапазонов значений с использованием знаков >, < и <> (см. выше).
`icode: <код>;`	Сравнить код ICMP сообщения с указанным.

Вместо IP-адресов могут использоваться переменные, заданные выше по тексту следующим образом:

```
var <имя_переменной> <значение_переменной>
```

Чтобы сослаться на переменную далее в тексте, перед ее именем следует поставить знак доллара `$`.

В текст файла правил можно включать комментарии, которые отделяются знаком `#`. Вся информация справа от этого знака и до конца строки считается комментарием и не интерпретируется системой:

```
#<комментарий>
```

Рассмотрим пример задания двух переменных последующего их использования в правиле, фильтрующем входящие ICMP-пакеты ECHO (тип 8):

```
# Глобальные переменные
var HOME_NET 192.168.247.1
var EXTERNAL_NET !$HOME_NET

# Обнаружение эхо-запросов (ping'ов)
alert icmp $EXTERNAL_NET any -> $HOME_NET any
(msg:"Incoming ECHO REQUEST"; itype: 8;)
```

4.5.4. Использование СОА Snort

СОА Snort можно использовать как анализатор трафика, обладающий значительными возможностями по фильтрации пакетов. Например, можно создать файл с правилами, использующими исключительно действия типа `log`. В результате из входящего потока данных будут отобраны и сохранены пакеты, удовлетворяющие указанным правилам. Так как по умолчанию журнал ведется в двоичном формате tcpdump, он может быть импортирован почти всеми специализированными программами анализа трафика. Обычно эти программы позволяют наглядно отображать содержимое пакетов, но не обладают такими возможностями по их фильтрации, как Snort.

Для запуска Snort в режиме анализатора трафика, как и для запуска его в режиме системы обнаружения атак, необходимо выполнить следующую команду в командной строке Windows:

```
snort -i <интерфейс> -c <файл_конфигурации>
-l <путь_к_журналу>
```

где

`<интерфейс>` — номер интерфейса, полученный в результате выполнения команды `snort –W`

<файл_конфигурации> — путь к файлу, в котором хранятся настройки программы и правила обнаружения

<путь_к_журналу> — путь к каталогу, в котором необходимо сохранить файл журнала

Пример:

```
snort -i 3 -c ../etc/my.conf -l ../log
```

Следует обратить внимание, что при записи пути используются не обратные, а прямые «косые черты».

Для завершения работы СОА Snort, необходимо нажать клавиши <Ctrl+C>.

Рассмотрим несколько правил (табл. 4.3), которые позволят обнаруживать атаки, описанные в разделе 2. Текст правил должен записываться в одну строку.

Таблица 4.3

Примеры правил СОА Snort

№	Описание	Правило
1	Обнаружение входящих ECHO-запросов (ping'ов)	`alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg: "Incoming ECHO REQUEST"; itype: 8;)`
2	Обнаружение исходящих ECHO-ответов	`alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg: "Outgoing ECHO REPLY"; itype: 0;)`
3	Обнаружение больших ICMP-пакетов (атака «Ping of Death»)	`alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg: "Incoming large ICMP packet"; dsize: >800;)`
4	DoS-атака Winnuke	`alert tcp $EXTERNAL_NET any -> $HOME_NET 135:139 (msg: "DoS Winnuke attack"; flags: U+;)`

5	Запрос на подключение к 139 порту (служба SMB) из внешней сети (два варианта)	`alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg: "NETBIOS SMB IPC$ share access"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "\\IPC$|00|"; nocase;)`
		
		`alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS SMB IPC$ share access (unicode)"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content:`

№	Описание	Правило
		"|5c00|I|00|P|00|C|00|$| 00|"; nocase;)
6	Запрос на подключение к 445 порту (служба SMB) из внешней сети (два варианта)	alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg: "NETBIOS SMB IPC$ share access"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "\\IPC$|00|"; nocase;) alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"NETBIOS SMB IPC$ share access (unicode)"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "|5c00|I|00|P|00|C|00|$|00|"; nocase;)
6	Обнаружение сканирования портов методом NULL.	alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"NULL port scanning"; flags:!FSRPAU;)
7	Обнаружение сканирования портов методом XMAS.	alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"XMAS port scanning"; flags:FPU+;)

ВЫПОЛНИТЬ!

5. Создать в каталоге «\snort\etc» файл «my.conf», содержащий следующие строки:

   ```
   var HOME_NET <IP-адрес_СОА>
   var EXTERNAL_NET !$HOME_NET
   ```

6. Добавить в файл «my.conf» правило, позволяющее обнаруживать входящие ECHO-запросы. Проверить, происходит ли обнаружение, запустив СОА из каталога «\snort\bin» следующей командой (из «командной строки»):

   ```
   snort -i <интерфейс> -c ../etc/my.conf -l ../log
   ```

   Для проверки выполнить несколько ECHO-запросов с другого компьютера, используя команду:

   ```
   ping <IP-адрес_СОА>
   ```

7. Дополнить файл «my.conf» правилами, указанными в табл. 4.3. Для проверки обнаружения подключений к службе SMB использовать команду:

   ```
   net use \\<IP-адрес_СОА>\IPC$ "" /user:""
   ```

К какому порту производится подключение? Как это зависит от используемой операционной системы?

4.5.5. Выявление факта сканирования портов

В СОА Snort встроен программный модуль, позволяющий выявлять сканирование портов защищаемой системы. Алгоритм, обнаруживающий сканирование, основан на том, что при сканировании портов существенно увеличивается количество исходящих TCP-пакетов с установленным флагом RST. Установка этого флага на отправляемом в ответ пакете означает, что порт, к которому производилось обращение, закрыт. Таким образом, анализируя количество пакетов с установленным флагом RST, можно обнаружить факт сканирования портов системы.

Программный модуль, или препроцессор, как его называют разработчики Snort, инициализируется из файла конфигурации следующим образом. В конфигурационный файл следует добавить следующие строки:

```
preprocessor flow: stats_interval 0 hash 2
preprocessor sfportscan: proto { <протокол> } scan_type
{ <тип_сканирования> } sense_level
{ <чувствительность> } logfile { <файл_с_отчетом> }
```

Первая строка предназначена для инициализации препроцессора Flow, без которого модуль обнаружения сканирования не работает. Вторая строка инициализирует препроцессор sfPortscan, при этом задаются следующие параметры (жирным шрифтом показаны рекомендуемые значения):

`<протокол>` — анализируемый протокол (`tcp`, `udp`, `icmp`, `ip_proto`, **`all`**)

`<тип_сканирования>` — выявляемые типы сканирования (`portscan`, `portsweep`, `decoy_portscan`, `distributed_portscan`, **`all`**)

`<чувствительность>` — чувствительность (`low`, **`medium`**, `high`)

`<файл_с_отчетом>` — имя файла, в который будет помещен отчет об обнаруженных попытках сканирования портов

Файл с отчетом будет располагаться в каталоге, указанном параметром `-c` при запуске Snort. Чувствительность определяет перечень анализируемой информации и в итоге сказывается на вероятности «ложной тревоги» (для high она наибольшая). За более подробной информацией о параметрах модуля sfPortscan следует обращаться к документации на Snort.

Приведем пример настройки модуля sfPortscan:

```
preprocessor flow: stats_interval 0 hash 2
preprocessor sfportscan: proto { all } scan_type { all }
sense_level { medium } logfile { portscan.log }
```

При данной настройке Snort будет выявлять все описанные в разделе 2 методы сканирования портов. Необходимо отметить, что две и больше процедуры сканирования портов, выполненные во время одного сеанса работы Snort, будут отражены в файле регистрации событий только один раз. Это остается справедливым даже в том случае, когда используются разные методы сканирования. Таким образом, для поверки возможности обнаружения сканирования, выполняемого разными методами, Snort необходимо закрывать и запускать снова.

ВЫПОЛНИТЬ!

8. Дополнить файл «my.conf» приведенными выше строками для настройки препроцессора sfPortscan. С использованием утилиты nmap проверить, происходит ли обнаружение попыток сканирования портов защищаемого узла. Использовать следующие команды для запуска сканирования:
   `nmap <IP-адрес_СОА> -v -sT -p <диапазон_портов>` — для сканирования методом с полным циклом подключения (метод Connect)
   `nmap <IP-адрес_СОА> -v -sS -p <диапазон_портов>` — для сканирования с неполным циклом подключения (метод SYN)
   `nmap <IP-адрес_СОА> -v -sN -p <диапазон_портов>` — для сканирования при помощи TCP-пакета со сброшенными флагами (метод NULL)
   `nmap <IP-адрес_СОА> -v -sX -p <диапазон_портов>` — для сканирования при помощи TCP-пакета со всеми установленными флагами (метод XMAS)
9. Какие методы сканирования позволяют практически выявлять наличие открытых портов? Как это зависит от используемой операционной системы? Все ли указанные методы сканирования обнаруживает СОА Snort?

5. ОРГАНИЗАЦИЯ ВИРТУАЛЬНЫХ ЧАСТНЫХ СЕТЕЙ

5.1. Задачи, решаемые VPN

Защищенные компьютерные сети на сегодняшний день применяют технологию защиты информации, включающую в себя как элементы межсетевого экранирования, так и механизмы криптографической защиты сетевого трафика. Такая технология получила название VPN — Virtual Private Network (виртуальная частная сеть). В литературе (см. [2]) встречаются различные определения виртуальной частной сети. Мы будем использовать следующее. VPN — это технология, объединяющая доверенные сети, узлы и пользователей через открытые сети, к которым нет доверия. Основная идея данного определения приведена на схеме (рис. 5.1).

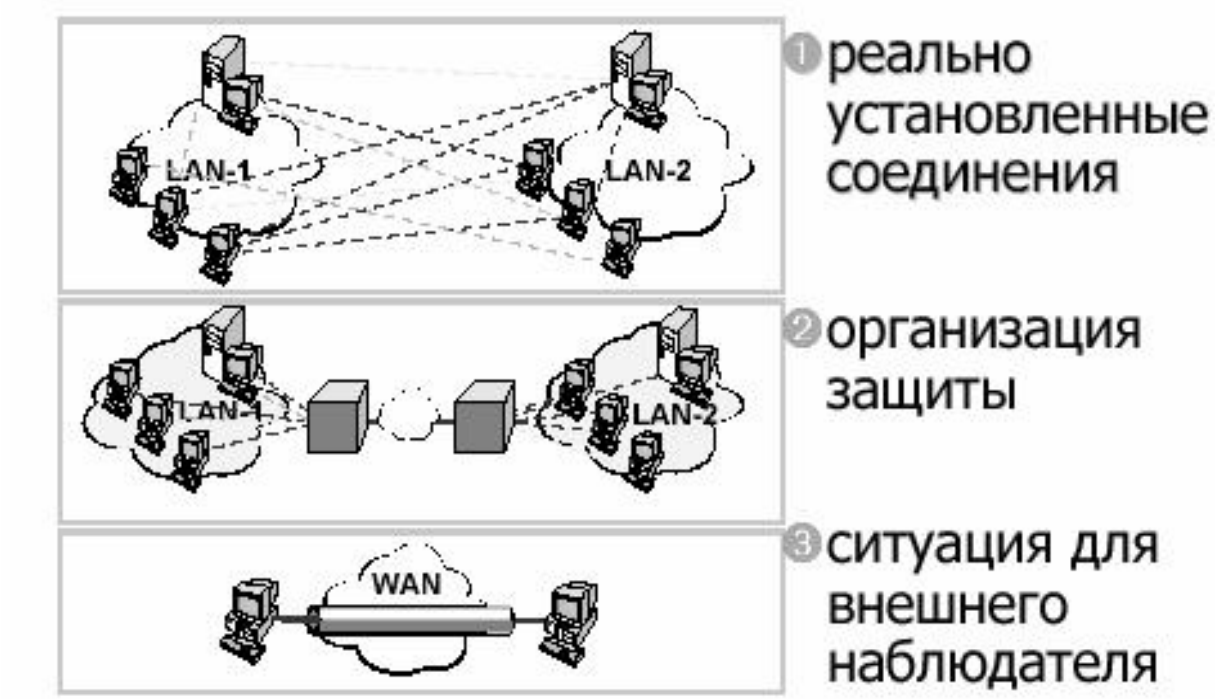


Рис. 5.1. Схема VPN

Предположим, имеются две локальных сети (LAN-1 и LAN-2, рис. 5.1), принадлежащие одной организации (например, головной офис и филиал). Обе эти локальные сети объединены при помощи иной сети, в большинстве случаев для этого используется Интернет. С точки зрения пользователей соединения могут устанавливаться между любыми узлами этих локальных сетей. На самом же деле реальные соединения устанавливаются через посредников, неких «черных ящиков», устанавливаемых на входе в каждую из них. Задача этих «черных ящиков» так обработать идущий между ними сетевой трафик, чтобы злоумышленник или просто внешний наблюдатель не мог совершить с

передаваемой информацией какого-либо действия, приводящего к ущербу. А именно, не должен нарушить конфиденциальность, целостность и подлинность информации. Иными словами, передаваемая информация, включая адреса ее получателя и отправителя, должна быть зашифрована и криптографически подписана. Кроме того, задача «черных ящиков» — защищать сами локальные сети от несанкционированного доступа к ним из глобальной сети. Таким образом, внешний наблюдатель должен увидеть в сети лишь зашифрованный обмен информацией между двумя «черными ящиками» и ничего более.

Таким образом, можно сформулировать, что VPN призвана решать следующие задачи:

− обеспечивать защиту (конфиденциальность, целостность, подлинность) передаваемой по сетям информации[1]. Как указывалось выше, данная задача решается применением криптографического метода защиты передаваемой информации;

− выполнять защиту внутренних сегментов сети от НСД извне. Решение задачи возможно благодаря встроенным в VPN-системы функциям межсетевого экранирования, а также криптографическим механизмам, запрещающим незашифрованный сетевой трафик;

− обеспечивать идентификацию и аутентификацию пользователей. Данная задача возникает вследствие того, что, как сказано в определении VPN, в сети должны взаимодействовать лишь доверенные узлы, доверие к которым возможно после прохождения процедур идентификации и аутентификации.

Отдельно стоящей задачей, решаемой VPN, является экономия финансовых ресурсов организации, когда для обеспечения защищенной связи с филиалами применяются не защищенные выделенные каналы связи, а Интернет.

Сформулируем ряд требований, которые предъявляются к программно-аппаратным комплексам, реализующим VPN:

− масштабируемость, т. е. возможность со временем подключать новые локальные сети без необходимости изменения структуры имеющейся VPN;

− интегрируемость, т. е. возможность внедрения VPN-системы в имеющуюся технологию обмена информацией;

− легальность и стойкость используемых крипоалгоритмов, т. е. система должна иметь соответствующий сертификат, позволяющий ее использовать на территории Российской Федерации с целью защиты информации ограниченного доступа;

− пропускная способность сети, т. е. система не должна существенно увеличивать объем передаваемого трафика, а также уменьшать скорость его передачи;

---

[1] Заметим, что классическую задачу защиты информации в виде обеспечения ее доступности технология VPN самостоятельно решать не может.

− унифицируемость, т. е. возможность устанавливать защищенные соединения с коллегами по бизнесу, у которых уже установлена иная VPN-система;

− общая совокупная стоимость, т. е. затраты на приобретение, развертывание и обслуживание системы не должны превосходить стоимость самой информации, особенно если речь идет о защите коммерческой тайны.

5.2. Туннелирование в VPN

Как указывалось выше, основная задача, решаемая VPN, — скрыть передаваемый трафик. При этом необходимо скрыть как передаваемые данные, так и адреса реальных отправителя и получателя пакетов. И кроме того, необходимо обеспечить целостность и подлинность передаваемых данных. Для защиты передаваемых данных и реальных IP-адресов применяются криптографические алгоритмы. При отправке пакетов применяется туннелирование, т. е. в пакетах, которые идут в открытой сети, в качестве адресов фигурируют только адреса «черных ящиков». Кроме того, туннелирование предполагает, что внутри локальных сетей трафик передается в открытом виде, а его защита осуществляется только тогда, когда он попадает в «туннель».

Итак, пусть у нас имеется пакет, содержащий данные и IP-заголовок, которые подлежат защите (рис. 5.2). Для защиты применим криптографические методы и зашифруем и данные, и заголовок вместе. Так как необходимо обеспечить скорость обработки информации, то для зашифрования, естественно, будем использовать симметричный алгоритм.

Известно, что применение симметричных алгоритмов шифрования требует решения задачи распространения симметричных ключей. Поэтому поступим следующим образом: прикрепим симметричный ключ прямо к зашифрованным с его использованием данным. Назовем симметричный ключ пакетным ключом (его еще называют сеансовым ключом). Этот пакетный ключ будем генерировать случайным образом при отправлении каждого нового пакета (тогда он действительно «пакетный» ключ). Либо будем его генерировать также случайно при каждом сеансе обмена. Тогда данные всех пакетов, передаваемых в данном сеансе связи, будут шифроваться одним и тем же ключом, и это уже «сеансовый» ключ.

Конечно, нельзя отправлять пакетный ключ в открытом виде, прикрепляя его к зашифрованным им данным. Следует его зашифровать. Воспользуемся тем, что ключ, в отличие от данных, — это лишь пара сотен бит (в зависимости от реализации, например, 256 бит — длина ключа алгоритма ГОСТ 28147-89, 56 бит — длина ключа алгоритма DES). Таким образом, можем применить более медленные асимметричные алгоритмы и зашифровать с их помощью пакетный ключ. Вместе с тем, для шифрования пакетного ключа может быть применен и симметричный алгоритм. Ключ алгоритма шифрования пакетного ключа назовем ключом связи.

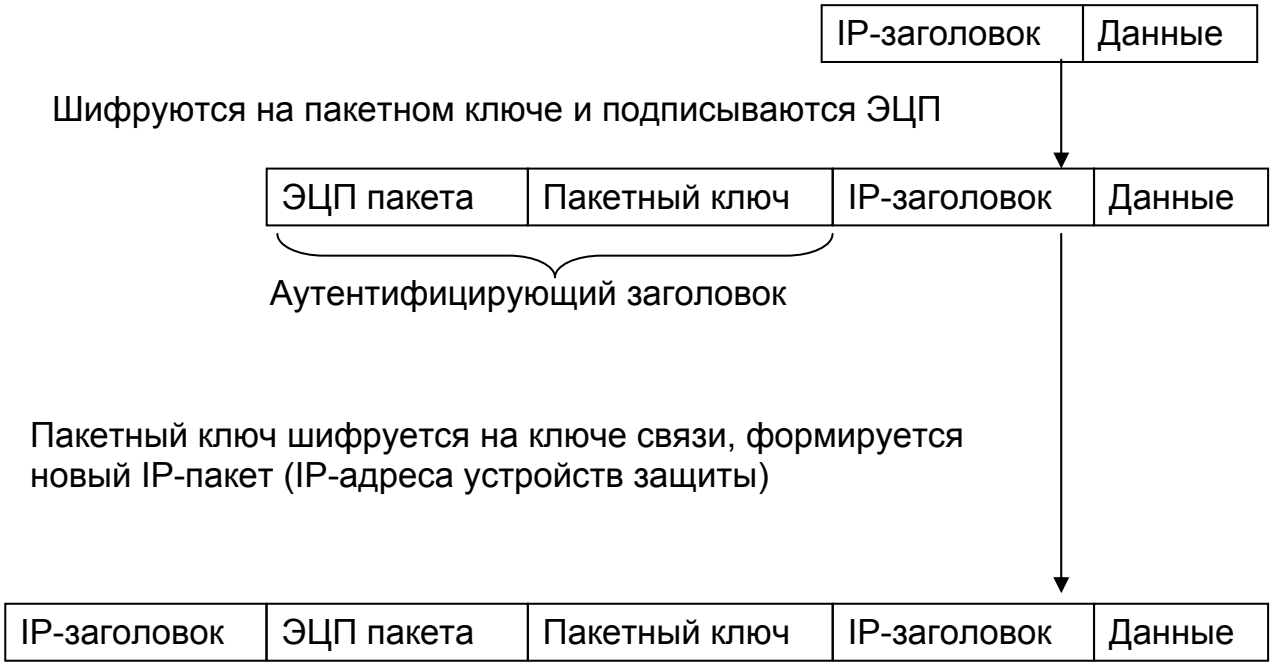


Рис. 5.2. Преобразование отправляемого пакета

Кроме того, для обеспечения целостности пакетов сгенерируем электронно-цифровую подпись (ЭЦП) нашего пакета и прикрепим ее к формируемому пакету.

Совокупность ЭЦП и зашифрованного пакетного ключа называют аутентифицирующим заголовком.

Для того чтобы отправить сгенерированный нами пакет, необходимо добавить к нему IP-адреса источника и приемника. В случае туннеля этими адресами будут адреса пограничных VPN-узлов. Если же защищается трафик между двумя узлами без применения туннеля, то эти адреса совпадут с адресами в исходном пакете.

Таким образом, исходный пакет защищен. Осталось выяснить ряд моментов. Во-первых, каким образом будет осуществлен обмен ключом связи и, во-вторых, что будем понимать под шифруемыми данными: только лишь данные прикладного уровня либо относящиеся к транспортному или сетевому уровню.

Чтобы ответить на второй вопрос, рассмотрим уровни защищенных каналов.

5.3. Уровни защищенных каналов

Итак, необходимо разобраться, данные какого уровня модели OSI подлежат шифрованию в процессе организации VPN.

Рассмотрим упрощенную модель OSI, реализованную в стеке протоколов TCP/IP. Эта модель предполагает наличие четырех уровней: прикладного, транспортного, сетевого и канального. Соответственно, для каждого уровня возможность шифрования передаваемой информации различна. Так, на при-

кладном уровне можно скрыть данные, например, электронного письма или получаемой web-страницы. Однако факт передачи письма, т. е. диалог по протоколу SMTP скрыть невозможно. На транспортном уровне может быть вместе с данными скрыт и тип передаваемой информации, однако IP-адреса получателя и приемника остаются открытыми. На сетевом уровне уже появляется возможность скрыть и IP-адреса. Эта же возможность имеется и на канальном уровне.

Чем ниже уровень, тем легче сделать систему, функционирование которой будет незаметно для приложений высокого уровня, и тем большую часть передаваемой информации можно скрыть.

Для каждого уровня модели разработаны свои протоколы (табл. 5.1).

Таблица 5.1

Уровни защищенных каналов и протоколы

Уровень	**Протоколы**
Прикладной	S/MIME / PGP / SHTTP
Транспортный (TCP/UDP)	SSL / TLS / SOCKS
Сетевой (IP)	IPSec / SKIP
Канальный	PPTP / L2F / L2TP

Так, на прикладном уровне для защиты электронной почты применяется протокол S/MIME (Secure Multipurpose Internet Mail Extension) либо система PGP. Для защиты обмена по протоколу HTTP применяется протокол SHTTP (Secure HTTP). На данном уровне шифруется текст передаваемого почтового сообщения или содержимое HTML-документа. Недостатками организации VPN на базе протоколов прикладного уровня является узкая область действия, для каждой сетевой службы должна быть своя система, способная интегрироваться в соответствующие приложения. В пособии мы не будем подробно рассматривать системы этого уровня.

На транспортном уровне чаще всего применяются протоколы SSL (Secure Socket Layer) и его более новая реализация — TSL (Transport Layer Security). Также применяется протокол SOCKS. Особенность протоколов транспортного уровня — независимость от прикладного уровня, хотя чаще всего шифрование осуществляется для передачи по протоколу HTTP (режим HTTPS). Недостатком является невозможность шифрования IP-адресов и туннелирования IP-пакетов.

На сетевом уровне используются два основных протокола: SKIP (Simple Key management for Internet Protocol – простое управление ключами для IP-протокола) и IPSec. На данном уровне возможно как шифрование всего трафика, так и туннелирование, включающее скрытие IP-адресов. На сетевом уровне строятся самые распространенные VPN системы.

Канальный уровень представлен протоколами PPTP (Point-to-Point Tunneling Protocol), L2F (Layer-2 Forwarding) и L2TP (Layer-2 Tunneling Protocol). Достоинством данного уровня является прозрачность не только для приложений прикладного уровня, но и для служб сетевого и транспортного уровня. В частности, достоинством является независимость от применяемых протоколов сетевого и транспортного уровня — это может быть не только IP-протокол, но и протоколы IPX (применяется в локальных сетях с серверами на основе ОС Novell Netware) и NetBEUI (применяется в локальных сетях Microsoft). Шифрованию подлежат как передаваемые данные, так и IP-адреса.

В каждом из указанных протоколов по-разному реализованы алгоритмы аутентификации и обмена ключами шифрования.

5.4. Защита данных на канальном уровне

На канальном уровне применяются упомянутые выше протоколы PPTP (разработчик Microsoft), L2F (разработчик Cisco Systems) и L2TP (совместная разработка Microsoft и Cisco Systems).

Протоколы PPTP и L2TP основываются на протоколе Point-to-Point Protocol (PPP). PPP — протокол канального уровня, разработан для инкапсуляции данных и их доставки по соединениям типа точка-точка.

В основе протокола PPTP лежит следующий алгоритм: сначала производится инкапсуляция данных с помощью протокола PPP, затем протокол PPTP выполняет шифрование данных и инкапсуляцию. PPTP инкапсулирует PPP-кадр в пакет Generic Routing Encapsulation (протокол GRE). Схема инкапсуляции приведена на рис. 5.3.

IP заголовок	GRE заголовок	PPP заголовок	**IP заголовок**	TCP, UDP	Данные

Рис. 5.3. Инкапсуляция в протоколе PPTP

К исходному отправляемому IP-пакету (обозначенному на рисунке серым цветом) последовательно добавляются PPP-, GRE- и IP-заголовки. В новом IP-пакете в качестве адресов указываются адреса туннелирующих узлов.

Протокол PPTP очень часто используется провайдерами Интернет при организации прямого кабельного подключения пользователей. В этом случае пользователям назначается IP-адрес из диапазона «домашних» сетей (например, 10.1.1.189 или 192.168.1.1). Сервер провайдера имеет два адреса — внутренний (для «домашней» сети) и внешний («настоящий»). Когда пользователь авторизуется на PPTP-сервере провайдера, ему динамически выделяется реальный IP-адрес.

Внутри локальной сети между пользователем и PPTP-сервером циркулируют IP-пакеты с внутренними IP-адресами, внутри которых инкапсулированы пакеты с внешними адресами.

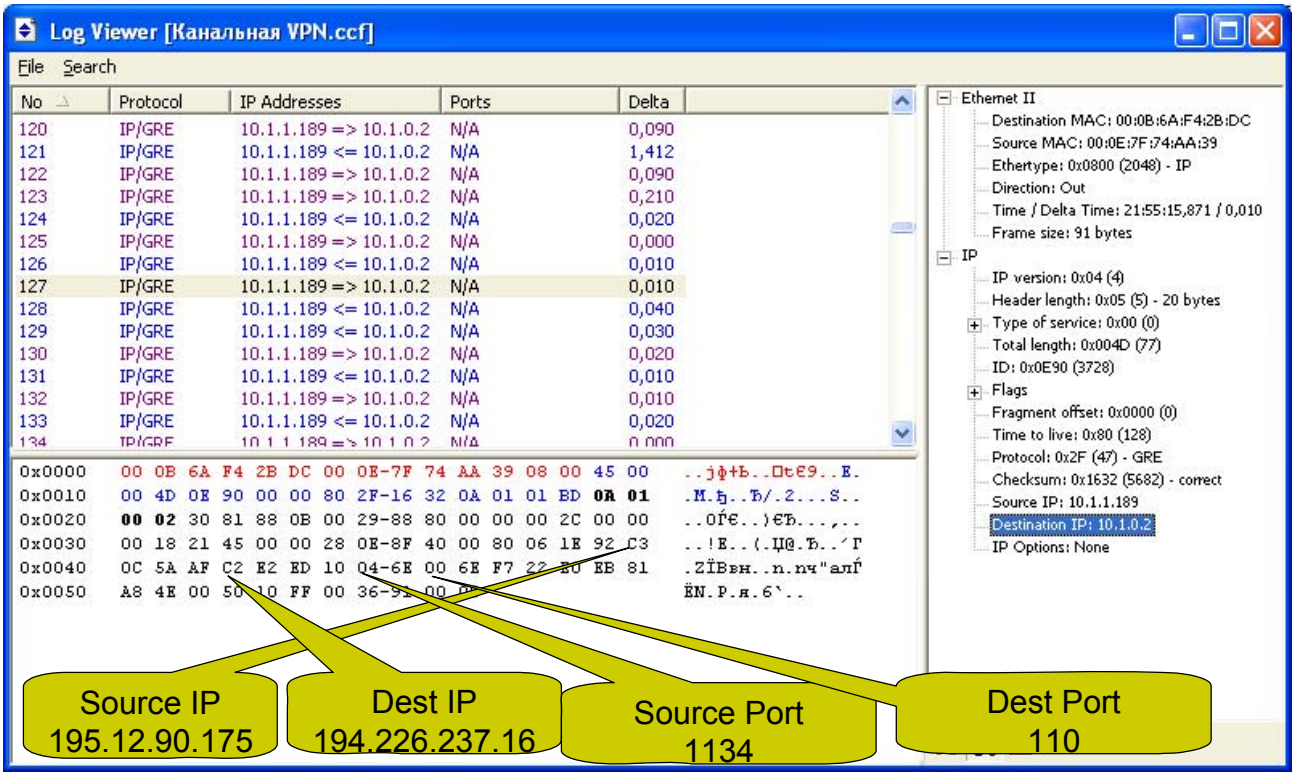


Рис. 5.4. Пакет протокола PPTP

На рис. 5.4 приведен пример обмена по протоколу POP3 (порт приемника 110), осуществляемого между удаленным POP3-сервером с адресом 194.226.237.16 и пользователем, которому назначен динамический адрес 195.12.90.175. В локальной сети видны пакеты протокола IP/GRE, проходящие между узлами 10.1.1.189 (внутренний адрес пользователя) и 10.1.0.2 (внутренний адрес PPTP-сервера).

Обычно провайдеры не включают возможность шифрования и сжатия инкапсулируемых пакетов, поэтому при анализе трафика в локальной сети содержимое IP/GRE-пакетов легко распознать и увидеть адреса, протокол и передаваемые данные.

Для шифрования передаваемых данных с использованием клиентов с ОС Windows XP необходимо в настройках подключения указать пункт «Require Data Encryption» («Требовать шифрование данных», рис. 5.5).

В протоколе PPTP для аутентификации предусматриваются различные протоколы аутентификации:

- Extensible Authentication Protocol (EAP),
- Microsoft Challenge Handshake Authentication Protocol (MSCHAP),
- Challenge Handshake Authentication Protocol (CHAP),
- Shiva Password Authentication Protocol (SPAP)
- Password Authentication Protocol (PAP)

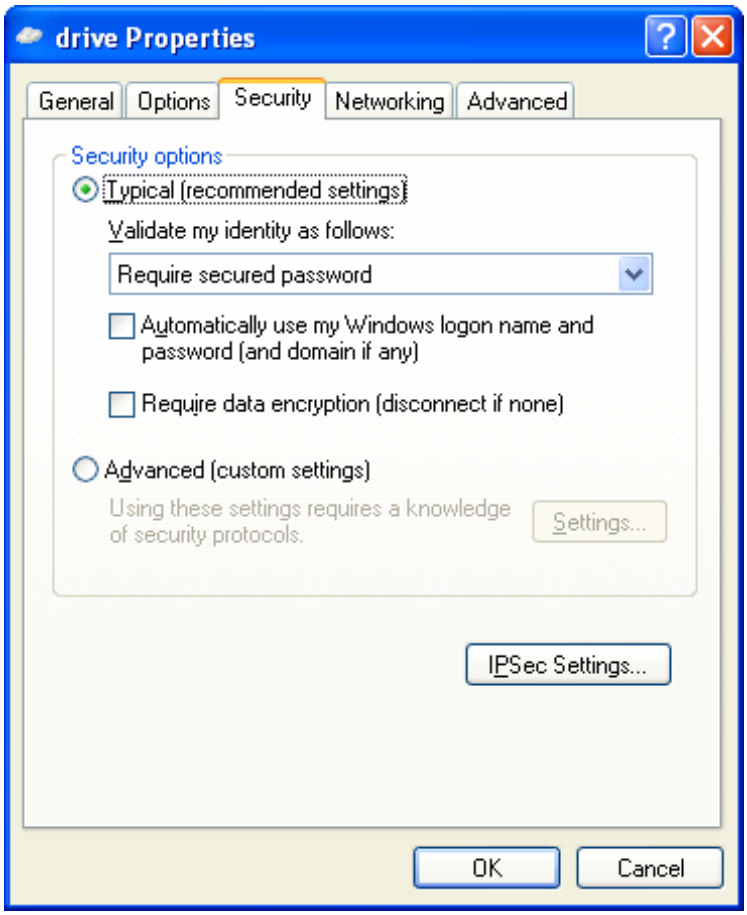


Рис. 5.5. Настройка клиента протокола PPTP

Наиболее стойким является протокол MSCHAP версии 2, требующий взаимную аутентификацию клиента и сервера. В протоколе MSCHAP могут быть использованы три различных варианта передачи пароля:

- клиент передает серверу пароль в открытом текстовом виде;
- клиент передает серверу хэш пароля;
- аутентификация сервера и клиента с использованием вызова и ответа.

Последний вариант наиболее защищенный, алгоритм его состоит в следующем (рис. 5.6).

- Клиент запрашивает вызов сетевого имени.
- Сервер возвращает 8-байтовый случайный вызов (например, «01234567», рис. 5.7).
- Клиент вычисляет хэш-функцию пароля алгоритмом «Lan Manager» (например, «C2 34 1A 8A A1 E7 66 5F AA D3 B4 35 B5 14 04 EE»), добавляет пять нулей для создания 21-байтовой строки и делит строку на три 7-байтовых ключа. Каждый ключ используется для шифрования вызова с использованием алгоритма DES, что приводит к появлению 24-байтного шифрованного значения (например, «AA AA AA AA AA AA AA AA BB BB BB BB BB BB BB BB CC CC CC CC CC CC CC CC»). Клиент выполняет то же самое с хэш-функцией пароля, получаемой алгоритмом хэширования, реализованном в ОС

семейства Windows NT. В результате формируется 48-байтное значение, которое возвращается серверу как ответ.

− Сервер ищет значение хэш-функции в своей базе данных, шифрует запрос с помощью хэш-функции и сравнивает его с полученными шифрованными значениями. Если они совпадают, аутентификация заканчивается.

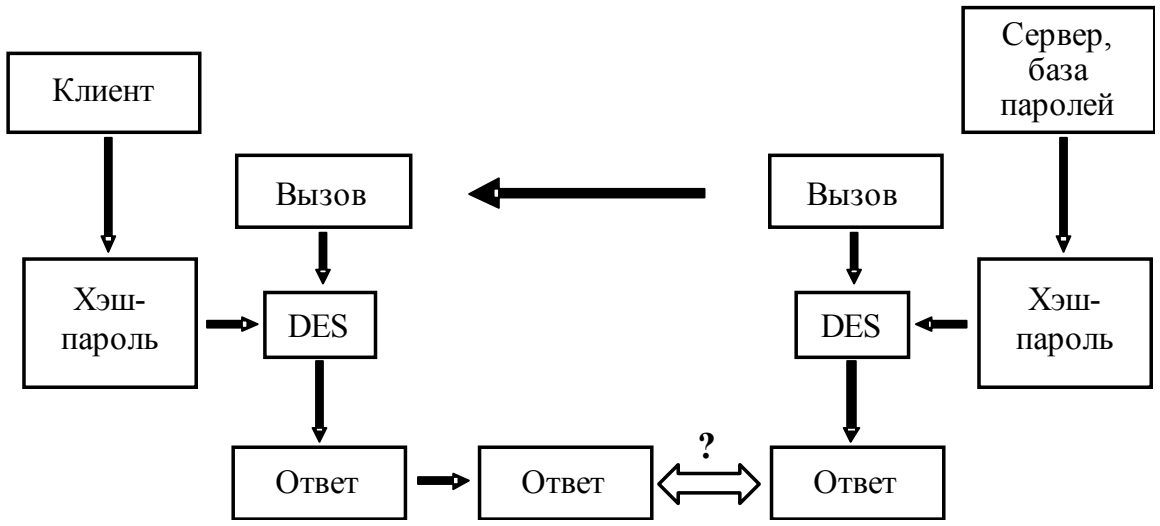


Рис. 5.6. Аутентификация в протоколе MSCHAP

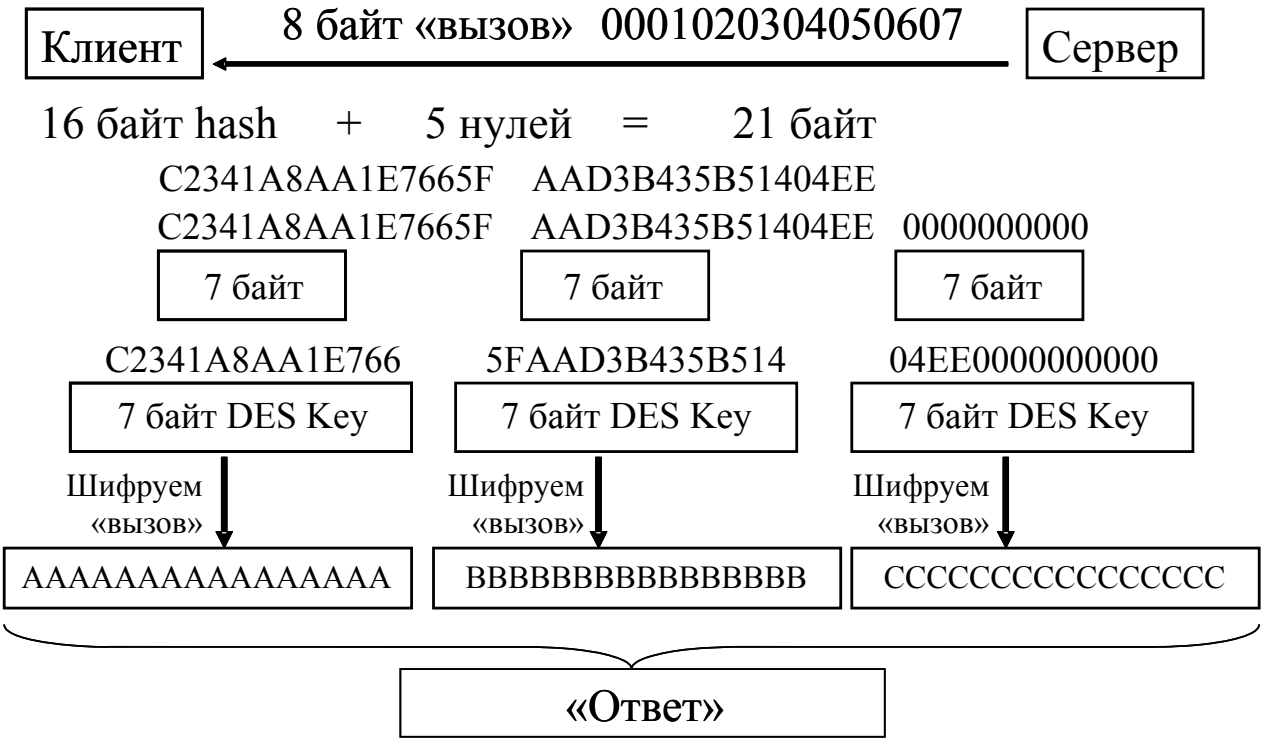


Рис. 5.7. Схема формирования «ответа» в протоколе MSCHAP

Для шифрования передаваемых данных применяется поточный шифр RC4 с 40- либо 128-разрядным ключом. Алгоритм предполагает существова-

ние секретного ключа, известного обоим участникам соединения. Данный ключ формируется из хэш-функции «Lan Manager» пароля пользователя, известного и клиенту, и серверу.

5.5. Организация VPN средствами протокола PPTP

5.5.1. Постановка задачи

Предлагается организовать соединение по протоколу PPTP между двумя сетевыми узлами. При этом имитируется соединение, которое пользователь Интернет устанавливает с сервером провайдера в том случае, когда используется подключение по выделенному каналу на основе Ethernet. В результате подключения пользователю выделяется IP-адрес, который может быть известен пользователю заранее либо выделяться динамически. Динамическое выделение адресов позволяет затруднить идентификацию узла пользователя из Интернет, сделав его в какой-то степени анонимным. Кроме того, это дает возможность провайдеру более эффективно использовать выделенное ему адресное пространство.

Для имитации предполагается использовать два рабочих места. Первое рабочее место (рис. 5.8) имитирует PPTP-сервер Интернет-провайдера, этим сервером является компьютер под управлением ОС Windows 2000/XP. На этом же рабочем месте имитируется пользовательский компьютер, который выполняется в виде виртуальной машины VMWare с установленной Windows 2000.

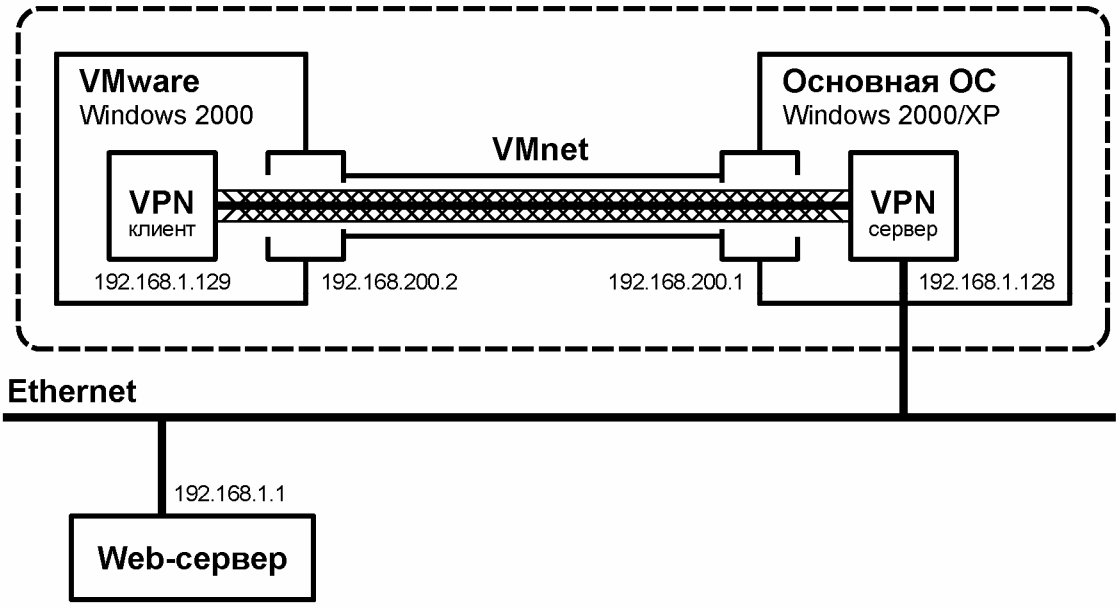


Рис. 5.8. Схема имитируемой VPN-сети

Второе рабочее место (им может быть любой компьютер в локальной сети) имитирует удаленный web-сервер.

Предполагается, что удаленный web-сервер имеет IP-адрес 192.168.1.1, основной компьютер имеет два интерфейса — внутренний с адресом 192.168.200.1 и внешний с адресом 192.168.1.128. Пользовательский компьютер имеет внутренний адрес 192.168.200.2. Пройдя авторизацию на PPTP-сервере, пользовательский компьютер получит адрес внешней сети 192.168.1.129. В дальнейшем пользовательский компьютер будет обращаться к внешнему web-серверу по протоколу HTTP.

Анализ трафика будет осуществляться в локальной сети между пользовательским компьютером и PPTP-сервером.

5.5.2. Установка и настройка VPN

ВЫПОЛНИТЬ!

1. Настроить виртуальную сеть между основной ОС и виртуальной машиной Windows 2000. Для этого выполнить следующие действия.
2. В общих настройках виртуальной сети включить адаптер VMnet1 (опция «Enable adapter», рис. 5.9).

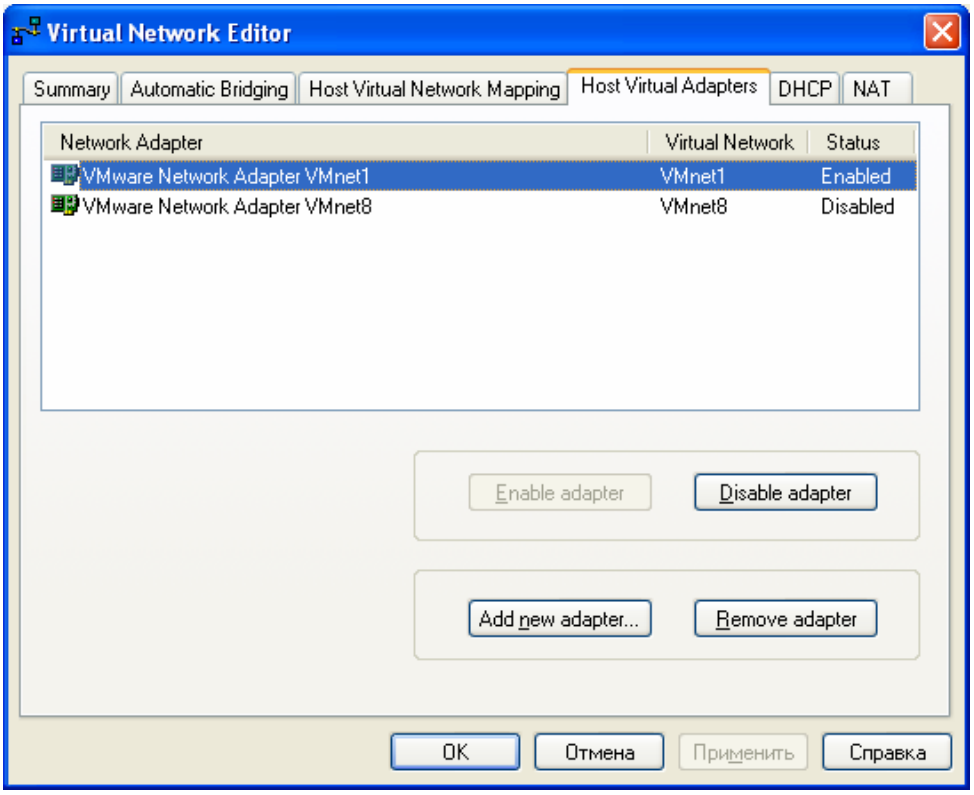


Рис. 5.9. Активация адаптера VMnet1

3. В разделе «Host Virtual Network Mapping» настроить свойства адаптера VMnet1, указав подсеть 192.168.200.0 (рис. 5.10).

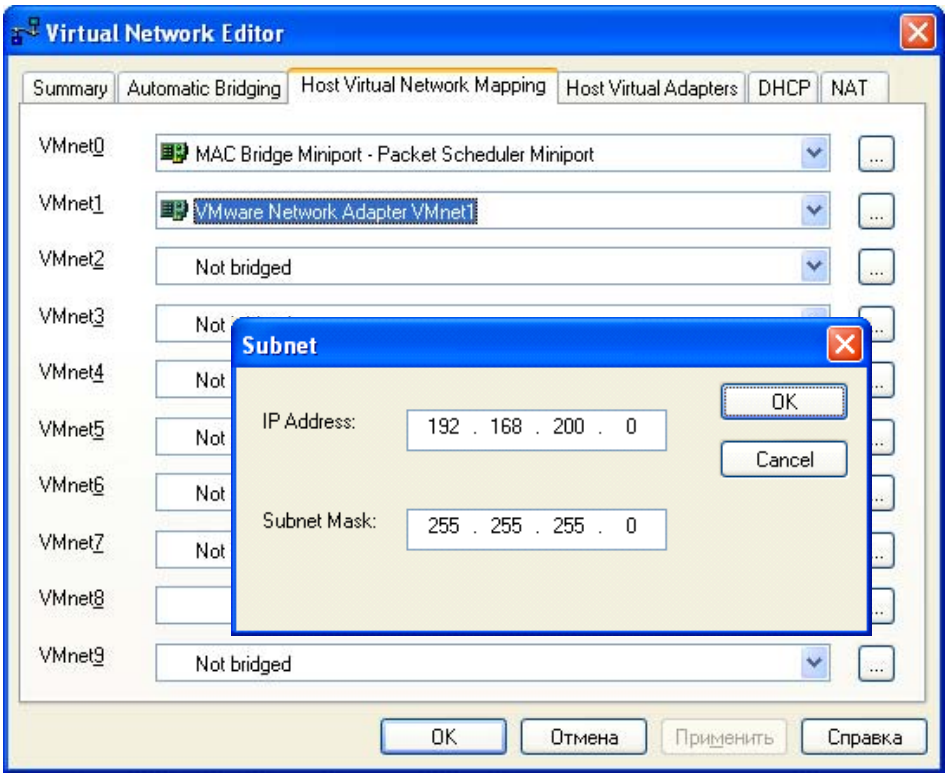


Рис. 5.10. Настройка подсети адаптера VMnet1

4. В настройках загружаемой виртуальной машины указать подключение к адаптеру VMnet1 (рис. 5.11).
5. Установить IP-адрес виртуальной машины 192.168.200.2.
6. Установить IP-адрес адаптера VMnet1 основной ОС (VMware Network Adapter VMnet1) 192.168.200.1.
7. Подключение по локальной сети основной ОС настроить на IP-адрес 192.168.1.128.
8. Добавить в основной ОС входящее подключение VPN, для чего в свойствах «Сетевого окружения» запустить «Мастер новых подключений». С помощью мастера последовательно установить следующие параметры: «Установить прямое подключение к другому компьютеру»; «Принимать входящие подключения»; «Разрешить виртуальные частные подключения»; указать учетную запись, которая будет использована для подключения.
9. Настроить в основной ОС входящее подключение VPN в разделах:
   «Общие» $\Rightarrow$ «Разрешить другим пользователям устанавливать частное подключение к моему компьютеру с помощью туннеля в Интернете или другой сети» (установлен).
   «Пользователи» $\Rightarrow$ «Все пользователи должны держать в секрете свои пароли и данные» (сброшен)

«Сеть» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Разрешить звонящим доступ к локальной сети» (установлен)
«Сеть» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Указать IP-адреса явным образом» (192.168.1.128 — 192.168.1.254)
«Сеть» ⇒ «Клиент для сетей Microsoft» (установлен)
«Сеть» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (установлен)
Остальные параметры оставить по умолчанию.

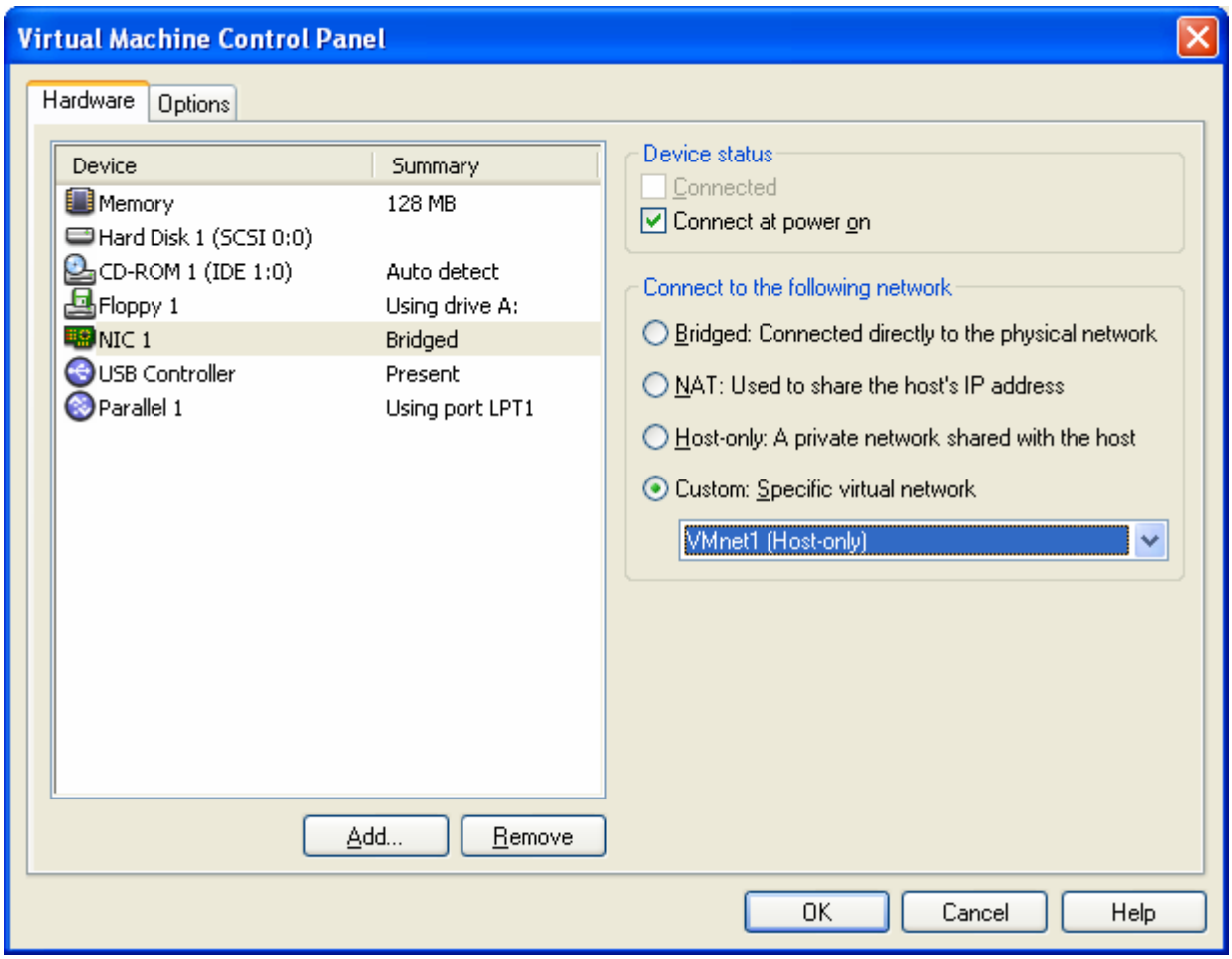


Рис. 5.11. Настройка адаптера виртуальной машины на адаптер VMnet1

10. Добавить в ОС виртуальной машины подключение к виртуальной частной сети через Интернет со следующими параметрами:
«IP-адрес компьютера, к которому осуществляется подключение» (IP-адрес назначения): 192.168.200.1
«Безопасность» ⇒ «Требуется шифрование данных» (сброшен)
«Сеть» ⇒ «Тип вызываемого сервера VPN» ⇒ «Туннельный протокол точка-точка (PPTP)»
«Сеть» ⇒ «Тип вызываемого сервера VPN» ⇒ «Настройка» ⇒ «Программное сжатие данных» (сброшен)
«Сеть» ⇒ «Клиент для сетей Microsoft» (установлен)

«Сеть» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (установлен)

11. Чтобы предотвратить возможность сетевого доступа к файлам и каталогам основной ОС с виртуальной машины в обход туннеля VPN, необходимо дополнительно установить следующие параметры для соединения VMnet1 в основной ОС:
    «Общие» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Дополнительно» ⇒ «WINS» ⇒ «Отключить NetBIOS через TCP/IP»
    «Общие» ⇒ «Клиент для сетей Microsoft» (сброшен)
    «Общие» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (сброшен)
    Аналогичные параметры должны быть установлены для подключения к локальной сети в ОС виртуальной машины (тоже, фактически, VMnet1):
    «Общие» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Дополнительно» ⇒ «WINS» ⇒ «Отключить NetBIOS через TCP/IP»
    «Общие» ⇒ «Клиент для сетей Microsoft» (сброшен)
    «Общие» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (сброшен)
12. Установить виртуальное частное подключение. Выяснить адрес, выделенный клиенту, а также адрес сервера. При установленном параметре «Разрешить звонящим доступ к локальной сети» подключившийся таким образом клиент становится узлом локальной сети, но только на сетевом уровне модели OSI и выше.

5.5.3. Анализ защищенности передаваемой информации

Предлагается изучить степень защищенности передаваемой по туннельному соединению информации с использованием анализатора сетевого трафика.

ВЫПОЛНИТЬ!

13. На втором рабочем месте запустить произвольный web-сервер.
14. Запустить анализатор трафика и настроить его на перехват пакетов, передаваемых виртуальным сетевым адаптером VMnet1.
15. Отправить из ОС виртуальной машины несколько ECHO-запросов в адрес сервера двумя способами: сначала напрямую через сеть VMnet1 (адрес сервера 192.168.200.1), а затем через туннельное соединение (адрес сервера необходимо выяснить при помощи диалогового окна состояния соединения). Обратите внимание, что пакеты, посылаемые через туннельное соединение, не опознаются как ICMP-пакеты. Поскольку шифрование передаваемой информации и программное сжатие отключены, то содержимое исходного IP-пакета сохраняется в первоначальном виде. Изменения в передаваемой информации заключаются только в том, что к исходному паке-

ту добавляется заголовок протокола PPTP, который затем снимается при выходе пакета из туннеля.

16. Перевести IP-адреса источников и приемников ECHO-запросов (всего 4 различных адреса) в шестнадцатеричную систему исчисления. Найти эти адреса в перехваченных пакетах. Убедиться, что при туннелировании IP-адреса остаются неизменными и могут быть восстановлены в случае перехвата трафика. Привести пакеты ECHO-запросов, отправленных напрямую и через туннель, и выделить в них соответствующие IP-адреса.
17. Запустить на виртуальной машине Internet Explorer и подключиться к запущенному в локальной сети web-серверу. При помощи анализатора трафика посмотреть пакеты, передаваемые через интерфейс VMnet1. Найти HTTP-запросы, отправляемые на 80 (50h) порт web-сервера, а также ответы сервера, отправляемые с 80 порта. Текст HTTP-запроса начинается со слова GET, следующего за ним пробела и далее URL запрашиваемого ресурса. Сравнить эти пакеты с пакетами, передаваемыми по локальной сети. В чем выражено отличие этих пакетов?
18. Разорвать виртуальное соединение.
19. Включить шифрование передаваемой информации, для этого в свойствах соединения в ОС виртуальной машины установить следующий параметр:
    Безопасность ⇒ Шифрование данных
20. Установить виртуальное соединение. Отправить из ОС виртуальной машины несколько ECHO-запросов через туннельное соединение. Просмотреть перехваченный трафик, есть ли возможность установить, пакеты какого содержания передавались? Зашифрованы ли поля заголовков? Какая информация может быть перехвачена злоумышленником в случае его подключения к линии связи?

5.6. Защита данных на сетевом уровне

На сетевом уровне применяются два основных алгоритма: SKIP и IPSec. Различие в алгоритмах, главным образом, состоит в способе генерации и передачи ключей для шифрования содержимого пакетов.

5.6.1. Протокол SKIP

Протокол SKIP (Simple Key management for Internet Protocol – простое управление ключами для IP-протокола) разработан компанией Sun Microsystems в 1994 году. Основными его свойствами являются: аппаратная независимость, прозрачность для приложений и независимость от системы шифрования. Последнее очень важно ввиду того, что в большинстве стран мира, включая и Россию, существуют ограничения на применяемые в данной стране стандарты шифрования передаваемых данных. Таким образом, при реализации алгоритма в каждой стране может быть применен свой стандарт шифрования, в частности в России применяется симметричный алгоритм

ГОСТ 28147-89. Широко известная реализация — линейка программных продуктов «Застава» российской компании «ЭЛВИС+».

В основе алгоритма лежит система открытых ключей Диффи-Хелмана. В этой системе предполагается наличие у каждого из пользователей пары ключей. Каждый пользователь системы защиты информации имеет секретный ключ *Кс*, известный только ему, и открытый ключ *Ко*. Открытые ключи могут быть выложены на любом общедоступном сервере.

Особенностью схемы является то, что открытый ключ *Ко* вычисляется из секретного ключа **Кс**. Вычисление осуществляется следующим образом: $Ko = gKc \bmod n$, где g и n — некоторые заранее выбранные достаточно длинные простые целые числа.

При этом если узел *J* устанавливает соединение с узлом *I*, то они легко могут сформировать общий ключ для симметричного алгоритма шифрования данных, воспользовавшись возможностью вычисления общего для них разделяемого секрета *Kij*:

$Kij = Koj * Kci = (gKcj) * Kci \bmod n = (gKci) * Kcj \bmod n = Koi * Kcj = Kij$.

Иными словами, отправитель и получатель пакета могут вычислить разделяемый секрет на основании собственного секретного ключа и открытого ключа партнера.

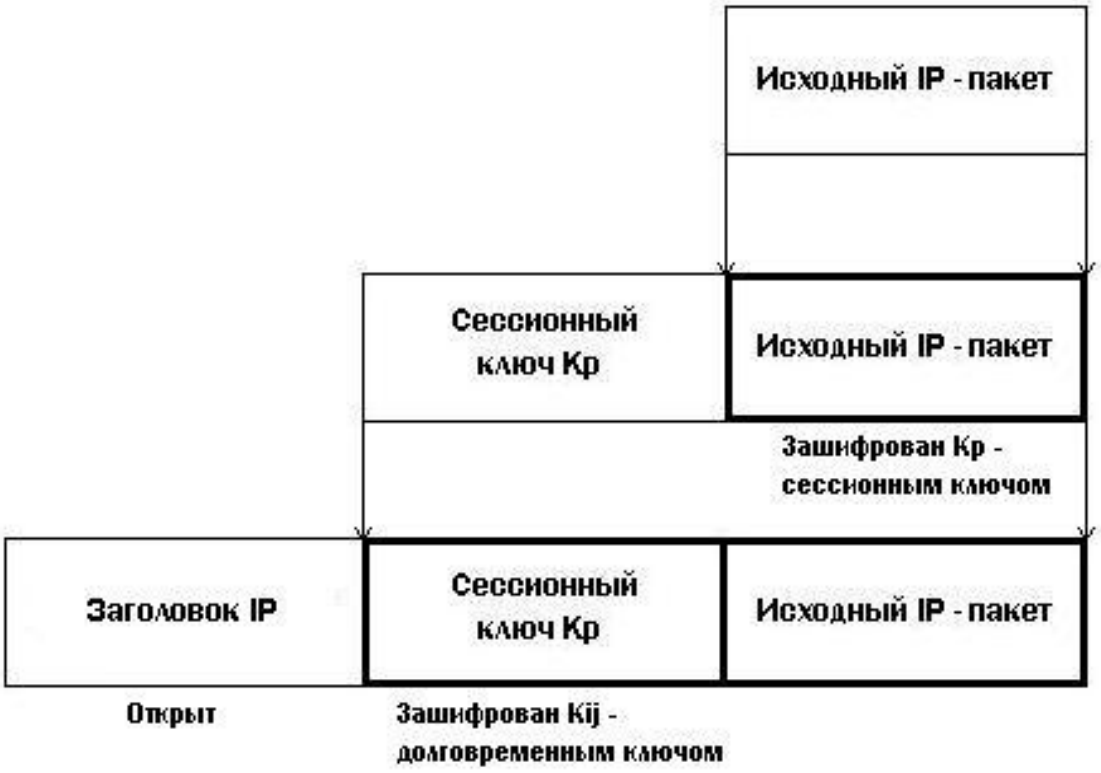


Рис. 5.12. Схема создания SKIP-пакета

Полученный ключ *Kij* является долговременным разделяемым секретом для любой пары абонентов *I* и *J* и не может быть вычислен третьей стороной,

так как секретные ключи *Kci* и *Kcj* в сетевом обмене не участвуют и третьей стороне не доступны.

Таким образом, разделяемый секрет не требуется передавать по линии связи для организации соединения, и он пригоден в качестве ключа для симметричного алгоритма шифрования. Однако на практике для шифрования отдельных пакетов применяют так называемый пакетный ключ, который помещают в заголовок SKIP-пакета и зашифровывают с помощью разделяемого секрета.

Далее полученный пакет дополняется новым IP-заголовком, адресами в котором являются адреса туннелирующих узлов (рис. 5.12).

Преимуществами такого решения являются, во-первых, дополнительная защита разделяемого секрета, так как он используется для шифрования малой части трафика (только лишь пакетного ключа) и не даёт вероятному противнику материал для статистического криптоанализа в виде большого количества информации, зашифрованного им; во-вторых, в случае компрометации пакетного ключа ущерб составит лишь небольшая группа пакетов, зашифрованных им.

В том случае, когда отсутствует необходимость шифрования или подписывания данных, соответствующие элементы, а именно пакетный ключ и ЭЦП пакета, могут отсутствовать. Необходимость шифрования и/или подписывания указывается при установке параметров SKIP-соединения. Так, в примере настроек SKIP-протокола в СЗИ «Застава», приведенном на рис. 5.13 (в нижней части рисунка), указано на необходимость шифрования данных пакетов с использованием алгоритма DES, требование аутентификации, т. е. применения ЭЦП пакета, отсутствует.

Технология, применяющая протокол SKIP, не свободна от ряда организационных проблем:

− необходимо обеспечить безопасное хранение секретных ключей *Kc* и кэширования разделяемых секретов *Kij*;

− необходимо обеспечить безопасный способ генерации и хранения (в течение относительно короткого времени жизни) пакетных ключей *Kp*;

− обеспечить сертификацию открытых ключей.

Проблема обеспечения сертификации открытых ключей возникает вследствие возможности проведения известной атаки «man-in-the-middle». Идея данной атаки не нова и состоит в следующем. Атакующая сторона находится внутри сети, где обмениваются информацией пользователи *i* и *j*. Цель атаки — хакер должен предложить от своего имени пользователю i «поддельный» открытый ключ *Koj*, а пользователю *j*, соответственно, «поддельный» ключ *Koi*. Данное действие вполне возможно вследствие того, что открытые ключи пользователей должны располагаться в общедоступном месте, где обязательно должна быть разрешена запись файлов (иначе никто не сможет поместить туда свой открытый ключ). После того, как подмена ключей осуществится, третья сторона сможет принимать весь шифрованный трафик от одного

абонента, расшифровывать, читать, шифровать под другим ключом и передавать другому абоненту. Иными словами, весь зашифрованный трафик пойдет через «человека в центре».

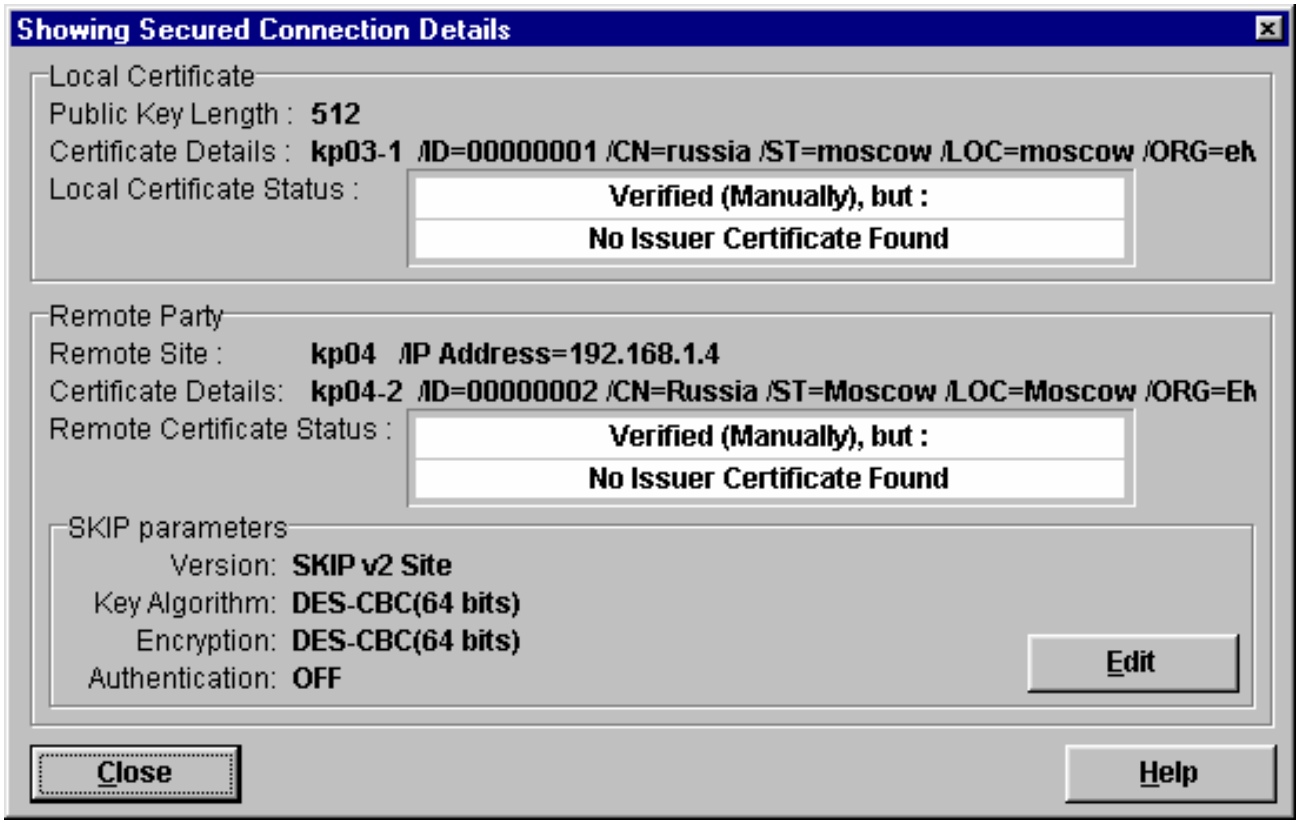


Рис. 5.13. Настройки параметров протокола SKIP

В качестве защиты от подобной атаки применяется сертификация открытых ключей. Смысл сертификации заключается в создании электронного документа — сертификата открытого ключа. В данном документе кроме самого электронного ключа должна содержаться информация о том, кому данный сертификат выдан, каков срок его действия, кем выдан, и, самое важное, должна присутствовать ЭЦП открытого ключа, сгенерированная организацией, выдавшей сертификат. Зная эту организацию, любой пользователь, желающий проверить подлинность сертификата, может получить ее открытый ключ и проверить ЭЦП, хранящуюся в сертификате.

Предполагается, что распределением открытых ключей должна заниматься заслуживающая доверия сторона. В зарубежной литературе для подобного органа используется термин Certificate Authority («Нотариус»), в российских документах он именуется Центром сертификации (ЦС).

Как уже говорилось, сертификат — файл определенного формата. Наибольшее распространение получил формат сертификата, установленный Международным телекоммуникационным союзом — ITU Rec. X.509. Электронный сертификат стандарта X.509 содержит: имя издателя сертификата; имя владельца сертификата; открытый ключ владельца; срок действия открытого

(секретного) ключа издателя и владельца; дополнения; списки отозванных сертификатов. Пример сертификата открытого ключа в формате X.509 приведен в табл. 5.2.

Таблица 5.2

Пример сертификата открытого ключа

Поле	**Пример значения**
Версия сертификата	1, 2, 3
Серийный номер сертификата	40:00:00:00:00:00:00:ab:38:1e:8b:e9:00:31:0c:60
Идентификатор алгоритма ЭЦП	ГОСТ Р 34.10-94
Имя издателя сертификата	C=RU, ST=Moscow,O=PKI, CN=Certification Authority
Срок действия сертификата	Действителен с: Ноя 2 06:59:00 1999 GMT Действителен по: Ноя 6 06:59:00 2004 GMT
Имя владельца сертификата	C=RU, ST=Moscow, O=PKI, CN=Sidorov
Открытый ключ владельца	тип ключа: Открытый ключ ГОСТ длина ключа: 1024 значение: AF:ED:80:43.....
Уникальный идентификатор издателя	
Уникальный идентификатор владельца	
ЭЦП Центра сертификации	

Протокол SKIP содержит механизмы защиты от следующих видов атак.

− Атаки из сети на сервисы ОС и на прикладные программы, подключение неавторизованных узлов к сети. Механизм: в защищаемую сеть или компьютер пропускаются пакеты только от владельца разделяемого секрета.

− Прослушивание трафика. Механизм: передаваемые пакеты могут быть прочитаны только владельцем разделяемого секрета.

− Повторение пакетов. Механизм: в аутентифицирующую часть заголовка SKIP-пакета перед вычислением криптосуммы пакета подставляется, в частности, текущее время.

− Подмена/маскарад. Механизм: все пакеты и их адресная информация аутентифицируются и защищаются от подделки криптосуммой по пакету, разделяемому секрету и текущему времени.

− Перехват сессий. Механизм: в сеть может войти только владелец разделяемого секрета.

− Атака Man-in-the-middle. Механизм: подписанные ЦС сертификаты.

− Анализ топологии сети. Механизм: топология сети полностью скрывается туннелированием всех исходящих из сети пакетов.

− Криптоанализ. Механизм: большая длина пакетных ключей (до 256 бит); частая смена пакетных ключей – через каждые 5-10 IP- пакетов; отсутствие данных для криптоанализа разделяемого секрета — он не используется непосредственно для криптообработки.

− Атака: отказ в обслуживании. Механизм: нейтрализуется для всех DoS атак, ведущихся на уровне выше чем IP. В сеть пропускаются пакеты только от владельца разделяемого секрета.

Вместе с тем, защита от ряда атак протоколом не реализуется:

− осуществляется защита лишь части трафика, например направленного в удаленный филиал. Остальной трафик (например, к web-серверам) проходит через VPN-устройство без обработки;

− нет защиты от действий пользователей, имеющих санкционированный доступ в корпоративную сеть.

5.6.2. Протокол IPSec

Протокол IPSec позволяет осуществлять две важнейшие функции сетевой защиты — осуществлять криптографическую защиту трафика и выполнять фильтрацию входящих/исходящих пакетов. Протокол реализован в ОС Windows 2000/XP. Протокол обеспечивает аутентификацию участников сетевого обмена (протокол IKE — Internet Key Exchange), защиту целостности (заголовок аутентификации AH — Authentication Header) и шифрование (ESP — Encapsulating Security Payload)

Аутентифицирующий заголовок (AH) выполняет защиту от атак, связанных с несанкционированным изменением содержимого пакета. Для этого особым образом применяется алгоритм MD5: в процессе формирования AH последовательно вычисляется хэш-функция от объединения самого пакета и некоторого предварительно согласованного ключа, затем от объединения полученного результата и преобразованного ключа.

Заголовок ESP служит для обеспечения конфиденциальности данных, предполагает возможность использования любого симметричного алгоритма шифрования.

Протокол обмена ключами IKE отвечает за первоначальный этап установки соединения, способ инициализации защищенного канала, процедуры обмена секретными ключами, выбор метода шифрования. Предполагает три

различных способа аутентификации: технологию «вызов-ответ» с использованием хэш-функции с общим секретным ключом, применение сертификатов открытых ключей и использование протокола Керберос.

5.7. Организация VPN средствами СЗИ VipNet

5.7.1. Постановка задачи

Рассмотрим некоторую гипотетическую организацию, ведущую проектирование инженерной документации, составляющей коммерческую тайну. Готовые проекты передаются по защищенному каналу в удаленные филиалы.

Внедряемая система защиты должна обеспечить защиту от несанкционированного доступа к передаваемым данным, удовлетворяя следующим требованиям:

− только зарегистрированные пользователи могут иметь возможность входа в систему и обмена конфиденциальной информацией;

− передаваемая конфиденциальная информация должна быть защищена криптографическими методами, обеспечивающими её конфиденциальность, целостность и подлинность;

− в целях расследования возможных инцидентов должна вестись регистрация в журналах наиболее важных событий, связанных с передачей защищаемой информации по каналам связи;

− должна быть обеспечена безопасная работа пользователей в Интернет средствами межсетевого экранирования.

Пусть в данной организации работают администратор безопасности и два пользователя. Один пользователь работает в головном офисе, другой в удаленном филиале. Задачами администратора безопасности являются: создание логической структуры сети, определение необходимых соединений между узлами, создание ключевых наборов и генерация пользовательских паролей, установка различных уровней защиты сетевого трафика. Задачей пользователей, имеющих допуск к клиентской части ViPNet, является обмен конфиденциальной информацией.

В ходе работы имитируется функционирование четырех рабочих станций: две станции для работы пользователей и две станции для работы администратора безопасности. Администратор использует два функционально различных компьютера: ViPNet Менеджер и ViPNet Координатор.

Кроме того, имитируется работа компьютера стороннего наблюдателя (злоумышленника), имеющего возможность захватывать сетевой трафик на пути его следования. Задачей стороннего наблюдателя является анализ возможности получения доступа к конфиденциальной информации.

Лабораторная работа выполняется двумя слушателями на двух рабочих местах с использованием технологии виртуальных машин (см. Приложение 1).

Первое рабочее место имитирует компьютер пользователя основного офиса и компьютер «ViPNet Менеджер» администратора безопасности. Второе рабочее место имитирует компьютер пользователя филиала и компьютер «ViPNet Координатор» администратора безопасности. На каждом рабочем месте запускаются две виртуальные машины с ОС Windows 2000 для установки СЗИ VipNet.

Основные операционные системы на обоих рабочих местах имитируют компьютеры сторонних наблюдателей и используются для анализа сетевого трафика.

Для исследования применяется демонстрационная версия СЗИ VipNet.

5.7.2. Настройка сетевых соединений виртуальных машин

Задача данного этапа — подготовить сетевые настройки виртуальных компьютеров для обеспечения сетевого взаимодействия между ними. Сетевые настройки виртуальных машин устанавливаются для имитации присутствия в сети независимого компьютера с отдельным IP-адресом (рис. 5.14).

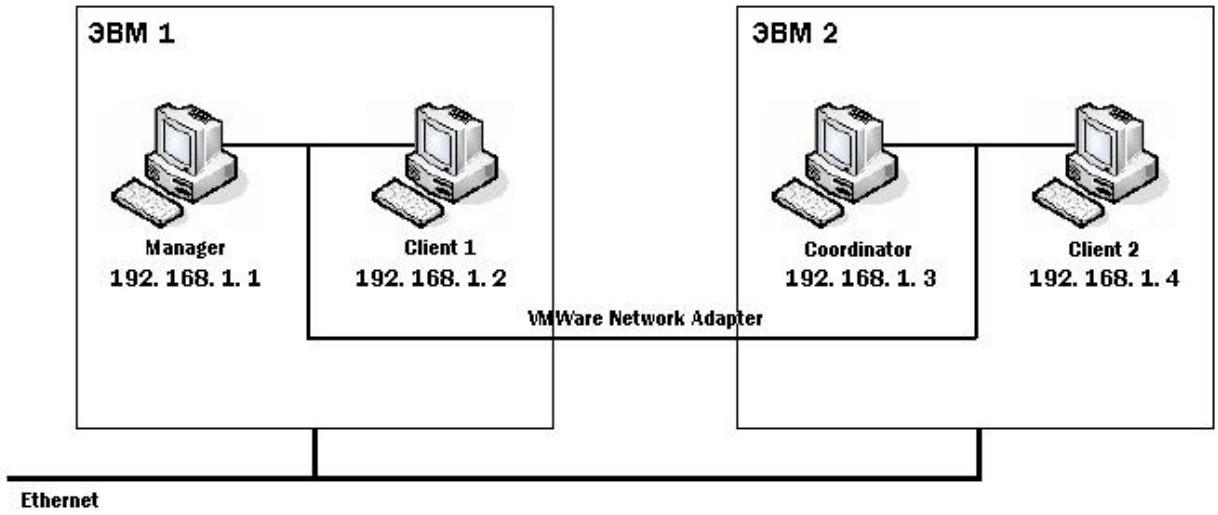


Рис. 5.14. Схема организации VPN с помощью виртуальных машин

ВЫПОЛНИТЬ!

1. На каждом рабочем месте в системе VMware открыть по два образа ОС Windows 2000. Для каждого образа на вкладке Edit выбрать меню «Virtual Machine Settings» и установить размер потребляемой памяти (Guest size) — 64 MB, а тип сетевого подключения — «bridged». Запустить все четыре виртуальные машины.
2. Назначить виртуальным ОС уникальные сетевые имена («Manager», «Client1» — для первого рабочего места, «Coordinator», «Client2» — для второго). Для этого в каждой из виртуальных ОС следует перейти на

вкладку «Сетевая идентификация» окна «Свойства системы», в графе «Имя компьютера» ввести сетевое имя данной виртуальной машины.

3. Настроить IP-адреса запущенных виртуальных машин следующим образом. Назначить для первого рабочего места адреса: 192.168.1.1 (для «ViPNet Менеджера» на узле «Manager»), 192.168.1.2 (для «ViPNet Клиента1» на узле «Client1») и 192.168.1.5 (для основной ОС). Для второго рабочего места назначьте адреса: 192.168.1.3 (для «ViPNet Координатора» на узле «Coordinator»), 192.168.1.4 (для «ViPNet Клиента2» на узле «Client2») и 192.168.1.6 (для основной ОС). Для этого необходимо в каждой из виртуальных ОС зайти в свойства подключения по локальной сети, выбрать пункт «Протокол Интернета (TCP/IP)») и ввести IP-адрес.
4. С помощью программ ipconfig и ping убедитесь в правильной настройке сетевых адресов, а именно, в возможности получить ICMP-ответ от каждого из узлов.
5. Осуществите захват трафика в основных ОС, убедитесь в возможности анализа ICMP-пакетов.
6. Организуйте передачу текстового файла с одного клиентского компьютера («Client1») на другой («Client2»). Убедитесь в возможности захвата трафика и получения передаваемого документа.

5.7.3. Установка СЗИ VipNet

Установка ViPNet Office осуществляется в три этапа: сначала инсталлируется модуль менеджера, затем модуль координатора и в последнюю очередь – модули клиентов.

В идеологии данной версии СЗИ VipNet предполагается, что на компьютерах пользователей устанавливаются модули клиентов (рис. 5.15). Координаторы — это компьютеры, выполняющие функции туннелирующих узлов. Менеджер — это центральный узел, хранящий ключи и пароли всех пользователей.

Для построения имитируемой сети необходим один менеджер, один координатор и два клиента.

Для установки модуля «ViPNet Manager» необходимо запустить мастер (файл «Setup.exe» из каталога «\Soft\ViPNet Manager»). После перезагрузки компьютера следует активизировать программу «ViPNet Manager», нажав иконку  на рабочем столе, и создать структуру сети при помощи «Мастера создания сети ViPNet». С помощью мастера создается структура ViPNet сети, ключевые наборы и начальные пароли для всех пользователей в режиме автоматической генерации структуры с использованием готового сценария (рис. 5.16).

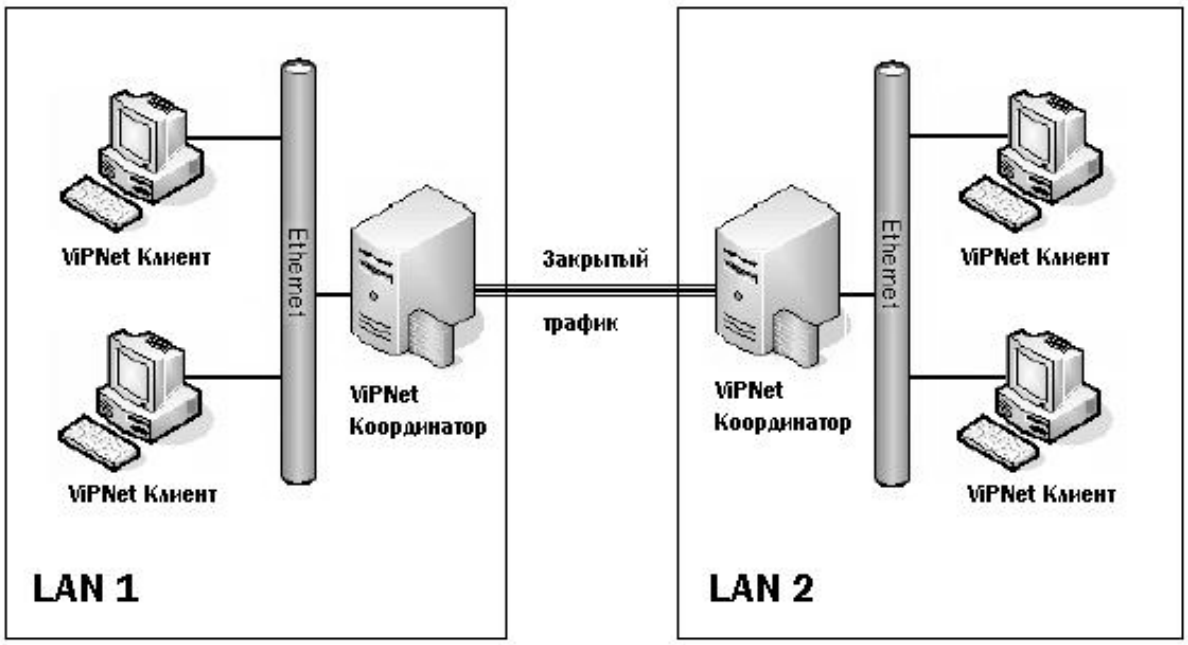


Рис. 5.15. Схема взаимодействия компонентов ViPNet в сети

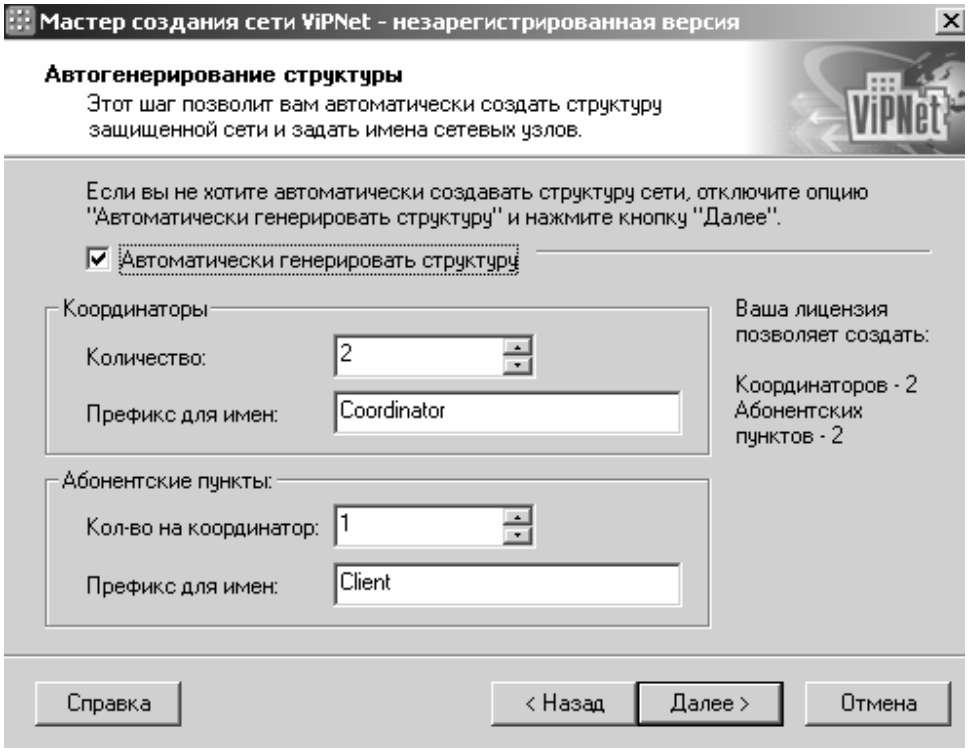


Рис. 5.16. Окно «Автогенерирование структуры»

По умолчанию префиксы имен Координаторов — «Coordinator», а префиксы имен Клиентов — «Client». Изменение вышеуказанных значений возможно при соблюдении следующих правил:

− количество Координаторов не должно быть не менее одного и не может быть больше, чем это определено лицензией;

− количество Клиентов может быть равно нулю, но не более количества определенных лицензией;

− префиксы имен сетевых узлов не должны содержать более 40 символов.

− После коррекции предложенных значений следует нажать кнопку Далее и перейти в окно «Автоматическое создание связей», в котором производится выбор стандартных сценариев для разрешенных соединений между узлами сети ViPNet.

Существуют следующие варианты установления связи:

− Связать все сетевые узлы — установлено по умолчанию. Все Клиенты и Координаторы будут иметь разрешенные VPN-соединения между собой.

− Связать все абонентские пункты каждого координатора — Клиенты, лицензированные для данного Координатора, будут иметь разрешенные VPN-соединения между собой и с соответствующим Координатором. Координаторы будут связаны VPN-соединениями между собой.

− Связать каждый абонентский пункт со своим координатором — каждый Клиент будет иметь разрешенное соединение только со своим Координатором. Координаторы будут по-прежнему иметь VPN-соединение по схеме «каждый с каждым».

После выбора оптимального варианта появляется окно «Редактирование структуры». На этом шаге предоставляется возможность модернизации созданной структуры, а также создания новой. В окне можно осуществлять следующие действия: добавлять новые сетевые узлы, переименовывать и удалять существующие узлы, переносить Клиента под обслуживание другим Координатором, удалять сетевую структуру.

После завершения редактирования структуры появляется окно, в котором следует сгенерировать системный пароль, затем, следуя инструкциям мастера установки, необходимо создать дистрибутив ключей. В процессе генерации появится окно «Электронная рулетка» — специальное приложение для генерации случайных значений.

Дистрибутив ключей для каждого сетевого узла размещен в файле с расширением «*.DST». Исходные ключи зашифрованы на парольной фразе и потому недоступны третьим лицам непосредственно из DST-файла.

Все наборы ключей и пароли к ним будут сохранены в подкаталоге «\NCC\KEYS» для каталога, куда был установлен «ViPNet Manager». Файлы с ключевыми наборами сохраняются в каталогах с именами сетевых узлов и имеют расширение «*.DST». Также будет создан файл «ViPNet.txt», в котором будут указаны пароли для соответствующих ключевых наборов.

ВЫПОЛНИТЬ!

7. Установить модуль «ViPNet Manager» на диск виртуальной машины с сетевым именем «Manager».
8. После перезагрузки виртуальной машины выполнить автоматическую генерацию структуры сети. Установить: количество координаторов — 1; ко-

личество клиентов — 2; все клиенты и координаторы должны иметь разрешенные VPN-соединения между собой (пункт «Связать все сетевые узлы»).

9. Сгенерировать наборы ключей и пароли к ним, основываясь на требованиях: словарь — английский, слов в парольной фразе — 6, используемых букв — 3.
10. После завершения работы мастера скопировать все наборы ключей и пароли к ним из каталога «C:\Program Files\InfoTeCS\ViPNet Manager\NCC\KEYS» на отдельную дискету (ключевую дискету).

Для установки «ViPNet Координатор» необходимо запустить файл «Setup.exe» из каталога «\Soft\ViPNet Coordinator\» и следовать указаниям мастера установки. В ходе последующей загрузки компьютера «ViPNet Координатор» автоматически попросит ввести соответствующий пароль из списка, находящегося в файле «ViPNet.txt», еще до появления запроса на пароль входа в ОС Windows (рис. 5.17).

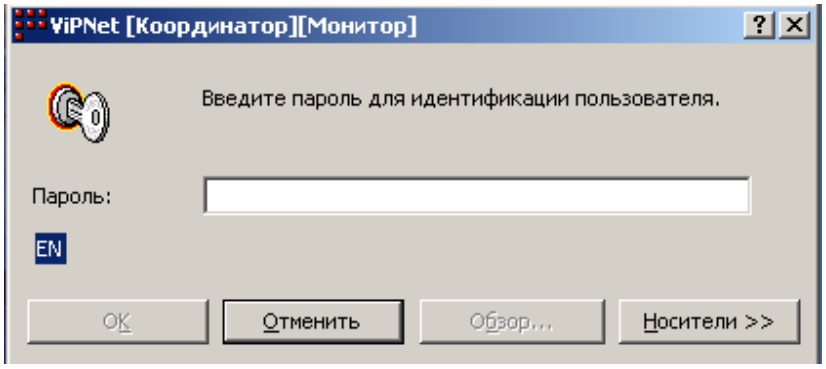


Рис. 5.17. Окно ввода пароля VipNet

В компьютер необходимо поместить внешний носитель (дискету), на котором предварительно в процессе работы с «ViPNet Manager» были записаны наборы ключей. При нажатии кнопки «Носители» появляется проводник, в котором указывается путь, где хранятся ключи на внешнем носителе. Соответствующие данному узлу ключи будут скопированы автоматически на системный диск. В дальнейшем для входа на этот же узел ViPNet внешний носитель уже не понадобится.

Установка «ViPNet Клиента» осуществляется аналогично.

ВЫПОЛНИТЬ!

11. Установить модуль «ViPNet Координатор» на диск виртуальной машины с сетевым именем «Coordinator», модули «ViPNet Клиент» на диски узлов «Client1» и «Client2».

5.7.4. Настройка СЗИ VipNet

Как указывалось выше, узлы ViPNet могут быть подключены к сети непосредственно либо могут располагаться за межсетевыми экранами и другими устройствами. Для каждого узла может быть указан один из способов подключения:

- непосредственное соединение с другими узлами;
- соединение с другими узлами через локальный Координатор, обеспечивающий технологию преобразования сетевых адресов (NAT — Network Address Translation) для трафика данного Клиента;
- соединение через межсетевой экран/NAT систему, NAT-правила которой могут быть модифицированы;
- соединение через межсетевой экран/NAT систему, установки которой не могут быть модифицированы.

После запуска каждый сетевой узел ViPNet посылает соответствующую информацию Координатору. В изучаемой лабораторной установке каждый сетевой узел имеет IP-адрес, свободно доступный другим ViPNet-узлам, поскольку все виртуальные машины находятся в одном сегменте сети. Таким образом, любому клиенту для организации взаимодействия достаточно послать Координатору только свой IP-адрес. Таким образом, при настройке узлов-клиентов достаточно для них выбрать соединение первого типа — «Непосредственное соединение с другими узлами».

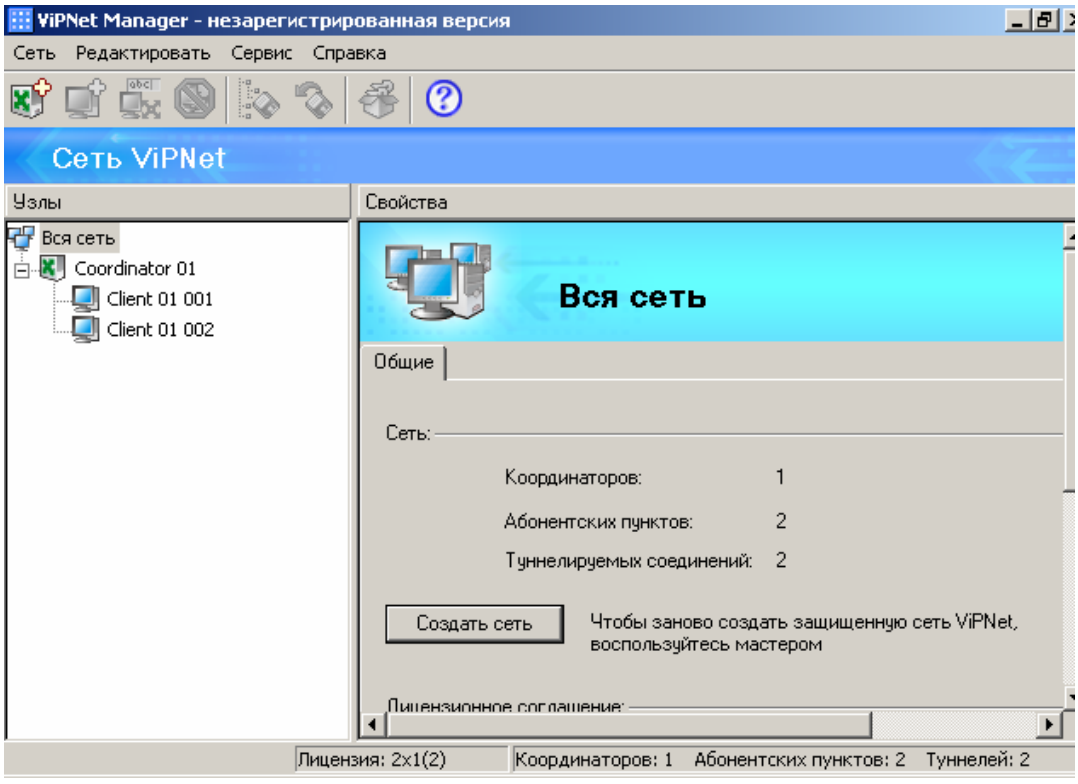


Рис. 5.18. Окно «Вся сеть» модуля «ViPNet Manager»

Настройка модуля «ViPNet Manager» может осуществляться для модернизации структуры сети, изменения сетевых узлов, создания ключевых наборов и просмотра параметров всей сети или отдельных сетевых узлов. Главное окно программы разделено на левую и правую части (рис. 5.18). На левой стороне приведена древовидная структура сети с отображением сетевых узлов. Просмотр информации о каждом отдельном объекте осуществляется посредством выбора этого объекта на изображении дерева.

При выборе корневого объекта дерева «Вся сеть» появляется информация о сети, в частности количество фактически созданных сетевых узлов, включая количество Клиентов, Координаторов и общее количество узлов.

Кнопка «Создать сеть» используется для создания абсолютно новой сети, но при этом все ранее созданные конфигурации теряются.

Информация о конкретном сетевом узле появляется в правой части главного окна после выбора этого узла на дереве структуры сети (рис. 5.19) и содержит следующие данные:

- тип узла — Координатор или Клиент;
- имя узла;
- максимально возможное количество туннелируемых соединений через Координатор (если в качестве узла выбран Координатор);
- пароль и соответствующая парольная фраза (если существует);
- путь к месту, где хранятся ключи сетевого узла (если они существуют).

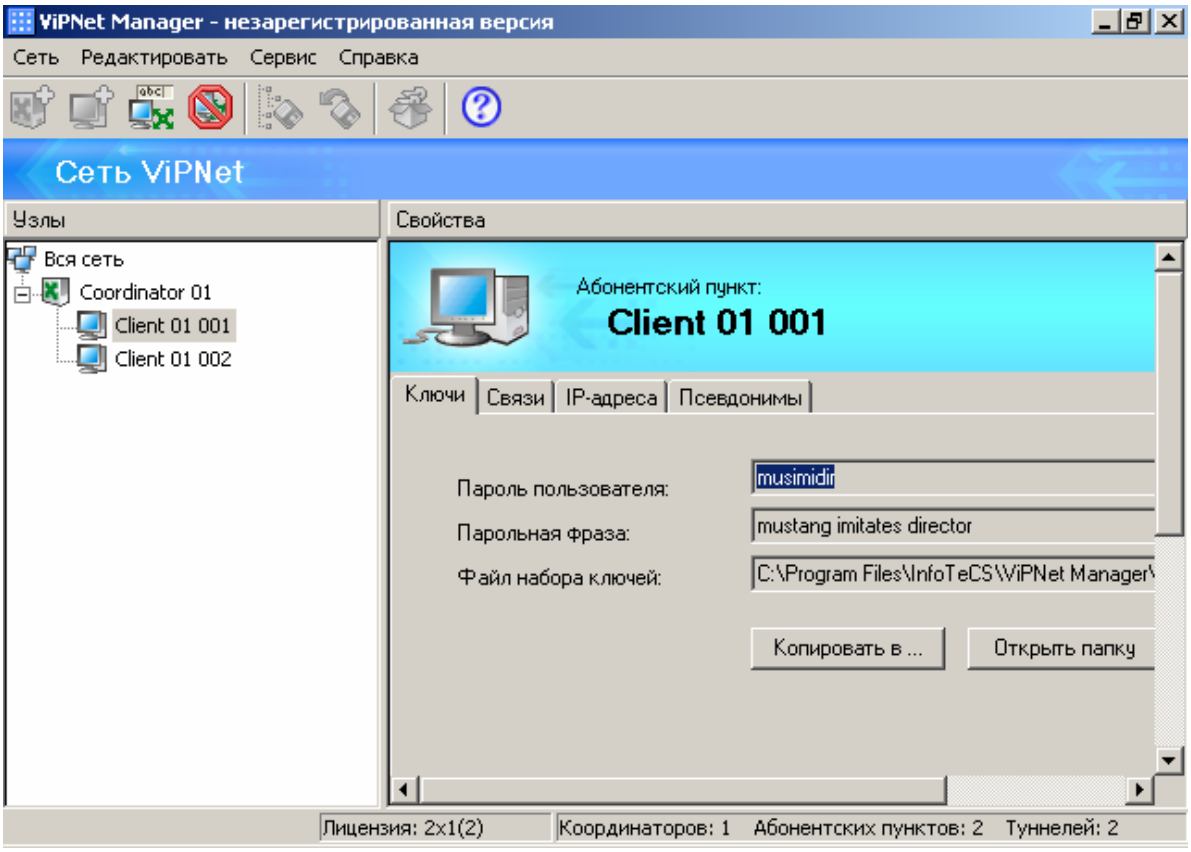


Рис. 5.19. Окно свойств сетевого узла модуля «ViPNet Manager»

Кнопка «Открыть папку» открывает подкаталог, содержащий DST-файл дистрибутива ключей для выбранного сетевого узла.

Кнопка «Копировать в…» запускает процедуру копирования ключевого набора в определенное администратором место.

Настройка модулей «ViPNet Координатор» и «ViPNet Клиент» осуществляется с помощью окна Монитора (рис. 5.20), для открытия которого следует воспользоваться иконкой , расположенной в области системного трея. Левая часть окна содержит средства конфигурирования и администрирования в виде каталогизированного дерева. Сразу после открытия окна по умолчанию выбрана секция «Защищенная сеть». В правой части окна показаны все сетевые узлы ViPNet, VPN соединение с которыми было разрешено на этапе создания структуры сети с помощью «ViPNet Manager». Сетевые узлы будут высвечиваться разными цветами:

- серый — сетевой узел отключен (находится в состоянии off-line);
- голубой — обозначает данный локальный узел;
- красный — обозначает доступные Координаторы;
- фиолетовый — ViPNet Клиенты в состоянии on-line.

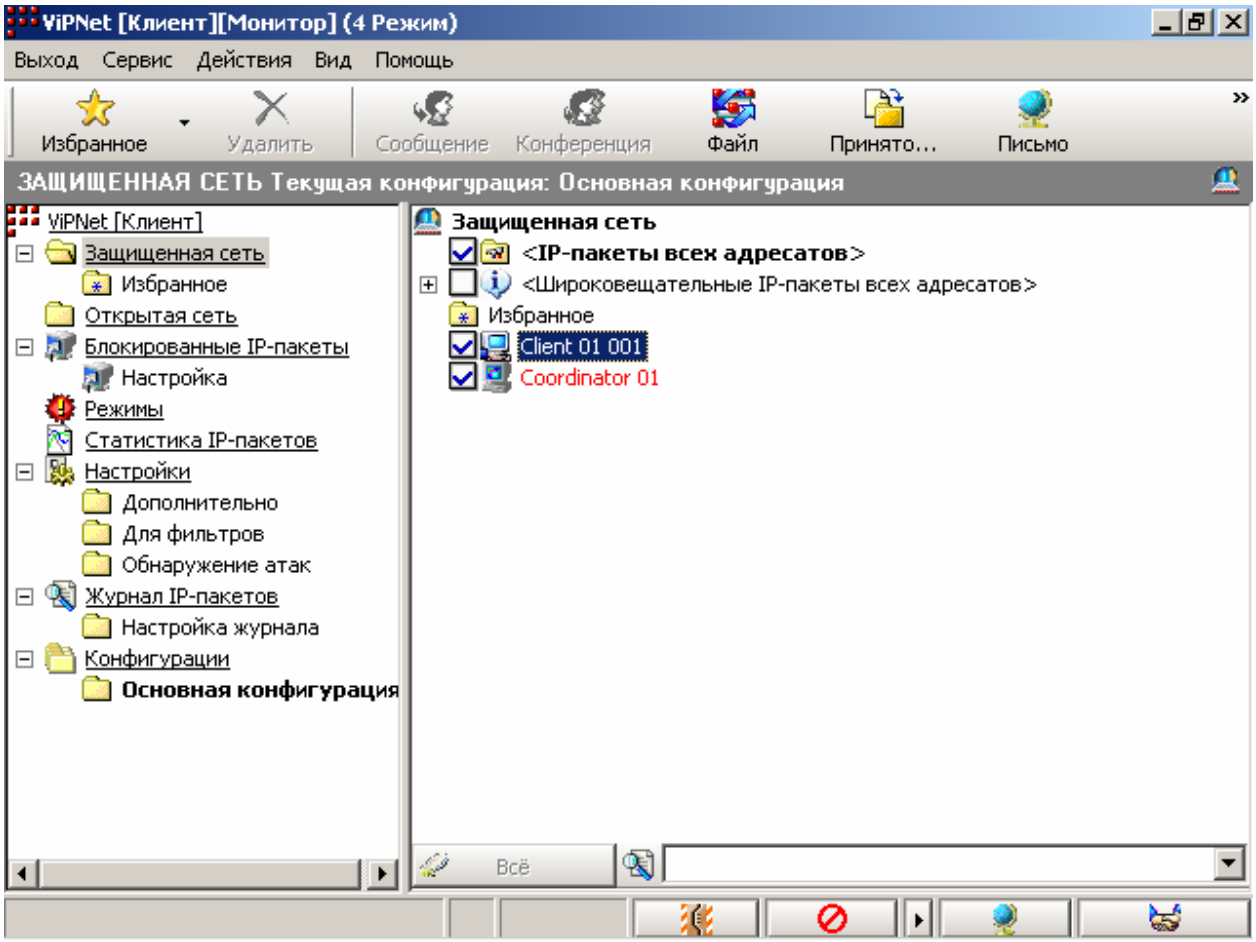


Рис. 5.20. Окно Монитор модуля «ViPNet Координатор»

Значительная часть настроек в секциях «Защищенная сеть», «Открытая сеть», «Блокированные IP-пакеты» и «Режимы связана» с настройкой работы интегрированного межсетевого экрана. Секция «Настройки» позволяет выбрать и настроить тип соединения в зависимости от реализованного физического способа подключения сетевого узла. Остальные элементы дерева содержат инструменты для получения статистической информации, создания готовых конфигураций, расширения возможностей администрирования и т. п.

ВЫПОЛНИТЬ!

12. Создать и проверить соединения между координатором и клиентами. Для этого необходимо загрузить программу «Координатор Монитор» на виртуальной машине с сетевым именем «Coordinator». В списке узлов «Защищенная сеть» выбрать соответствующего клиента, дважды кликнув на нем мышью открыть окно «Правило доступа», выбрать закладку «IP-адреса», нажать кнопку «Добавить» и ввести IP-адрес данного клиента (192.168.1.2 (для «ViPNet Клиента1» на узле «Client1») и 192.168.1.4 (для «ViPNet Клиента2» на узле «Client2»).
13. Загрузить программу «Клиент — Монитор» на виртуальных машинах «Client1» и «Client2». Провести аналогичные операции, введя IP-адрес координатора.
14. Убедиться в том, что соединение Координатор — Клиенты установлено. Для этого следует вернуться в секцию «Защищенная сеть» на виртуальной машине «Coordinator», выбрать соответствующего клиента в списке сетевых узлов, вызвать контекстное меню и выполнить команду «Проверить соединение».
15. Подготовить компьютеры «Client1» и «Client2» к файловому обмену. Для этого на «Client1» следует запустить программу «Клиент Монитор», вызвать программу «Деловая почта», нажав кнопку  на нижней панели. На верхней панели окна ViPNet [Клиент][Деловая почта] нажать кнопку «Отправить/Получить письма».
16. На панели инструментов появившегося окна (рис. 5.21) нажать кнопку «Настройки», чтобы появился список узлов ViPNet сети.
17. В появившемся списке узлов кликнуть два раза мышью на узле «Client1», чтобы появилось окно настроек почтового соединения (рис. 5.22). Установить следующие параметры:
    – Тип канала: MFTP;
    – Вызывать узел по нажатию кнопки «Опросить» (включить);
    – Ввести IP-адрес данного узла (192.168.1.2) (см. выше).

    Выполнить аналогичные операции для узла «Client2» (IP-адрес 192.168.1.4).
18. На узле «Client2» выполнить аналогичные настройки.

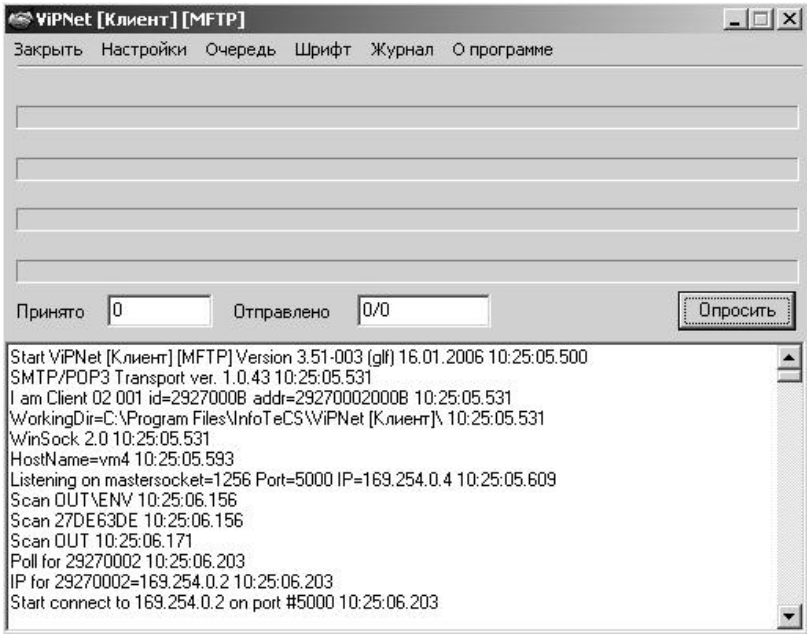


Рис. 5.21. Окно ViPNet [Клиент][Деловая почта]

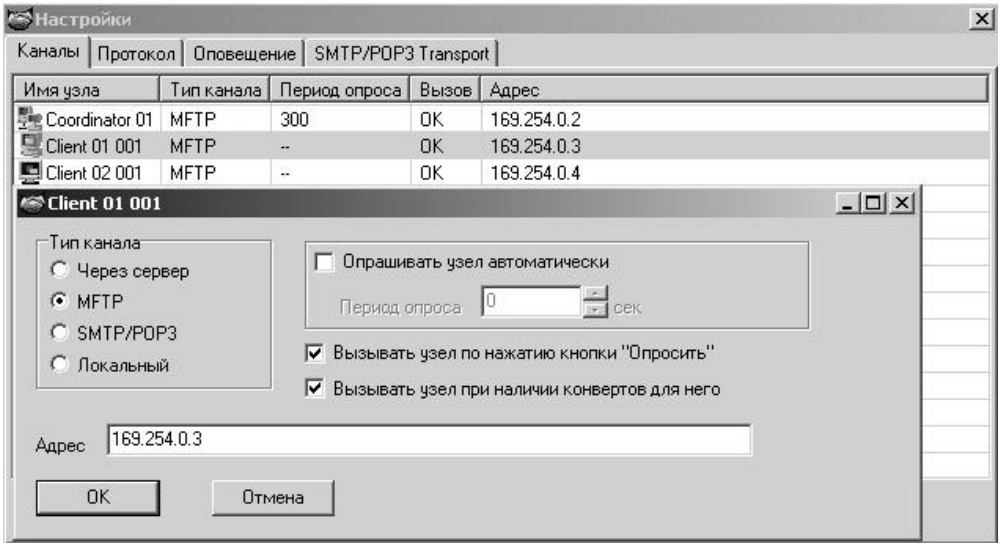


Рис. 5.22. Окно настроек почтового соединения

19. На компьютере «Client1» создать небольшой текстовый файл с произвольной информацией. Запустить анализатор трафика в режиме захвата пакетов. Запустить программу «Клиент Монитор», в папке «Защищенная сеть» выбрать получателя файла — «Client2», вызвать программу «Файловый обмен», нажав кнопку на верхней панели. В программе «Файловый обмен» ввести получателя (Client2) в поле «Адрес», присоединить текстовый файл нажатием кнопки «Добавить» и отправить письмо.
20. На компьютере «Client2» в программе «Клиент Монитор» нажать кнопку «Принято» и просмотреть полученный текстовый файл.
21. Убедиться в невозможности нахождения имени и содержимого текстового файла в захваченных сетевых пакетах.

5.8. Использование протокола IPSec для защиты сетей

5.8.1. Шифрование трафика с использованием протокола IPSec

ВЫПОЛНИТЬ!

1. Проверьте возможность анализа сетевого трафика при отключенном протоколе IPSec. Запустите анализатор сетевого трафика. Отправьте текстовый файл (набранный латинскими буквами) на виртуальный компьютер. Просмотрите захваченные пакеты. Убедитесь, что файл передается по протоколу SMB, текст файла передается в открытом виде.
2. Осуществите настройку протокола IPsec. ***Администрирование ⇒ Локальная политика безопасности ⇒ Политики безопасности IP***.
3. Обратите внимание на существование трех шаблонов: «Безопасность сервера» (при использовании данного шаблона не допускается нешифрованный трафик), «Клиент (Только ответ)» (при использовании данного шаблона возможен нешифрованный трафик, если сервер его не требует) и «Сервер (Запрос безопасности)» (при использовании данного шаблона возможен нешифрованный трафик, если клиент не поддерживает шифрование).
4. Выполните настройку шаблона «Безопасность сервера». Измените настройку фильтра «Весь IP-трафик» (рис. 5.23).

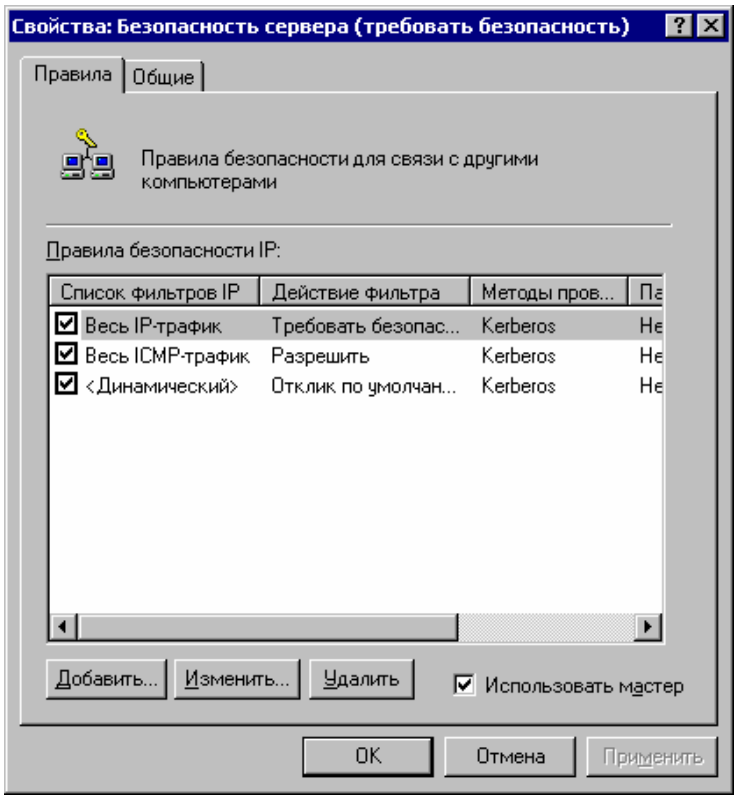


Рис. 5.23. Окно настройки шаблона «Безопасность сервера»

5. В разделе «Методы проверки подлинности» измените метод Kerberos.

6. Выберите пункт «Использовать данную строку для защиты обмена ключами» и введите произвольную текстовую строку.
7. Примените внесенные изменения и активизируйте политику, выбрав из контекстного меню данного шаблона пункт «Назначить».
8. Аналогичные действия осуществите на соседнем компьютере. Убедитесь, что ключ шифрования (текстовая строка) совпадает.

5.8.2. Проверка защиты трафика

9. Убедитесь, выполняя команду PING, что для проверки присутствия в сети вашего компьютера возможны ICMP-пакеты как от соседнего компьютера (на котором также включено шифрование), так и от любого другого.
10. Убедитесь, что в сетевом окружении вам доступен только соседний компьютер. При обращении к другим системам появляется ошибка.
11. Убедитесь путем анализа сетевого трафика при отправке на соседний компьютер текстового файла, что весь IP-трафик идет в зашифрованном виде.
12. Проверьте функционирование IPSec при использовании шаблонов «Клиент (Только ответ)» и «Сервер (Запрос безопасности)».
13. По окончании отключите шифрование трафика.

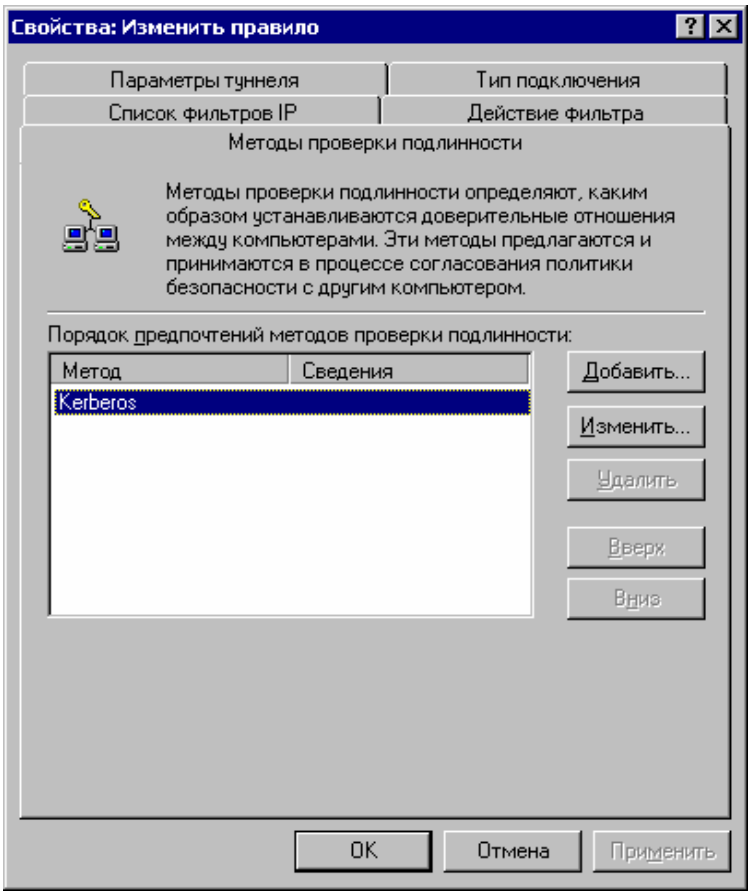


Рис. 5.24. Окно раздела «Методы проверки подлинности»

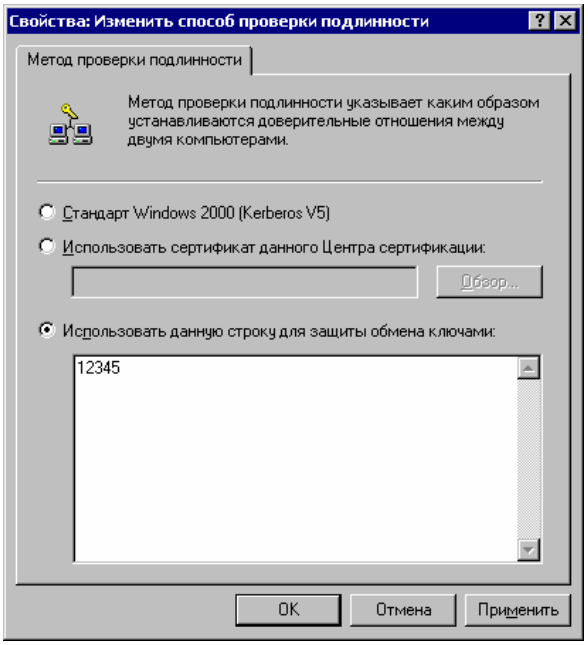


Рис. 5.25. Окно ввода ключевой строки

5.8.3. Настройка политики межсетевого экранирования с использованием протокола IPSec

Задача. Разработать политику для web-сервера, на котором разрешен только трафик через порты TCP/80 и TCP/443 из любой точки.

ВЫПОЛНИТЬ!

14. Запустите на своем узле web-сервер. Проверьте его функционирование, обратившись с другого узла.
15. Запустите утилиту настройки протокола IPSec: ***Администрирование*** ⇒ ***Локальная политика безопасности*** ⇒ ***Политики безопасности IP***.
16. Из контекстного меню выберите «Управление списками IP-фильтра и действиями фильтра». Создайте два действия (сначала сбросьте флажок «Использовать мастер»): «Разрешение» (определяющее допустимый метод безопасности) и «Блокировка» (заблокированный метод безопасности) (рис. 5.26).
17. Создайте список фильтров под названием «Любой», имеющий настройки по умолчанию, которые соответствуют всему трафику (рис. 5.27).
18. Создайте список фильтров под названием «web-доступ» (рис. 5.28) для web-сервера, разрешающего трафик на портах TCP/80 и TCP/443 из любой точки, основываясь на правилах:
    Правило 1. Источник – Любой IP-адрес. Назначение – Мой IP-адрес. Отображаемый – Да. Протокол – TCP. Порт источника – Любой (ANY). Порт назначения – 80.

Правило 2. Источник – Любой IP-адрес. Назначение – Мой IP-адрес. Отображаемый – Да. Протокол – TCP. Порт источника – Любой (ANY). Порт назначения – 443.

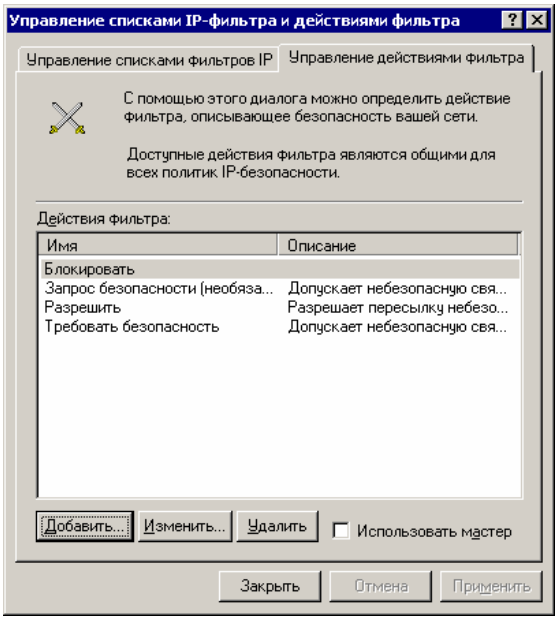


Рис. 5.26. Окно создания действий

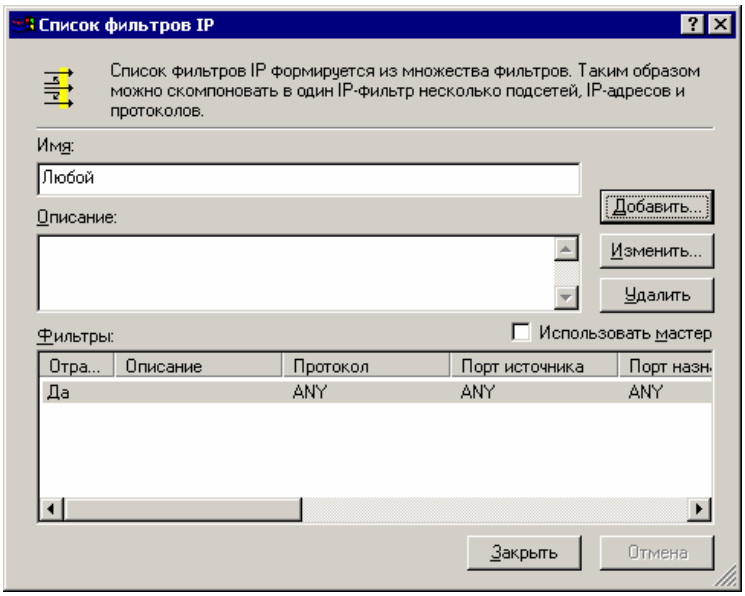


Рис. 5.27. Окно создания списка фильтров «Любой»

19. Создайте новую политику под названием «Web-доступ». Из контекстного меню окна «Политики безопасности IP» выберите «Создание политики безопасности IP». Воспользуйтесь мастером. Не активизируйте пункт «Использовать правило по умолчанию».

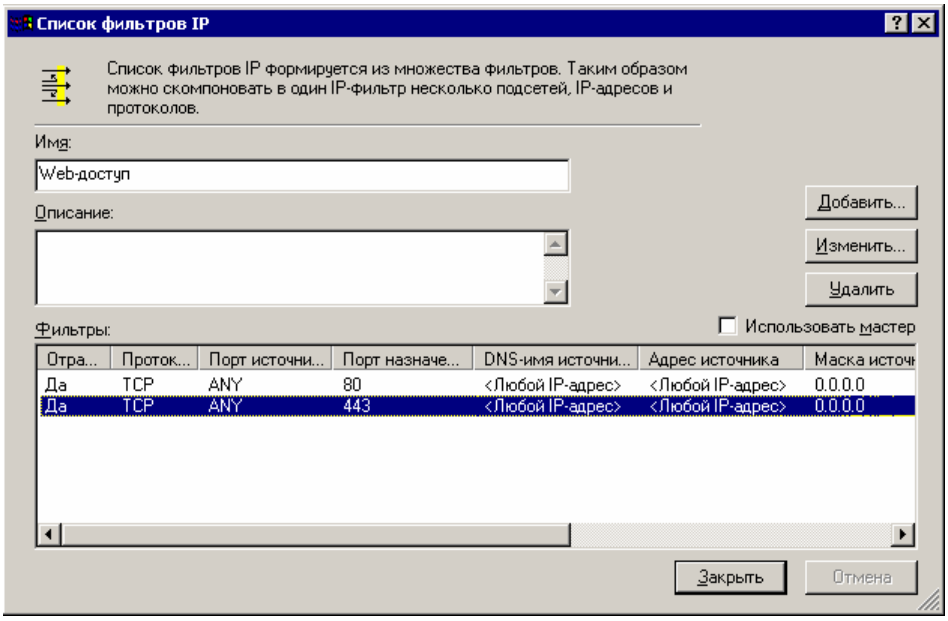


Рис. 5.28. Окно создания списка фильтров «Web-доступ»

20. Добавьте в созданную политику правило доступа «Web-доступ», использующее список фильтров «Web-доступ» и действие «Разрешение».
21. Добавьте в созданную политику правило доступа «Любой», использующее список фильтров «Любой» и действие «Блокировка» (рис. 5.29). Обратите внимание на последовательность правил.

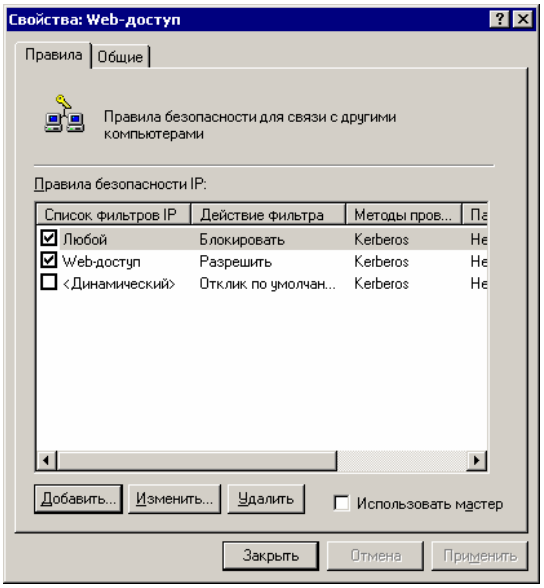


Рис. 5.29. Окно создания правила доступа

22. Примените политику «Web-доступ» (из контекстного меню политики «Web-доступ» выберите пункт «Применить», рис. 5.30).
23. Проверьте политику «Web-доступ», осуществив подключение к 80-ому порту с другого узла.

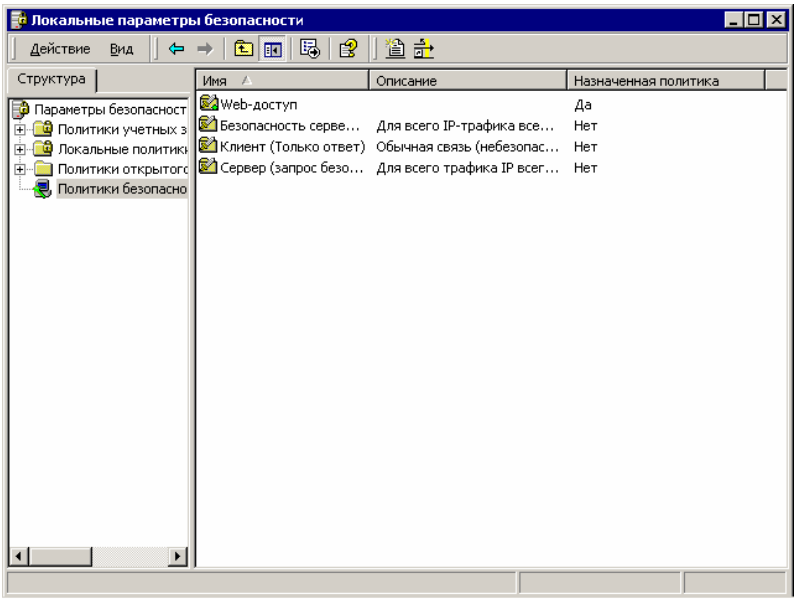


Рис. 5.30. Окно применения политики

5.9. Организация VPN средствами СЗИ StrongNet

5.9.1. Описание системы

Система StrongNet предназначена для построения защищенных виртуальных частных сетей, позволяет создать защищенный канал для передачи данных между компьютерами в локальной сети или Интернет. Вся информация передается по этому каналу с использованием туннелирования в зашифрованном виде.

Система StrongNet основана на предварительном распределении ключей. Принцип работы следующий: все данные, передаваемые по защищенному каналу, шифруются с помощью симметричных алгоритмов шифрования. При этом ключи шифрования (сеансовые ключи) передаются между компьютерами при установлении защищенного соединения и шифруются с помощью асимметричного алгоритма шифрования RSA. Открытые и личные ключи, использующиеся при установлении соединения, хранятся в базе данных ключей. Распределение ключей между пользователями осуществляется системным администратором с помощью центра генерации ключей. Таким образом, пользователи сети к моменту установления соединения уже имеют все необходимые ключи.

Кроме защиты данных, передаваемых между двумя компьютерами по сети, StrongNet предоставляет функции персонального межсетевого экрана, который осуществляет фильтрацию входящих и исходящих IP-пакетов по определенным критериям.

Для работы с системой StrongNet необходимо сгенерировать и распределить между пользователями открытые и личные ключи. У каждого пользо-

вателя системы StrongNet есть набор ключей, в который входит его личный ключ и открытые ключи других пользователей системы, с которыми он обменивается данными через защищенные каналы. Набор ключей может храниться в файле либо на электронном ключе.

Программа «StrongNet Центр генерации ключей» предназначена для создания базы данных ключей, составления из них наборов, записываемых в файл или на электронный ключ, и распределения этих наборов между пользователями системы. Генерация ключей происходит один раз при создании базы данных.

5.9.2. Постановка задачи

Пусть существует некая организация, в которой в удаленных друг от друга офисах работают два пользователя. Требуется с использованием технологии виртуальных машин создать структуру сети, состоящую из двух виртуальных узлов, и установить защищенное соединение (рис. 5.31). Основная ОС имитирует работу компьютера стороннего наблюдателя и используется для анализа сетевого трафика.

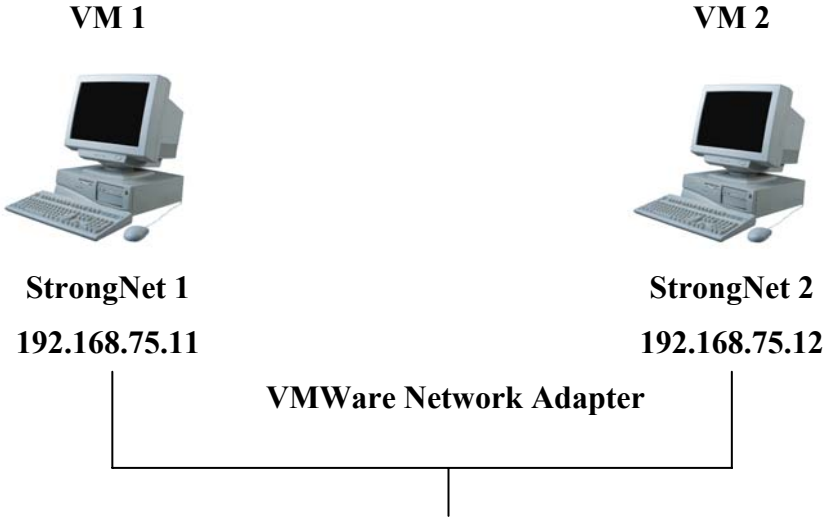


Рис. 5.31. Схема соединения виртуальных узлов

ВЫПОЛНИТЬ!

1. На рабочем месте открыть два образа ОС Windows 2000. Для каждого образа на вкладке Edit выбрать меню «Virtual Machine Settings» и установить размер потребляемой памяти (Guest size) — 64 MB, а тип сетевого подключения — «VMNet1 (Host Only)». Для обоих образов настроить виртуальные дисководы на единый файл. Запустить виртуальные ОС.
2. Настроить IP-адреса виртуальных машин (например, для первой ОС — 192.168.75.11, для второй ОС — 192.168.75.12). С помощью программ ipconfig и ping убедиться в правильной настройке сетевых адресов.

3. Осуществить захват трафика в основной ОС, убедиться в возможности анализа передаваемых ICMP-пакетов.
4. Установить систему StrongNet в обе виртуальные ОС, следуя указаниям установочной программы.

5.9.3. Генерация и распространение ключевой информации

Для успешной работы системы StrongNet необходимо создать базу данных ключей. Дистрибутив ключей для каждого сетевого узла размещен в файле с расширением «DST». Исходные ключи зашифрованы на парольной фразе и потому недоступны третьим лицам непосредственно из DST-файла. Чтобы создать базу данных ключей нужно запустить программу «Центр генерации ключей».

ВЫПОЛНИТЬ!

5. На одной из систем запустить программу «StrongNet Центр генерации ключей». В меню «Действие» выбрать пункт «Создать БД ключей». В появившемся окне установить количество генерируемых ключей — 2. Сохранить базу данных ключей.
6. Сгенерировать ключи для двух пользователей с учетом их дальнейшего взаимодействия. Для этого в правой части главного окна дважды щелкнуть левой кнопкой мыши на элементе «Пользователь 1». В появившемся окне «Создание КК» ввести имя, в списке «Все» выбрать Пользователь 2 и нажать кнопку « >> ». Нажать кнопку «Далее». В появившемся диалоговом окне «Запись КК» выбрать тип внешнего ключа — «Файл». Указать путь и имя файла, в котором будет храниться созданный набор ключей для Пользователя 1 и открытый ключ Пользователя 2. Аналогичные действия произвести для Пользователя 2.
7. После завершения работы мастера скопировать набор ключей Пользователя 2 на дискету (виртуальную дискету).

5.9.4. Настройка СЗИ StrongNet

ВЫПОЛНИТЬ!

8. В одной из виртуальных систем открыть главное окно программы StrongNet и нажать кнопку «Развернуть» (рис. 5.32).
9. На вкладке «Ключи» (рис. 5.33) выбрать тип внешнего ключа – файл. Указать файл с набором ключей Пользователя 2 и нажать кнопку «Загрузить». Переключатель «Загружать ключи при старте» поставить в состояние «Включено».
10. Во второй ОС аналогично загрузить набор ключей Пользователя 2, сохраненный на дискете.

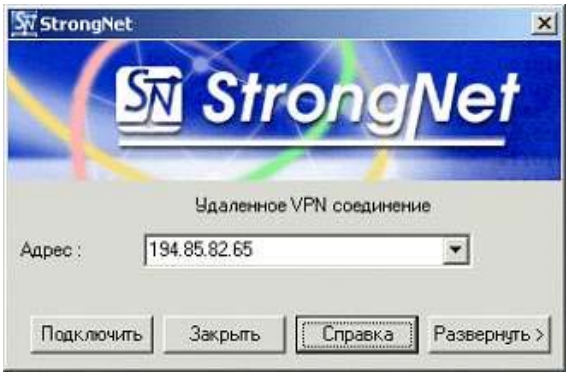


Рис. 5.32. Главное окно программы «StrongNet»

11. Используя вкладку «Настройки» (рис. 5.34) сделать так, чтобы сеансовый ключ в процессе работы защищенного соединения периодически менялся. Он может меняться по истечении некоторого промежутка времени, для этого переключатель «Генерировать ключ каждые» устанавливается во включенное состояние и в поле «Секунды» указывается длина соответствующего временного интервала. Чтобы защищенное соединение периодически проверялось на предмет активности, переключатель «Подтверждать соединение» устанавливается во включенное состояние и в поле «Секунды» указывается длина периода в секундах. Для вступления в силу сделанных изменений нажать кнопку «Применить».

Рис. 5.33. Загрузка ключевой информации

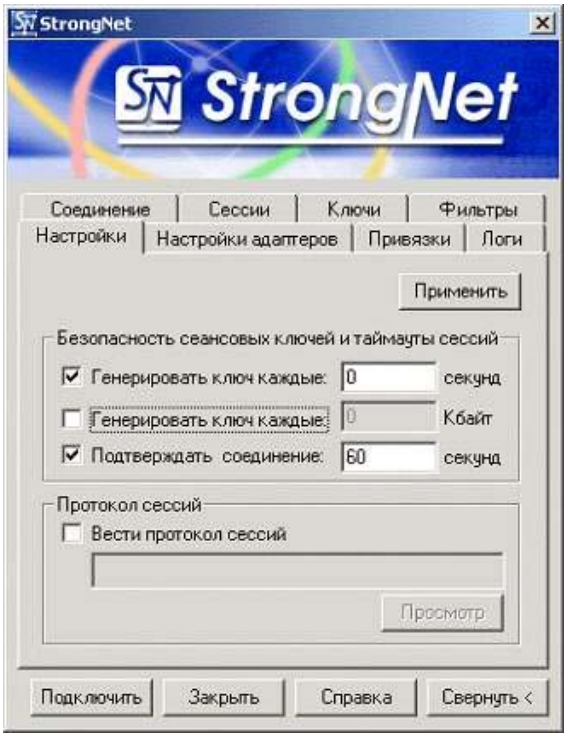


Рис. 5.34. Настройка параметров обновления ключевой информации

5.9.5. Установка защищенного соединения

ВЫПОЛНИТЬ!

12. Создать и проверить соединение между виртуальными ОС. Для этого на вкладке «Соединение» (рис. 5.35) одного из узлов указать IP-адрес второго узла и нажать кнопку «Подключить».
13. Убедиться в том, что соединение установлено. Для этого зайти на вкладку «Сессии» (рис. 5.36), подвести указатель мыши к соединению в списке, в контекстном меню выбрать пункт «Информация о соединении». Просмотреть информацию об установленном соединении.
14. Осуществить захват трафика в основной ОС, убедиться в том, что трафик является защищенным, проанализировать его тип.

Рис. 5.35. Настройка параметров соединения

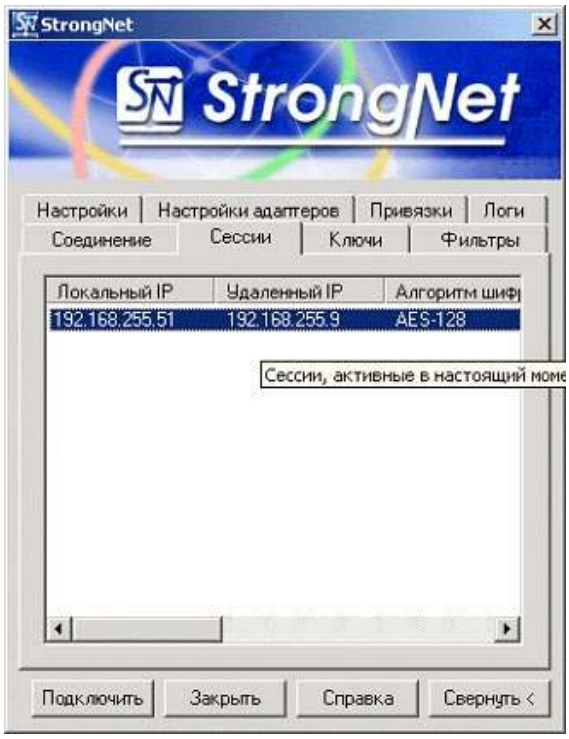


Рис. 5.36. Проверка информации о соединении

5.10. Защита на транспортном уровне

Для защиты на транспортном уровне применяются протоколы TLS и SSL. Особенностью защиты на данном уровне является независимость от прикладного уровня, однако чаще всего технология применяется для защиты данных, передаваемых по протоколу HTTP (режим HTTPS).

Подробнее рассмотрим функционирование протокола SSL. Протокол часто применяется для установки защищенного соединения, когда пользователь, обратившийся к web-серверу, передает или получает конфиденциальные сведения, например об объеме и стоимости покупки в Интернет-магазине, либо получает статистику своих соединений у Интернет-провайдера. В этом случае web-клиент, например Internet Explorer, автоматически переходит в защищенный режим, о чем свидетельствует пиктограмма «замок» в правой нижней части окна.

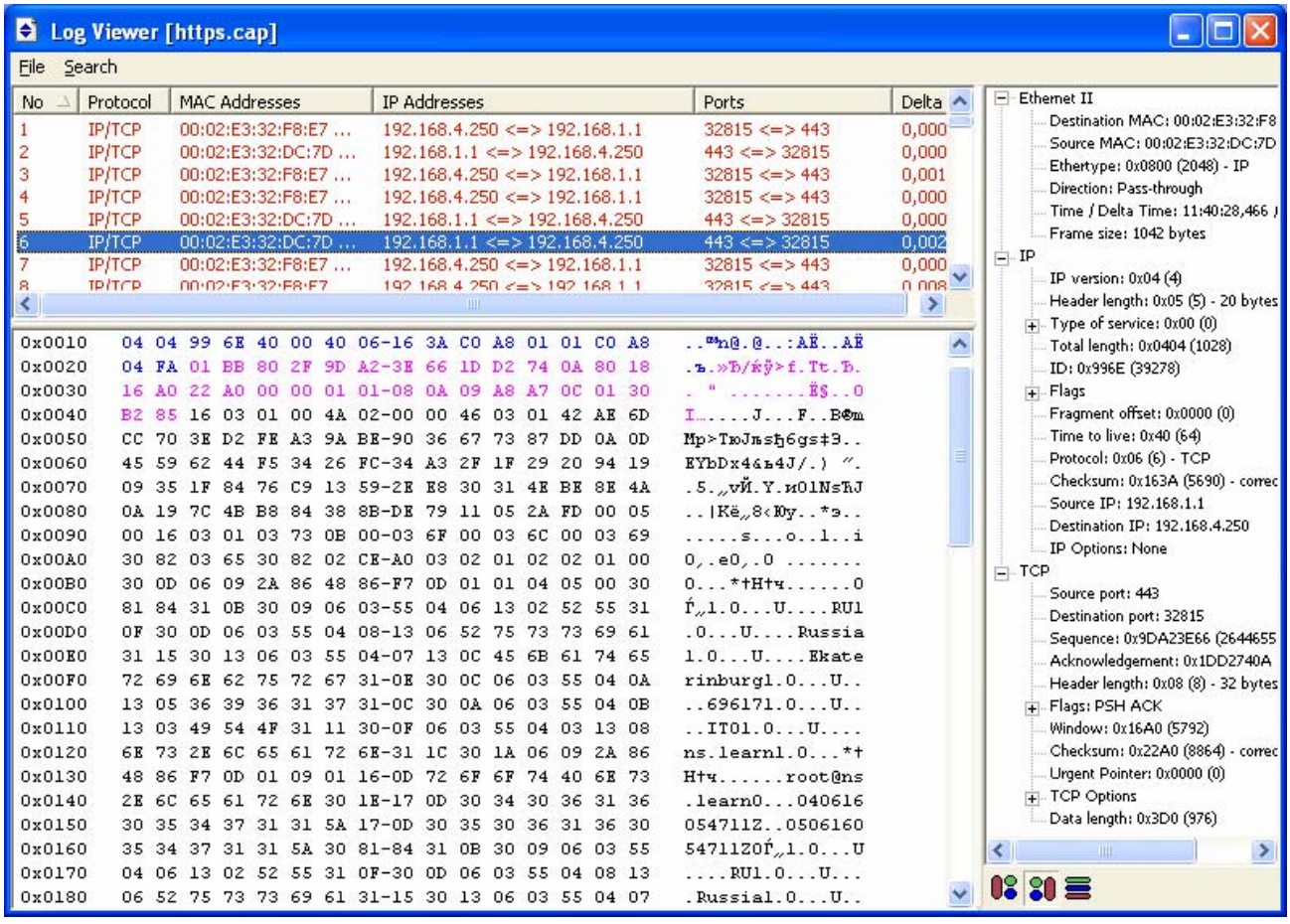


Рис. 5.37. Сетевой пакет с сертификатом открытого ключа сервера

Протокол SSL предусматривает функции аутентификации, шифрования данных и обеспечения целостности данных. Аутентификация осуществляется путем обмена цифровыми сертификатами при установлении соединения (сессии). Так как web-сервер обычно принимает запросы от произвольных клиен-

тов, то чаще всего аутентифицируется только сервер. Для шифрования данных применяется стандартный для VPN-соединений подход: для шифрования данных применяется симметричный сеансовый ключ. Обмен симметричными сеансовыми ключами происходит при установлении соединения, при передаче сеансовые ключи шифруются с помощью открытых ключей. Для обеспечения целостности к сообщению добавляется его хэш-код.

Рассмотрим этапы установки SSL-соединения. Сначала устанавливается стандартное TCP-соединение с портом сервера 443. Далее клиент передает сообщение «Client-Hello», в котором сообщает поддерживаемую им версию протокола SSL и случайную последовательность «Challenge_Data». В ответ сервер передает сообщение «Server-Hello», в котором указывает версию SSL, идентификатор соединения «Connection_id», список базовых шифров (протоколов) и сертификат сервера (подписанный открытый ключ).

Цель следующего сообщения, отправляемого клиентом (сообщение «Client_Master_Key»), — передача симметричного сеансового ключа, зашифрованного открытым ключом сервера. Таким образом, только сервер может расшифровать переданный симметричный ключ.

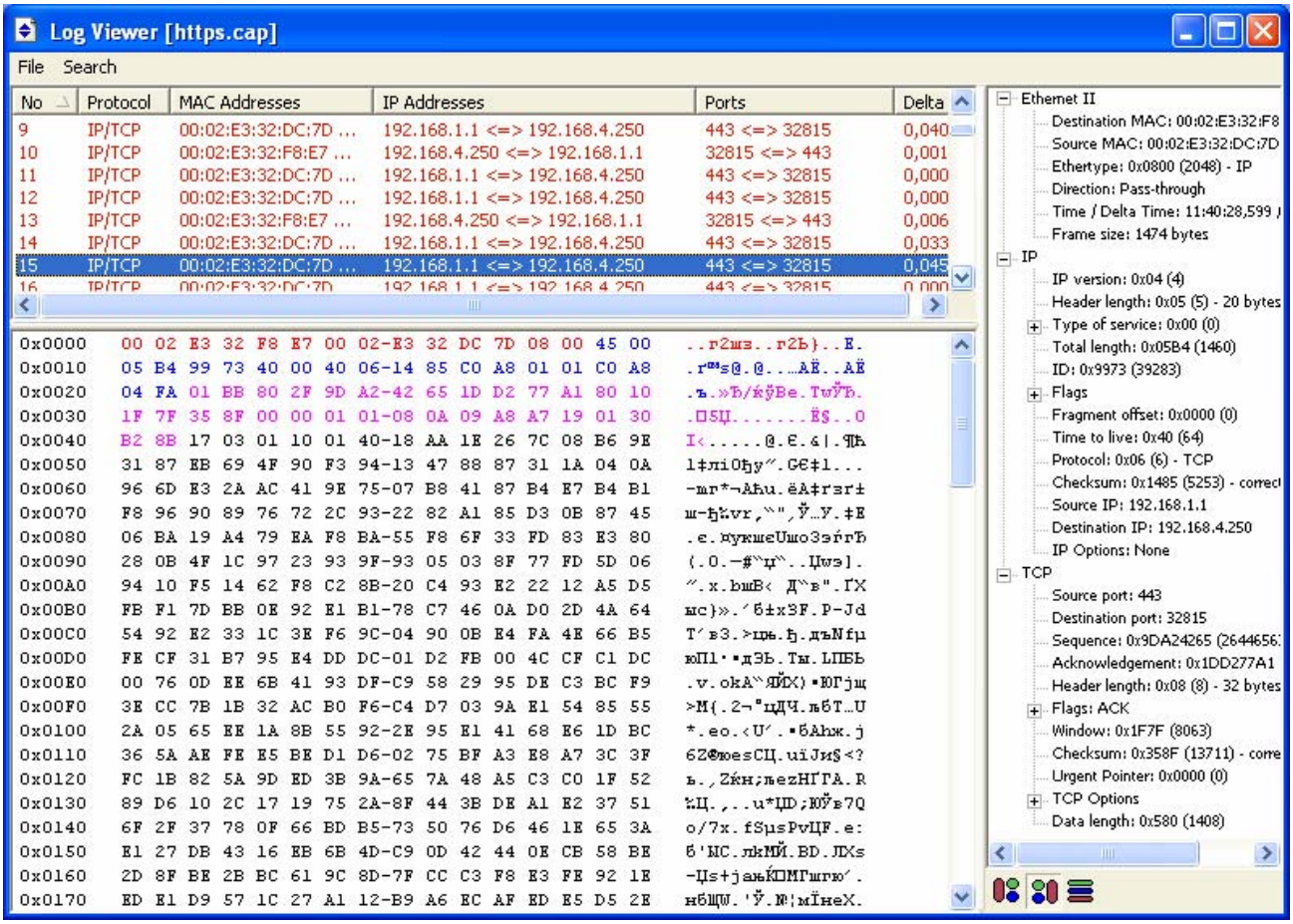


Рис. 5.38. Зашифрованный HTTP-трафик

Получив ключ, сервер зашифровывает этим ключом отправленную ранее последовательность «Challenge_Data» и передает ее в сообщении «Server-

Verify». Получив и расшифровав данное сообщение, клиент уверен, что сеансовый ключ получен и расшифрован сервером правильно. Для того чтобы сервер также мог убедиться в правильности полученного им сеансового ключа, клиент зашифровывает этим ключом идентификатор соединения «Connection_id», полученный от сервера, и передает его в сообщении «Client-Finished».

Таким образом, соединение установлено, сервер проверен, сеансовый ключ передан. Теперь весь трафик может передаваться в зашифрованном виде. Для внешнего наблюдателя виден трафик, идущий по 443 TCP-порту между двумя узлами с известными IP-адресами.

5.11. Организация VPN средствами протокола SSL в Windows Server 2003

Предположим, нам необходимо организовать защищенный обмен информацией между web-сервером и произвольным клиентом. Для организации воспользуемся ОС Windows Server 2003, в качестве web-сервера будем использовать встроенный в ОС компонент IIS (Internet Information Services).

Поставленная задача разбивается на три этапа: активизация IIS, генерация сертификата открытого ключа для web-севера и настройка SSL-соединения.

5.11.1. Активизация IIS

Компонент IIS по умолчанию в ОС Windows Server 2003 не установлен, целью данного этапа является его установка и проверка его функционирования с автоматически генерируемой web-страницей.

ВЫПОЛНИТЬ!

1. Установить компонент Internet Information Services (***Control Panel ⇒ Administrative Tools ⇒ Manage Your Server***).

В открывшемся диалоговом окне необходимо выбрать пункт «Add or remove a role», после чего ОС автоматически определит текущие сетевые настройки и отобразит диалоговое окно со списком возможных задач, выполняемых сервером (рис. 5.39). В этом списке необходимо выбрать пункт «Application servers (IIS, ASP.NET)». Установка дополнительных компонентов сервера FrontPage Extensions и ASP.NET не является обязательной, поэтому может быть пропущена. В результате указанных действий будут установлены компоненты, необходимые, в том числе для запуска web-сервера. Процесс установки может занять несколько минут, и для его успешного завершения понадобится дистрибутив Windows Server 2003.

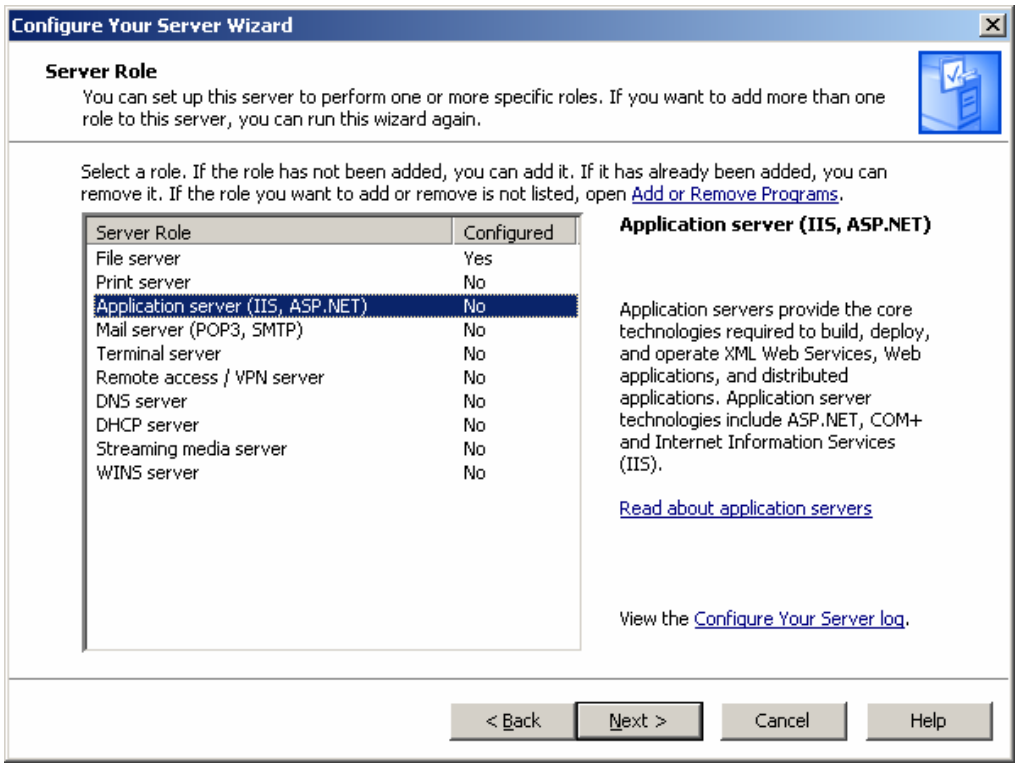


Рис. 5.39. Выбор пункта «Application servers (IIS, ASP.NET)»

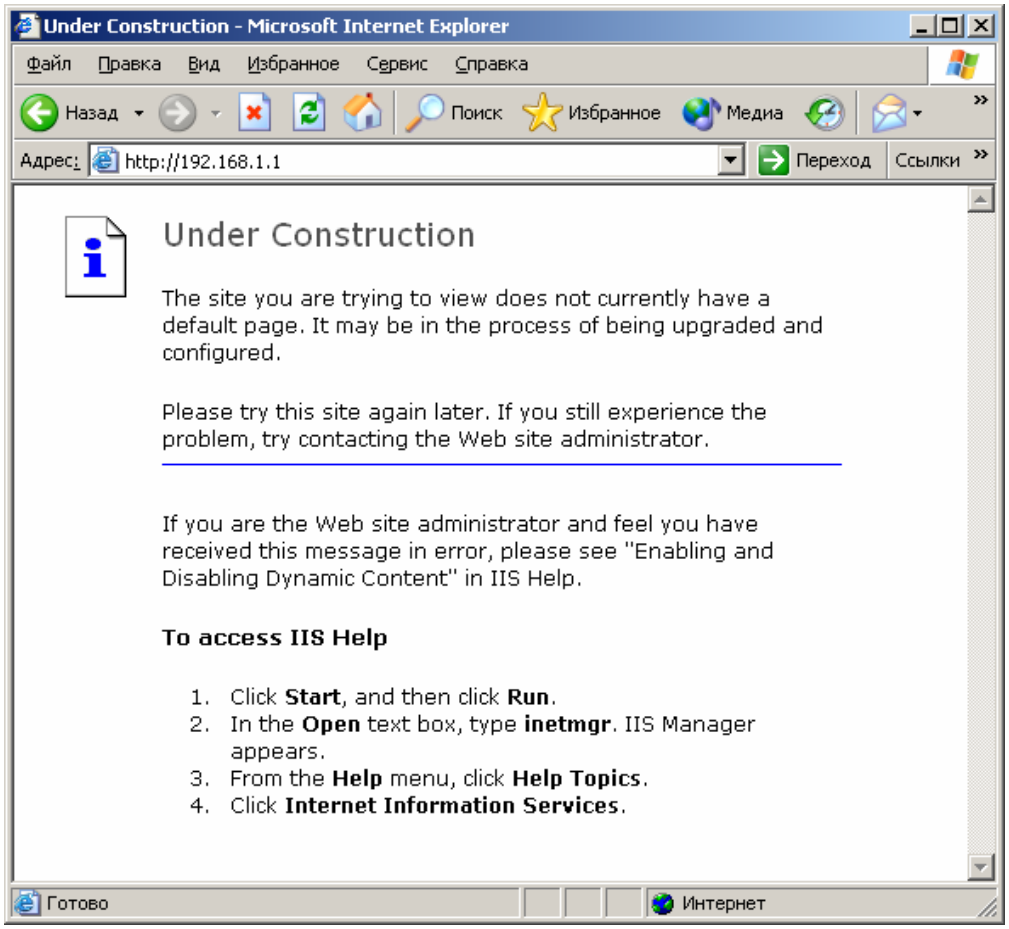


Рис. 5.40. Отображение web-страницы при обращении к серверу

После установки и перезагрузки web-сервер IIS автоматически запускается, в качестве стартовой используется автоматически генерируемая web-страница (рис. 5.40). Отображение этой страницы при обращении к серверу по его IP-адресу с указанием протокола HTTP говорит о том, что сервер отвечает на HTTP-запросы клиента (программы Internet Explorer). В результате выполнения данного этапа мы получили функционирующий web-сервер под управлением IIS.

ВЫПОЛНИТЬ!

2. Запустить анализатор сетевого трафика и просмотреть содержимое передаваемой между клиентом и сервером информации. Убедиться, что HTTP-запрос и HTTP-ответ передаются в открытом виде.

5.11.2. Генерация сертификата открытого ключа для web-сервера

Как указывалось выше, для шифрования передаваемой информации клиент и сервер должны получить общий ключ симметричного шифрования. В протоколе транспортного уровня данный ключ генерирует клиент и отправляет серверу. Однако для отправки ключа клиент применяет его зашифрование с использованием открытого ключа сервера, который должен быть известен клиенту. Для передачи открытого ключа применяется механизм сертификатов, цель которого обеспечить подлинность передаваемого открытого ключа. Таким образом, сервер должен иметь сертификат своего открытого ключа, который в общем случае должен быть подписан одним из доверенных центров сертификации.

В связи с тем, что мы организуем VPN-соединение в локальной сети учебного компьютерного класса, то в процессе работы самостоятельно сгенерируем сертификат открытого ключа и создадим его ЭЦП. Для этой цели нам понадобится Центр сертификации, для работы с которым необходимо добавить компонент Certificate Services (Службы сертификации). В процессе установки необходимо будет указать имя Центра сертификации (например, «Mycompany»), остальные настройки можно оставить по умолчанию.

После установки Центра сертификации необходимо от имени web-сервера выполнить запрос на получение нового сертификата.

ВЫПОЛНИТЬ!

3. Установить компонент Certificate Services (***Control Panel ⇒ Add or Remove Programs ⇒ Add/Remove Windows Components***, рис. 5.41).

4. Запустить оснастку Internet Information Services (IIS) Manager (***Control Panel ⇒ Administrative Tools ⇒ Internet Information Services (IIS) Manager***).

5. В разделе «Web Sites» (рис. 5.42) выбрать компонент «Default Web Site», щелкнуть на нем правой кнопкой и выбрать пункт «Properties» в контекстном меню. Далее выбрать вкладку «Directory Security» и нажать кнопку «Server Certificate…» в открывшемся окне (рис. 5.43).

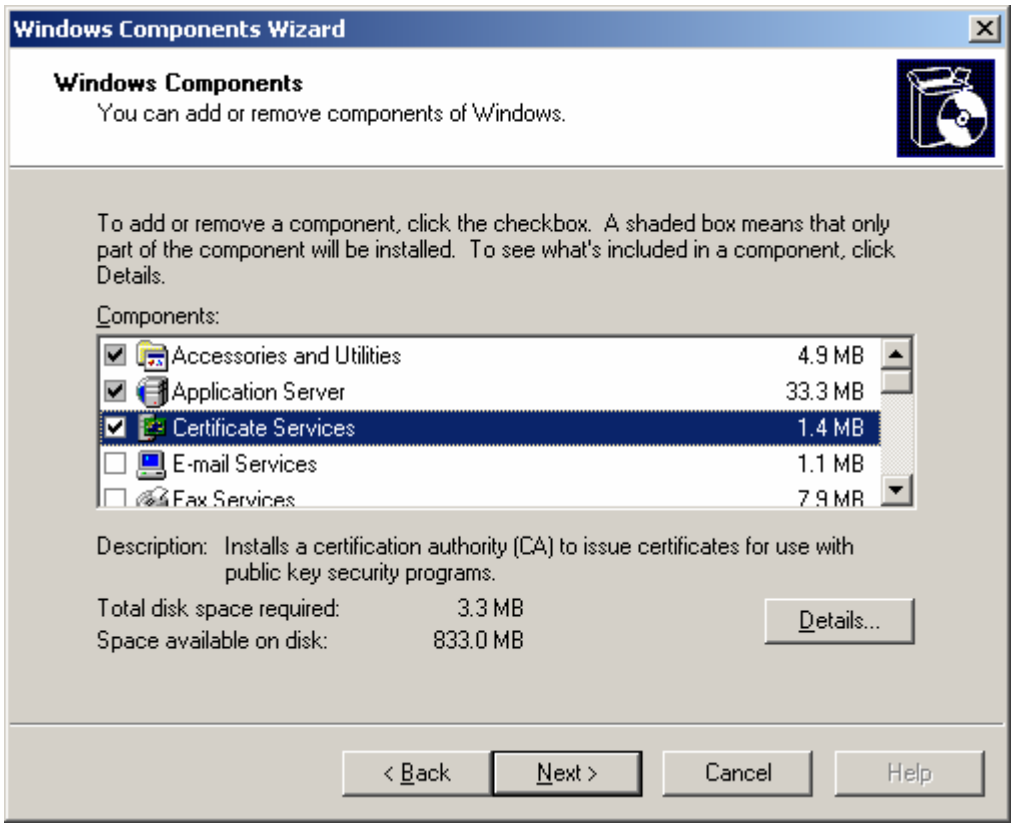


Рис. 5.41. Выбор компонента Certificate Services

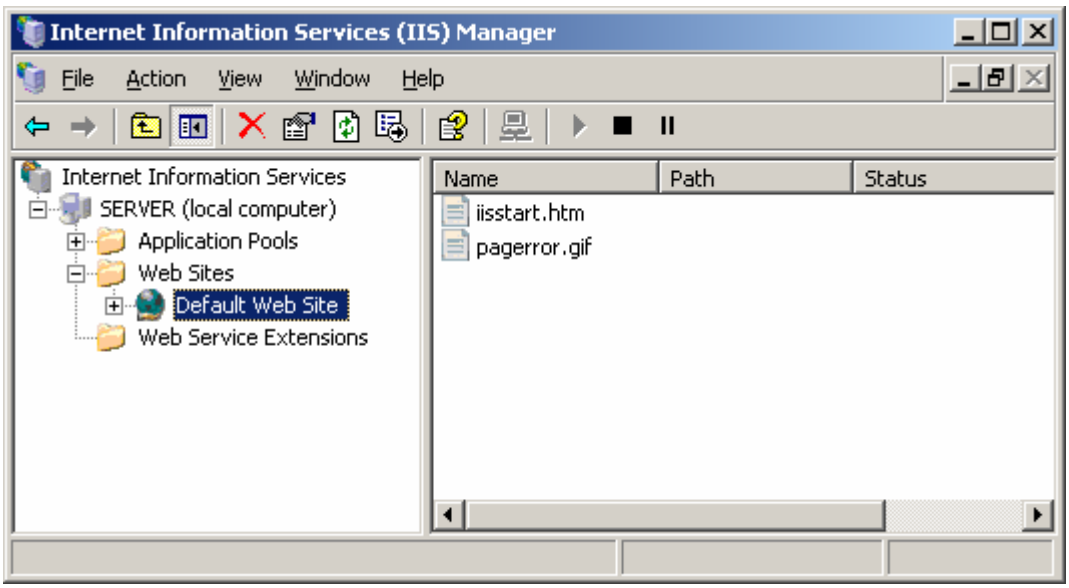


Рис. 5.42. Оснастка Internet Information Services (IIS) Manager

Будет запущен «мастер», позволяющий сформировать запрос на выдачу сертификата открытого ключа к Центру сертификации (Certification Authority). Необходимо выбрать опцию «Create a new certificate» (создать новый сертификат), а затем «Prepare the request, but send it later» (подготовить запрос, но отправить его позже). Будет предложено заполнить исходные данные, на основании которых будет выдан сертификат, в том числе наименования организации (Organization) и организационного подразделения (Organizational unit). Кроме того, необходимо указать доменное имя web-сайта (например, «www.mycompany.com») и его географическое местонахождение. Затем будет предложено сохранить текст запроса в виде текстового файла (рис. 5.44), содержимое которого необходимо отправить в Центр сертификации. Данный файл содержит открытый ключ web-сервера и заполненные сведения.

Так как Центром сертификации также является наш узел, то процесс отправки полученного текстового файла упрощается и заключается лишь в обработке данного файла с использованием оснастки Certification Authority (рис. 5.45). Результатом обработки будет создание файла-сертификата открытого ключа в формате X.509.

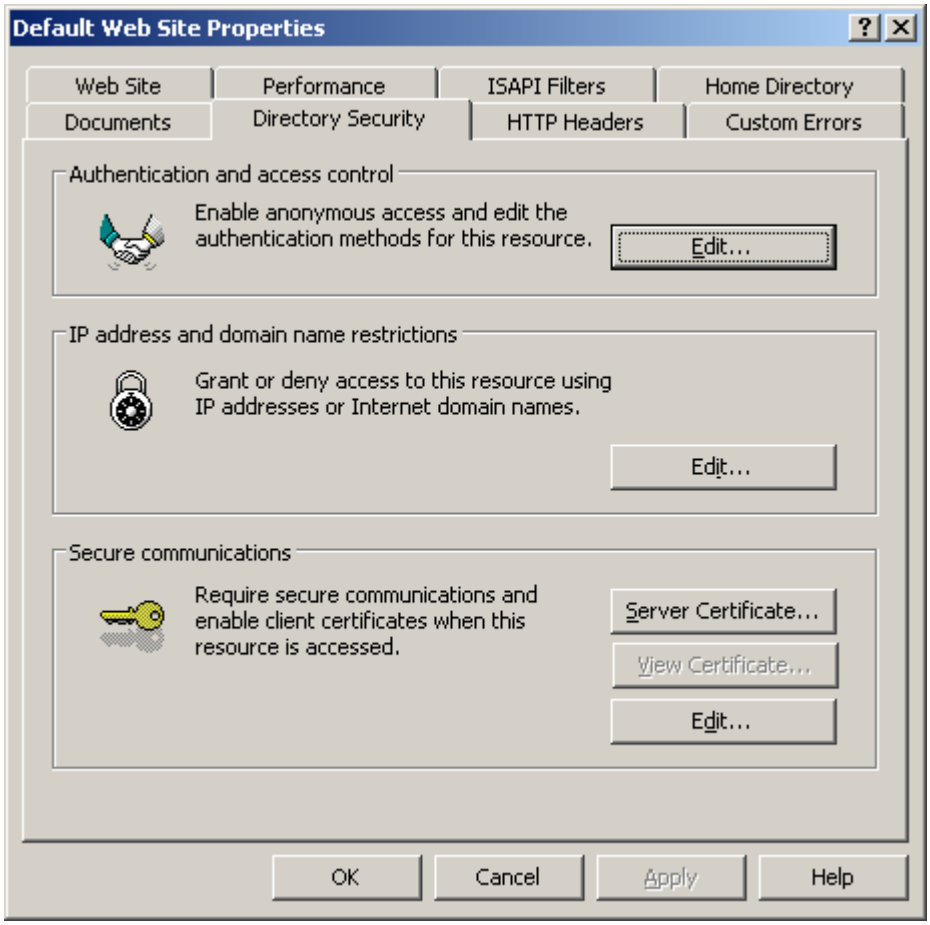


Рис. 5.43. Вкладка «Directory Security» окна свойств web-сайта

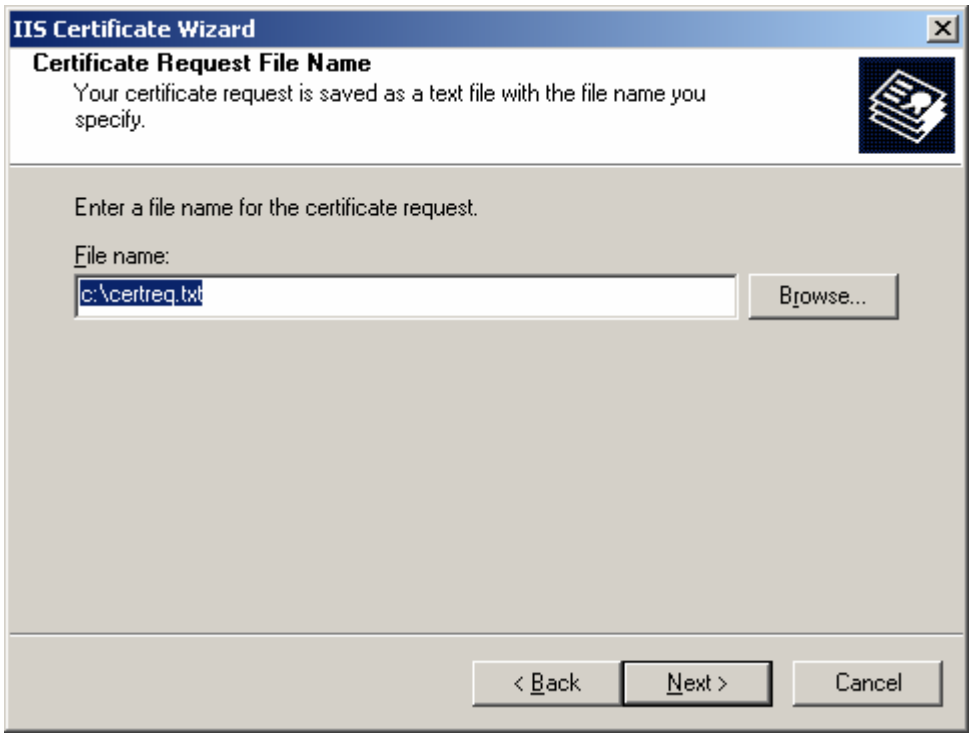


Рис. 5.44. Сохранение запроса сертификата

ВЫПОЛНИТЬ!

6. Запустить оснастку Certification Authority (***Control Panel ⇒ Administrative Tools ⇒ Certification Authority***).

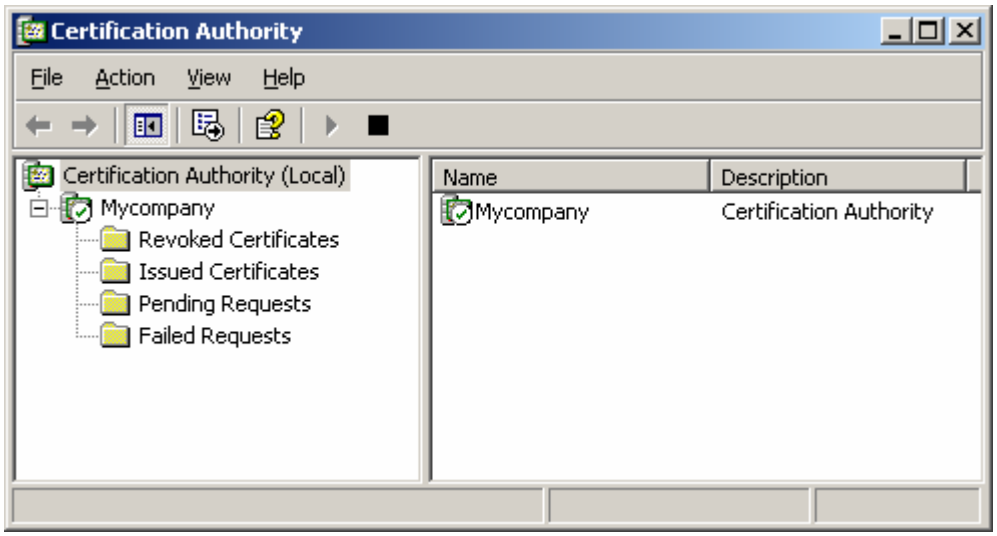


Рис. 5.45. Оснастка Certification Authority

Для добавления запроса необходимо из контекстного меню компонента «Mycompany» (в рассматриваемом примере) выбрать пункт ***All Tasks ⇒ Submit new request…*** Будет предложено выбрать файл с текстом запроса (он был создан ранее). Запрос добавляется в каталог «Pending Requests», чтобы обработать его, нужно из контекстного меню его записи выбрать пункт

All Tasks ⇒ Issue. Обработанный сертификат помещается в каталог «Issued Certificates». Чтобы сохранить сертификат в виде файла на жесткий диск, необходимо дважды щелкнуть на нем, перейти на вкладку «Details» и нажать кнопку «Copy to File…». Будет запущен «Мастер экспорта сертификатов», в котором нужно выбрать формат экспортируемого файла (выбрать «DER encoded binary X.509»), а также указать его имя (например, «c:\certnew.cer»).

ВЫПОЛНИТЬ!

7. Создать файл-сертификат открытого ключа.

Таким образом, сгенерирован файл-сертификат открытого ключа, который теперь может быть использован для организации VPN-соединения.

5.11.3. Настройка SSL-соединения

Настройка SSL-соединения заключается в установке на web-сервере сгенерированного сертификата и активизации SSL-соединения с указанием номера порта. Общепринятым номером порта для SSL-соединения является порт 443.

ВЫПОЛНИТЬ!

8. Запустить оснастку Internet Information Services (IIS) Manager и открыть окно свойств компонента «Default Web Site». На вкладке «Directory Security» нажать кнопку «Server Certificate…» и выбрать пункт «Process the pending request and install the certificate» (обработать находящийся на рассмотрении запрос и установить сертификат).

Будет предложено выбрать файл, содержащий указанный сертификат, а также указать SSL-порт, который будет использоваться web-сайтом (по умолчанию 443). В результате на данном web-сервере будет установлен сертификат открытого ключа.

Чтобы включить шифрование информации, передаваемой между клиентом и сервером, необходимо указать номер порта SSL на вкладке «Web Site», а затем на вкладке «Directory Security» нажать кнопку «Edit…» в разделе «Secure communications» и в открывшемся окне (рис. 5.46) установить отметку «Require secure channel (SSL)». Остальные настройки данного окна можно оставить без изменений. В частности, так как для web-сервера в общем случае не важно, имеется ли у клиента сертификат открытого ключа, то устанавливается значение «Игнорировать сертификаты клиентов» (Ignore client certificates).

Для активизации сделанных изменений необходимо нажать кнопку «Apply» (применить) в диалоговом окне «Default Web Site Properties».

После применения выполненных настроек web-сервер готов к осуществлению VPN-соединения с произвольным клиентом, обратившимся к ресур-

сам сервера с использованием режима HTTPS. Передаваемая информация будет зашифрована симметричным алгоритмом с 56- либо 128-битным ключом.

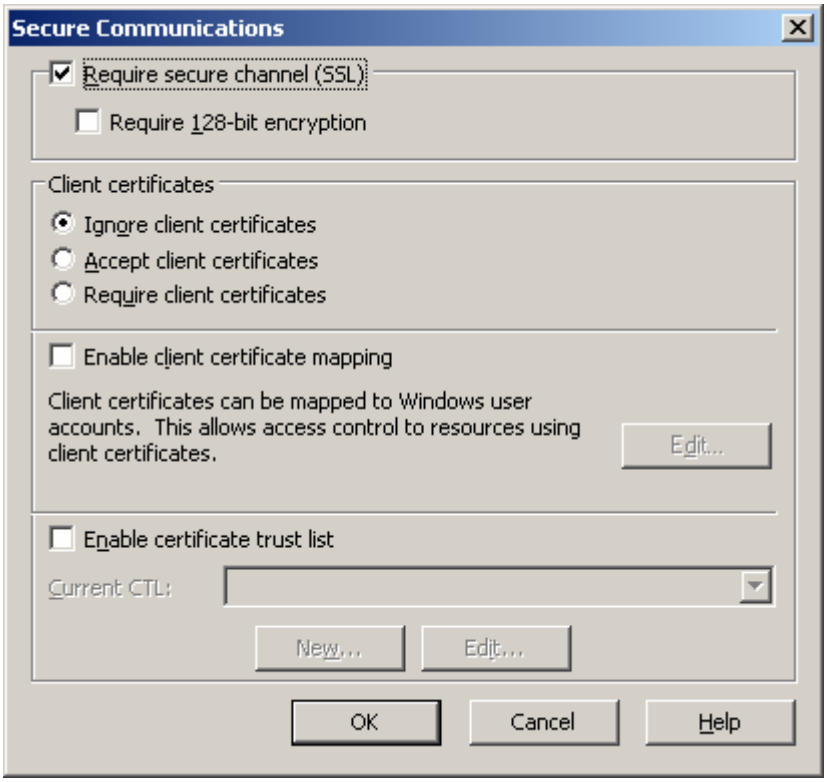


Рис. 5.46. Окно «Secure communications»

ВЫПОЛНИТЬ!

9. Запустить анализатор сетевого трафика и включить перехват пакетов.
10. Запустить в основной операционной системе программу «Internet Explorer» и обратиться к серверу по протоколу SSL, для этого ввести в строке адреса: «https://<IP-адрес_сервера>». Будет предложено согласиться с использованием сертификата, подписанного неизвестной удостоверяющей компанией, после чего должна быть открыта стартовая страница web-сайта (см. выше).
11. Выключить захват пакетов в анализаторе трафика. Проследить порядок установления соединения по протоколу SSL. Найти момент передачи текста сертификата от сервера к клиенту. Убедиться, что передача HTTP-запросов и HTTP-ответов происходит в зашифрованном виде.

5.12. Организация VPN прикладного уровня средствами протокола S/MIME и СКЗИ КриптоПро CSP

Предположим, нам необходимо организовать защищенный обмен почтовой информацией между двумя пользователями. В процессе организации воспользуемся двумя узлами. Один узел под управлением ОС Windows 2000 Professional будет выполнять роль почтового сервера, реализуемого сервером

Eserv, этот же узел будет являться рабочим местом первого пользователя (u1) для отправки почтовой корреспонденции с использованием программы Outlook Express. Второй узел под управлением ОС Windows Server 2003 будет рабочим местом второго пользователя (u2), дополнительно этот узел будет решать задачу по выдаче сертификатов открытых ключей. Шифрование почтовых сообщений будет осуществляться с помощью алгоритма ГОСТ 28147-89, реализуемого средствами СКЗИ КриптоПро CSP.

Поставленная задача разбивается на несколько этапов: организация почтового обмена без применения шифрования, активизация Web-сервера Internet Information Services (IIS) в ОС Windows Server 2003, установка СКЗИ КриптоПро CSP, установка Центра сертификации в ОС Windows Server 2003, получение сертификатов открытых ключей, организация защищенного обмена электронной почтой.

Для работы потребуются виртуальные образы систем Windows 2000 Professional (с установленным почтовым сервером Eserv) и Windows Server 2003, а также диски с дистрибутивами ОС Windows Server 2003 и СКЗИ КриптоПро CSP. Дополнительно требуется чистая дискета (может быть использована виртуальная дискета).

Предварительной операцией является настройка сетевого соединения виртуальных машин, имитирующих оба сетевых узла. Рекомендуется установить виртуальные сетевые адаптеры в режим Bridged и назначить сетевым узлам уникальные сетевые адреса, например, 192.168.*x*.1 и 192.168.*x*.2, где *x* — номер компьютера в учебном классе.

ВЫПОЛНИТЬ!

1. Открыть и запустить виртуальные образы, назначить IP-адреса, проверить установку связи с использованием команды ping.
2. С целью анализа сетевого трафика добавить в ОС Windows Server 2003 компонент Network Monitor (Control Panel ⇒ Add or Remove Programs ⇒ Add/Remove Windows Components ⇒ Management and Monitoring Tools ⇒ Network Monitor Tools).
3. Запустить установленную программу Network Monitor, убедиться в возможности захвата и анализа сетевого трафика.

5.12.1. Организация почтового обмена

Данный этап предусматривает настройку почтовых программ Outlook Express на двух узлах для отправки и получения электронной почты по протоколам SMTP и POP3 с сервера Eserv, установленного на узле с ОС Windows 2000.

4. Запустить сервер Eserv на узле с ОС Windows 2000, для чего выполнить командный файл Run.bat, находящийся на диске C: образа.

5. Проверить настройки сервера Eserv и убедиться в наличии учетных записей u1 и u2 (меню Общие настройки ⇒ Пользователи), установить пароли для указанных пользователей (задав и применив значение поля Password), убедиться в наличии настроек, указывающих в качестве локального домена адрес mail.ru (меню Почтовый сервер ⇒ SMTPсервер ⇒ Локальные домены).
6. Настроить почтовые программы Outlook Express на обоих узлах, создав учетные записи электронной почты для пользователей u1 (на ОС Windows 2000) и u2 (на ОС Windows Server 2003). В настройках указать адреса электронной почты, соответственно u1@mail.ru и u2@mail.ru, в качестве адресов SMTP- и POP3-серверов указать IP-адрес узла с ОС Windows 2000 (192.168.*x*.1).
7. Проверить функционирование почтового обмена путем отправки и получения почтовых сообщений. В процессе обмена в ОС Windows Server 2003 выполнить захват сетевого трафика, убедиться, что текст отправляемых и получаемых сообщений передается в открытом виде.
8. Убедиться в настройках учетных записей почты программы Outlook Express (Сервис ⇒ Учетные записи ⇒ Почта ⇒ Свойства ⇒ Безопасность) в наличии возможности шифрования с использованием алгоритмов DES, 3DES, RC2, а также в отсутствии сертификатов открытого ключа для подписи и шифрования.

5.12.2. Активизация IIS

Процесс активизации IIS подробно рассмотрен в разделе «Организация VPN средствами протокола SSL в ОС Windows Server 2003» учебного пособия. Приведем лишь перечень требуемых для выполнения команд.

ВЫПОЛНИТЬ!

9. Установить компонент Internet Information Services (Control Panel ⇒ Administrative Tools ⇒ Manage Your Server ⇒ Add or remove a role ⇒ Custom Configuration ⇒ Application servers (IIS, ASP.NET)).
10. Проверить функционирование Web-сервера, обратившись в ОС Windows 2000 с помощью программы Internet Explorer по адресу http://192.168.*x*.2.

5.12.3. Установка СКЗИ КриптоПро CSP

Установка СКЗИ КриптоПро CSP выполняется на обоих узлах с дистрибутивного диска. Применяется полнофункциональная версия СКЗИ без регистрации, что позволяет использовать ее в течение 30 дней. Инсталляция СКЗИ осуществляется стандартным образом. Настройки СКЗИ КриптоПро CSP доступны через Панель управления.

ВЫПОЛНИТЬ!

11. Установить СКЗИ КриптоПро CSP в ОС Windows 2000 и Windows Server 2003. Выполнить требуемую перезагрузку. После перезагрузки ОС Windows 2000 запустить сервер Eserv.
12. Проанализировать настройки учетных записей почты программы Outlook Express, выяснить изменения в разделе «Безопасность» свойств учетных записей, произошедшие после установки СКЗИ КриптоПро CSP.
13. Выполнить настройку считывателей программы КриптоПро CSP (Панель управления ⇒ КриптоПро CSP ⇒ Настроить считыватели). В ОС Windows 2000 в качестве считывателя использовать дисковод A:. В ОС Windows Server 2003 в качестве считывателя добавить реестр пользователя (User registry).

5.12.4. Установка Центра сертификации в ОС Windows Server 2003

В процессе выполнения данного пункта необходимо установить Службу сертификации отдельно стоящего корневого Центра сертификации. Установка Службы сертификации (Certificate Services) в ОС Windows Server 2003 подробно описана в разделе «Организация VPN средствами протокола SSL в ОС Windows Server 2003» учебного пособия.

ВЫПОЛНИТЬ!

14. Установить компонент Certificate Services (Control Panel ⇒ Add or Remove Programs ⇒ Add/Remove Windows Components). В процессе установки указать имя Центра сертификации (например, «Mycompany»), остальные настройки можно оставить по умолчанию.

5.12.5. Получение сертификатов открытых ключей

Одним из доступных способов получения сертификатов открытых ключей является обращение от имени каждого из пользователей к Центру сертификации через Web-интерфейс. Получение сертификата осуществляется в два этапа – сначала Пользователь обращается к Центру сертификации с запросом, а затем получает готовый сертификат и его инсталлирует в своей ОС. Между этими этапами администратор Центра сертификации должен осуществить обработку полученных запросов (издание сертификатов). Особенностью данного этапа будет получение сертификата, позволяющего работать с СКЗИ КриптоПро CSP. Сертификат должен быть сгенерирован для алгоритма ГОСТ Р 34.11-94.

ВЫПОЛНИТЬ!

15. От имени пользователя u1 в ОС Windows 2000 с помощью браузера Internet Explorer обратиться по адресу http://192.168.*x*.2/certsrv. Среди перечня задач выбрать пункт «Request a certificate».
16. При выборе типа запрашиваемого сертификата указать «advanced certificate request» и в следующем окне выбрать пункт «Create and submit a request to this CA».
17. В полученном информационном окне указать параметры пользователя, обязательными являются: имя пользователя (user1), точный адрес электронной почты (u1@mail.ru), тип сертификата «E-Mail Protection Certificate», настройки ключей (Key options) «CSP: Crypto-Pro Cryptographic Service Provider». Остальные параметры могут быть введены произвольно. Обратите внимание на то, что в параметрах алгоритма хэш-функции указан вариант «GOST R 34.11-94».
18. Для выполнения дальнейших действий разрешить выполнение элемента ActiveX, получаемого от сервера. В качестве носителя выбрать дисковод A: и поместить в него чистую дискету. Указать пароль, защищающий секретный ключ на носителе.
19. Выполнить аналогичный запрос от имени пользователя u2 в ОС Windows Server 2003, указав адрес почты u2@mail.ru. В качестве носителя выбрать «User registry». Указать пароль, защищающий секретный ключ в реестре.
20. В ОС Windows Server 2003 с помощью оснастки Certification Authority осуществить выдачу сертификатов. Для этого запустить оснастку Certification Authority (Control Panel ⇒ Administrative Tools ⇒ Certification Authority). В разделе «Pending Requests» найти запросы на получение сертификатов и обработать их, выбрав из контекстного меню записи запросов пункт «All Tasks ⇒ Issue».
21. Оснастку Certification Authority можно закрыть.
22. Повторно в каждой ОС с помощью Internet Explorer обратиться по адресу http://192.168.*x*.2/certsrv. В перечне задач выбрать «View the status of a pending certificate request».
23. Инсталлировать полученные сертификаты. В процессе инсталляции потребуется ввод пароля для доступа к секретным ключам на соответствующих носителях.
24. В каждой из ОС выполнить настройку почтовых учетных записей программы Outlook Express, подключив полученные сертификаты защиты электронной почты в разделе «Безопасность» свойств учетных записей.

5.12.6. Организация защищенного обмена электронной почтой

Для того чтобы два пользователя могли отправлять друг другу зашифрованные сообщения, они должны обменяться сертификатами открытых ключей. Самым простым способом обмена является отправка писем, подписанных

каждым из пользователей. После получения подписанного письма его отправитель должен быть добавлен в адресную книгу.

ВЫПОЛНИТЬ!

25. Создать, подписать (Сервис ⇒ Цифровая подпись) и отправить сообщение от имени пользователя u1, адресуемое пользователю u2.
26. Включить захват сетевого трафика. От имени пользователя u2 получить указанное сообщение. Остановить захват трафика.
27. Проанализировать захваченный сетевой обмен по протоколу POP3, убедиться в передаче открытого текста сообщения и его электронной цифровой подписи.
28. Открыть полученное сообщение, добавить пользователя u1 и его сертификат в адресную книгу пользователя u2.
29. Выполнить аналогичные действия, направив подписанное письмо от пользователя u2 к пользователю u1.
30. Осуществить обмен зашифрованными сообщениями, выполнив захват сетевого трафика в процессе отправки либо получения зашифрованного письма. Сделать вывод о том, какие из атрибутов письма передаются в зашифрованном виде.
31. Проанализировать свойства полученных зашифрованных писем и сделать вывод о том, какие алгоритмы и ключи применяются в процессе отправки зашифрованных сообщений, какие алгоритмы и ключи применяются при прочтении зашифрованных писем.
32. Дать ответ на вопрос, почему при формировании подписи требуется вводить пароль для доступа к ключевому носителю, а при отправке зашифрованного сообщения пароль вводить не требуется?
33. В ОС Windows 2000 с помощью программы-настройки СКЗИ КриптоПро CSP в окне «Сертификаты на носителе» выяснить состав сертификата пользователя. Указать назначение и содержимое полей: серийный номер, алгоритм подписи, поставщик, субъект, тип открытого ключа, идентификатор ключа субъекта, использование ключа.
34. Проанализировать свойства и содержимое ключевой дискеты. Проанализировать содержимое значений ключей реестра и их параметров в разделе реестра HKLM\Software\Crypto Pro\Setings\USERS\ Идентификатор_пользователя\Keys в ОС Windows Server 2003. Сделать вывод о необходимости организационной защиты ключевых носителей.
35. В ОС Windows 2000 с помощью оснастки «Сертификаты – текущий пользователь» (mmc ⇒ Консоль ⇒ Добавить/удалить оснастку ⇒ Сертификаты ⇒ Моей учетной записи) найти личный сертификат и сертификаты других пользователей.

6. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ТЕРМИНАЛЬНОГО ДОСТУПА

6.1. Общие сведения о технологии терминального доступа

Изначально терминальный режим работы появился и использовался на мэйнфреймах. Пользователи работали с терминалами, обеспечивавшими связь с терминальным сервером и отображение информации, полученной с главного компьютера. Все вычисления осуществлялись главным компьютером. На сегодняшний день суть терминального доступа не претерпела никаких идейных изменений. В современных схемах организации вычислительных процессов вместо специального аппаратного комплекса используются программы-клиенты, которые обеспечивают взаимодействие с сервером и отображение полученной от него информации. Всю вычислительную нагрузку также несет сервер.

Технология терминального доступа позволяет перенести вычислительные затраты с рабочих станций на сервер, решая ряд проблем:

- вся обработка данных выполняется на сервере, нет необходимости в сетевой передаче файлов, с сервера на рабочие станции передается лишь измененное содержимое информационных окон текстовых редакторов или СУБД, что упрощает защиту сетевого трафика и позволяет использовать в качестве рабочих станций практически любые компьютеры с любой ОС, в том числе бездисковые станции;
- отсутствует необходимость предоставлять пользователям потенциально опасный сетевой доступ к хранящимся на сервере файлам данных;
- магнитные, а также внешние носители, на которых может оказаться полная или частичная копия защищаемых файлов данных, расположены только на сервере и могут полностью контролироваться администратором.

Предполагается следующая схема использования технологии терминального доступа. На сервере устанавливается служба терминального доступа, развертываются приложения, необходимые для работы пользователей. Сервер терминального доступа не должен выполнять иных сетевых функций кроме обслуживания терминального режима, а именно, исключаются совместно предоставляемые сетевые ресурсы, включая принтеры. Перечень сетевых служб, функционирующих на сервере и доступных из сети, ограничивается только терминальной службой и, при необходимости, службой, обеспечивающей шифрование сетевого трафика.

На рабочих станциях пользователей устанавливается клиент терминала и настраивается на подключение к терминальному серверу. Запуск клиента терминала может осуществляться либо из основной ОС, установленной на компьютере пользователя, либо из ОС, запускаемой с внешнего носителя (дискеты или CD-ROM) или загружаемой с помощью сетевой карты удаленной загрузки.

В первом случае для работы с защищаемыми данными пользователь из основной ОС запускает клиента терминального доступа. При этом на компьютере могут быть установлены средства защиты информации от несанкционированного доступа. Преимуществом данного способа является возможность организации дополнительной защиты (шифрования) сетевого трафика путем использования протокола IPSec (в ОС Windows XP) либо специализированных СЗИ.

Во втором случае пользователь для работы с защищаемыми данными загружает компьютер со специально подготовленного носителя (CD-ROM или дискеты), на который записывается ОС Linux с клиентом терминального сервера. Может быть применена бездисковая станция, загружаемая с сервера при помощи сетевого адаптера, разрешающего удаленную загрузку. Отрицательным свойством этого решения является невозможность применения дополнительных средств шифрования трафика. Причина заключается в том, что не известны сертифицированные средства защиты информации, загружаемые с внешнего носителя или по сети.

Для обработки защищаемых данных пользователь запускает программу-клиента терминала, регистрируется на терминальном сервере с использованием рядовой учетной записи. Особенностью настройки терминального сервера является установка ряда запретов для пользователей, наиболее важным из которых является запрет использования совместного буфера обмена. Благодаря данному запрету решается проблема несанкционированного копирования защищаемых данных на носители рабочих станций. Пользователь терминала может выделить и скопировать в буфер обмена терминальной Windows как файл с данными, так и содержимое информационного окна. Однако операцию вставки можно выполнить только в окне терминального сервера. В окне рабочей станции возможность вставки из буфера будет заблокирована.

Таким образом, копирование всей защищаемой информации либо ее части может быть осуществлено лишь на носители, физически подключенные к серверу. Это накладывает некоторые ограничения на возможность экспорта/импорта данных, так как операции экспорта и импорта также осуществляются только через носители, установленные на сервере. Основным преимуществом является то, что все носители, включая внешние, на которых может оказаться полная или частичная копия защищаемых данных, расположены только на сервере под контролем администратора. Это упрощает централизованный антивирусный контроль и блокирует возможность появления вредоносных программ.

Проблема образования технологического «мусора» на рабочих станциях также решается автоматически. Для каждого терминального сеанса на сервере создается временный каталог. Если установлены соответствующие настройки, то по окончании сеанса этот каталог будет удален. Таким образом, технологический «мусор» остается лишь на носителях терминального сервера.

Проблема передачи открытого сетевого трафика решается прежде всего тем, что в технологии терминального доступа вся обработка защищаемых данных выполняется на сервере, а на рабочие станции передается лишь измененное содержимое информационных окон соответствующих приложений. Кроме того, возможно шифрование трафика средствами терминального сервера. Терминальный сервер поддерживает несколько уровней безопасности, каждый из которых определяет направление шифруемого трафика и длину ключа, используемого при шифровании.

В состав Windows Server 2003 включена служба Microsoft Terminal Services (MSTS) [18]. Она предоставляет возможность либо удаленно администрировать сервер, либо превратить его в сервер приложений (терминальный сервер). Кроме того, существует надстройка над данной службой, разработанная компанией Citrix, которая вводит ряд дополнительных возможностей и увеличивает число поддерживаемых платформ.

Следует отметить, что сама реализация MSTS не свободна от недостатков, которые потенциально могут быть применены злоумышленниками для нарушения безопасности данных. Так как все пользователи, подключающиеся к серверу в терминальном режиме, по сути, осуществляют интерактивный вход в систему, то они могут зарегистрироваться в системе с консоли сервера. Следовательно, использование терминального сервера предъявляет повышенные требования к администрированию и к выполнению необходимых настроек безопасности применяемого программного обеспечения.

Безопасность режима терминального доступа обеспечивается совокупностью настроек ОС Windows Server 2003, серверной части MSTS и протокола терминального доступа — RDP. В каждом из этих компонентов реализованы различные механизмы защиты, но в то же время каждый компонент имеет собственные уязвимости, которые могут быть использованы злоумышленниками.

6.2. Обеспечение безопасности ОС Windows Server 2003

Основными группами уязвимостей ОС Windows Server 2003, которые представляются актуальными для защиты в терминальном режиме, являются:

- возможность сетевого доступа к обрабатываемой сервером информации;
- возможность расширения полномочий при осуществлении локального доступа.

6.2.1. Ограничение возможности сетевого доступа

Возможность сетевого доступа реализуется благодаря излишнему количеству сетевых сервисов, по умолчанию предоставляемых сервером ОС Windows Server 2003. Упорядоченный по наименованию перечень сетевых сервисов, функционирующих в ОС по умолчанию, приведен в табл. 6.1. Полный перечень сетевых сервисов ОС Windows Server 2003 приведен в Приложении 2.

Таблица 6.1

Перечень сетевых сервисов ОС Windows Server 2003

Порт	Тип	Протокол	Наименование системной службы
137	TCP	NetBIOS Name Resolution	Computer Browser
137	UDP	NetBIOS Name Resolution	Computer Browser
138	UDP	NetBIOS Datagram Service	Computer Browser
139	TCP	NetBIOS Session Service	Computer Browser
139	TCP	NetBIOS Session Service	Fax Service
445	TCP	SMB	Fax Service
445	UDP	SMB	Fax Service
500	UDP	IPSec ISAKMP	IPSec Services
138	UDP	NetBIOS Datagram Service	License Logging Service
139	TCP	NetBIOS Session Service	License Logging Service
445	TCP	SMB	License Logging Service
445	UDP	SMB	License Logging Service
138	UDP	NetBIOS Datagram Service	Messenger
137	TCP	NetBIOS Name Resolution	Net Logon
137	UDP	NetBIOS Name Resolution	Net Logon
138	UDP	NetBIOS Datagram Service	Net Logon
139	TCP	NetBIOS Session Service	Net Logon
3389	TCP	Terminal Services	NetMeeting Remote Desktop Sharing
139	TCP	NetBIOS Session Service	Performance Logs and Alerts
139	TCP	NetBIOS Session Service	Print Spooler
445	TCP	SMB	Print Spooler
445	UDP	SMB	Print Spooler
135	TCP	RPC	Remote Procedure Call
139	TCP	NetBIOS Session Service	Remote Procedure Call Locator
445	TCP	SMB	Remote Procedure Call Locator
445	UDP	SMB	Remote Procedure Call Locator
4500	UDP	NAT-T	Routing and Remote Access
137	TCP	NetBIOS Name Resolution	Server
137	UDP	NetBIOS Name Resolution	Server
138	UDP	NetBIOS Datagram Service	Server
139	TCP	NetBIOS Session Service	Server
445	TCP	SMB	Server
445	UDP	SMB	Server
1900	UDP	SSDP	SSDP Discovery Service
5000	TCP	SSDP legacy event notification	SSDP Discovery Service
3389	TCP	Terminal Services	Terminal Services
137	TCP	NetBIOS Name Resolution	Windows Internet Name Service
137	UDP	NetBIOS Name Resolution	Windows Internet Name Service
123	UDP	NTP	Windows Time
123	UDP	SNTP	Windows Time

Как известно, перечень открытых сетевых портов может быть получен командой netatat –aon. Результаты выполнения этой команды для ОС Windows Server 2003 приведены на рис. 6.1.

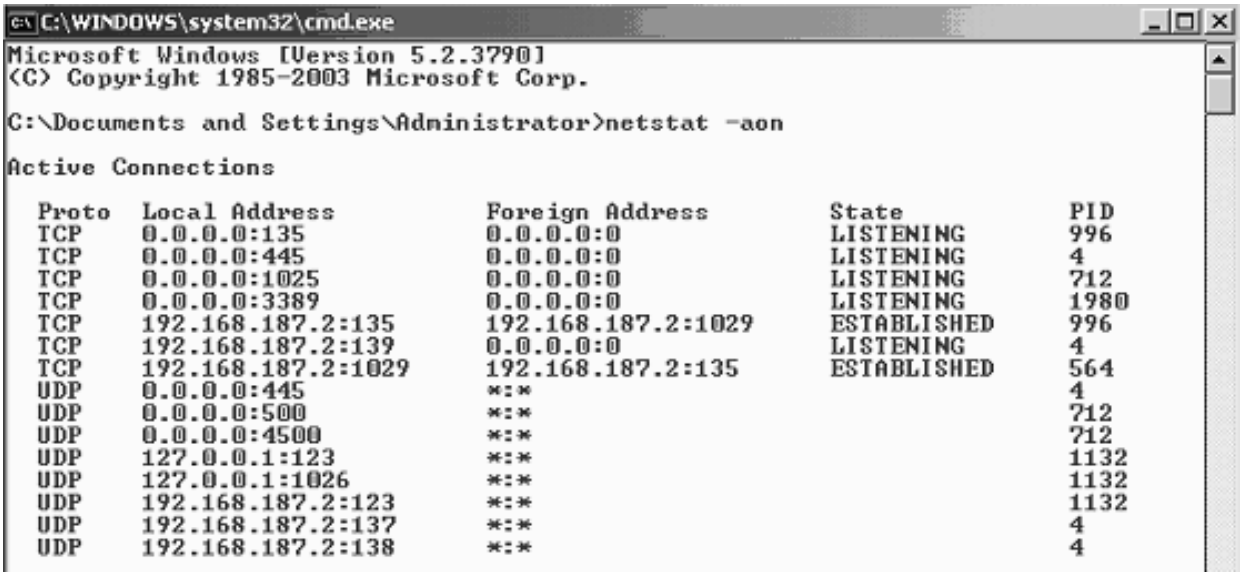


Рис. 6.1. Результаты выполнения команды netstat

Согласно приведенной выше схеме организации доступа к защищаемым данным, сервер терминального доступа должен выполнять только задачу обеспечения службы MSTS. Сервер терминального доступа, в частности, не должен выполнять функции контроллера домена. Таким образом, из перечня функционирующих по умолчанию сетевых служб должны быть исключены все службы кроме службы Terminal Services, TCP-порт 3389.

Для обеспечения защиты сетевого трафика дополнительно может понадобиться функционирование службы IPSec Services (UDP-порт 500), а также иных служб, используемых специализированными СЗИ.

Исключение служб может быть осуществлено двумя способами: остановом службы в оснастке Services (рис. 6.2) либо запрещением доступа к службе с применением межсетевого экранирования.

Способом останова должны быть исключены службы: Computer Browser, Server, Windows Internet Name Service, Net Logon, Messenger, License Logging Service, Fax Service, Performance Logs and Alerts, Print Spooler, Remote Procedure Call Locator, Routing and Remote Access, SSDP Discovery Service, Windows Time. Отключение службы Remote Procedure Call, функционирующей на 135 TCP-порту, приводит к неработоспособности узла, поэтому запрет доступа к этому порту будет производиться с использованием технологии межсетевого экранирования.

Путем модификации настройки сетевых соединений (рис. 6.3) и отключения службы NetBIOS через TCP/IP запрещаются службы, использующие порт TCP 139.

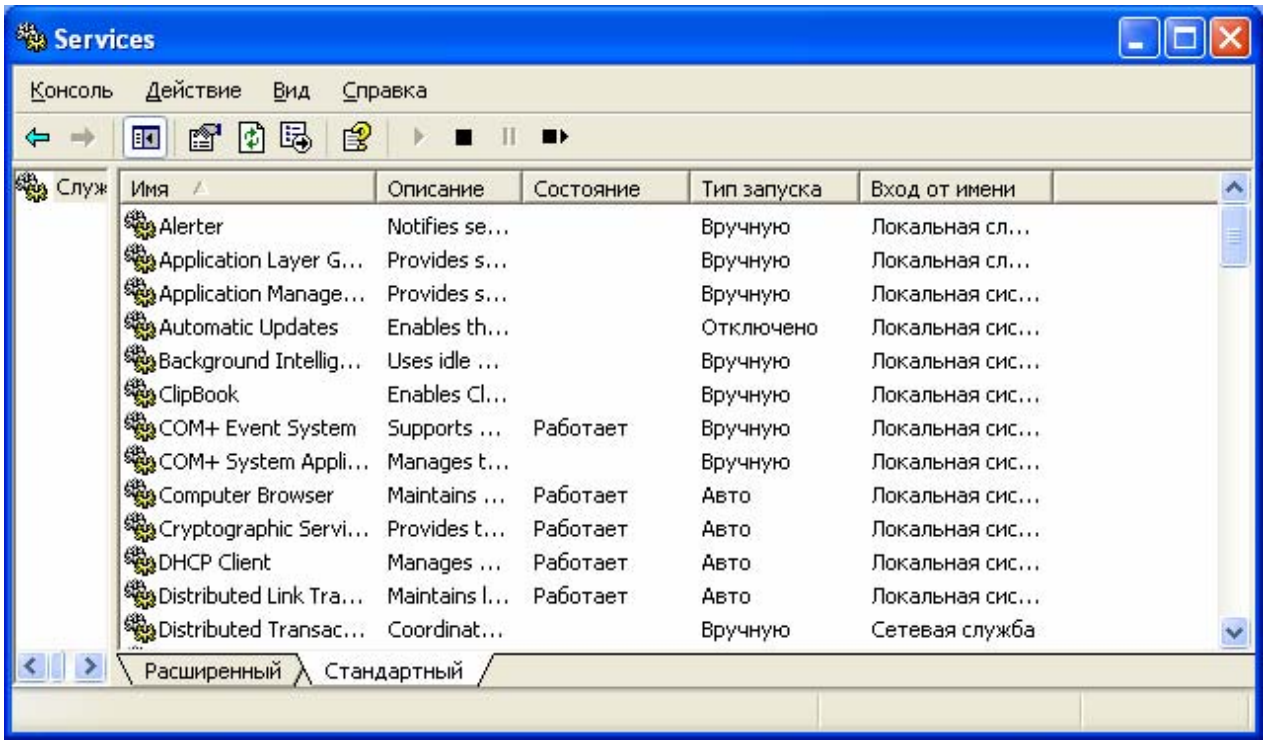


Рис. 6.2. Окно перечня служб ОС Windows

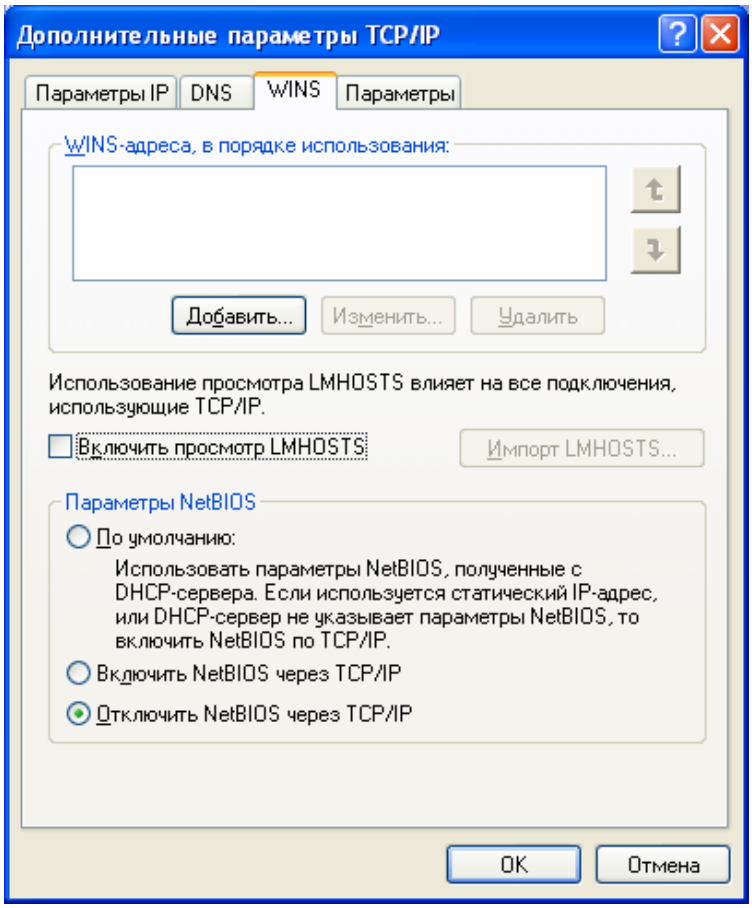


Рис. 6.3. Окно настройки свойств протокола TCP/IP

В итоге после отключения вышеуказанных служб открытыми остаются порты TCP: 135, 445, 1025, 3389; UDP: 445, 500, 4500 (рис. 6.4). Запретить данные порты, являющиеся, безусловно, опасными с точки зрения осуществления несанкционированного доступа, возможно лишь путем межсетевого экранирования.

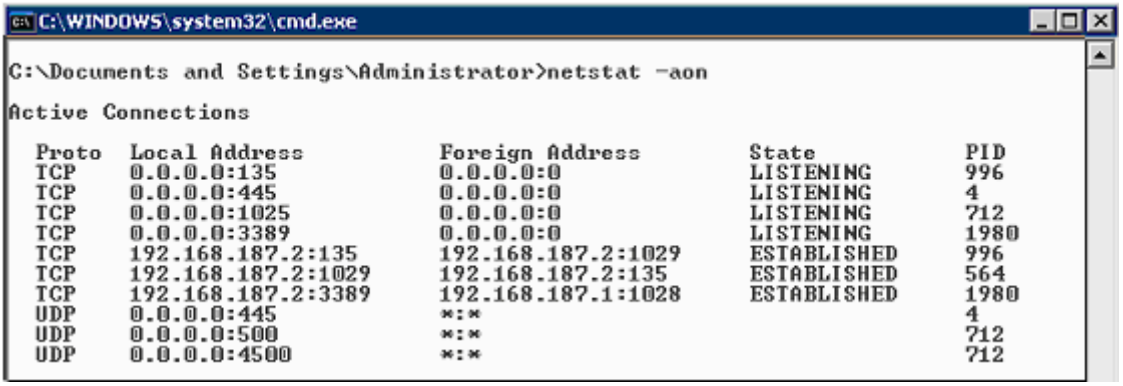


Рис. 6.4. Перечень открытых портов, остающихся после останова сетевых служб

Технология межсетевого экранирования в ОС Windows Server 2003 может быть применена с использованием:

- настроек протокола IPSec;
- штатного межсетевого экрана «Брандмауэр Windows»;
- дополнительного межсетевого экрана, реализованного сертифицированным средством защиты информации.

В случае использования настроек протокола IPSec ограничение доступа к портам может быть осуществлено либо через параметры фильтрации протокола TCP/IP в окне дополнительных параметров свойств протокола TCP/IP (рис. 6.5), либо путем создания шаблона безопасности для IP-протокола в окне «Локальная политика безопасности» (рис. 6.6).

При использовании параметров фильтрации протокола TCP/IP необходимо запретить все порты, кроме порта TCP 3389, отвечающего за функционирование MSTS (рис. 6.7). Однако в этом случае не удается запретить функционирование протокола ICMP, т. е. невозможно исключить входящие ICMP-запросы и исходящие ICMP-ответы.

При использовании шаблона безопасности для IP-протокола создается новая политика, разрешающая функционирование TCP-порта 3389 и запрещающая функционирование иных IP-протоколов, включая ICMP.

В случае использования штатного межсетевого экрана «Брандмауэр Windows» (рис. 6.8) необходимо также запретить использование всех портов, за исключением TCP-порта 3389 (рис. 6.10). Указанный порт именуется в программе «Дистанционным управлением рабочим столом» (рис. 6.9). Дополнительно следует отключить функционирование протокола ICMP (рис. 6.11).

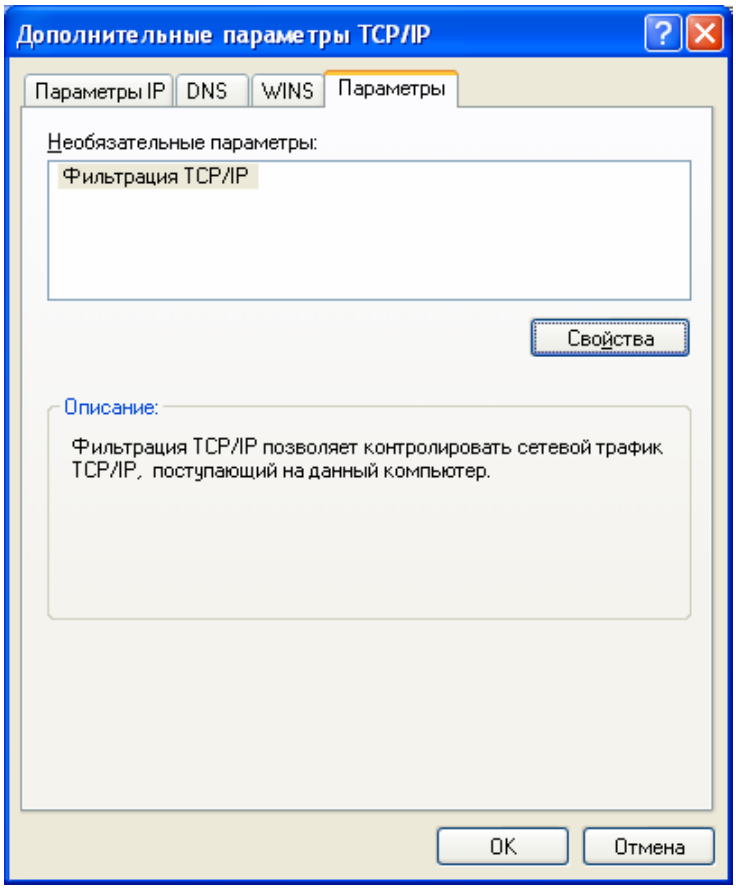


Рис. 6.5. Окно дополнительных параметров свойств протокола TCP/IP

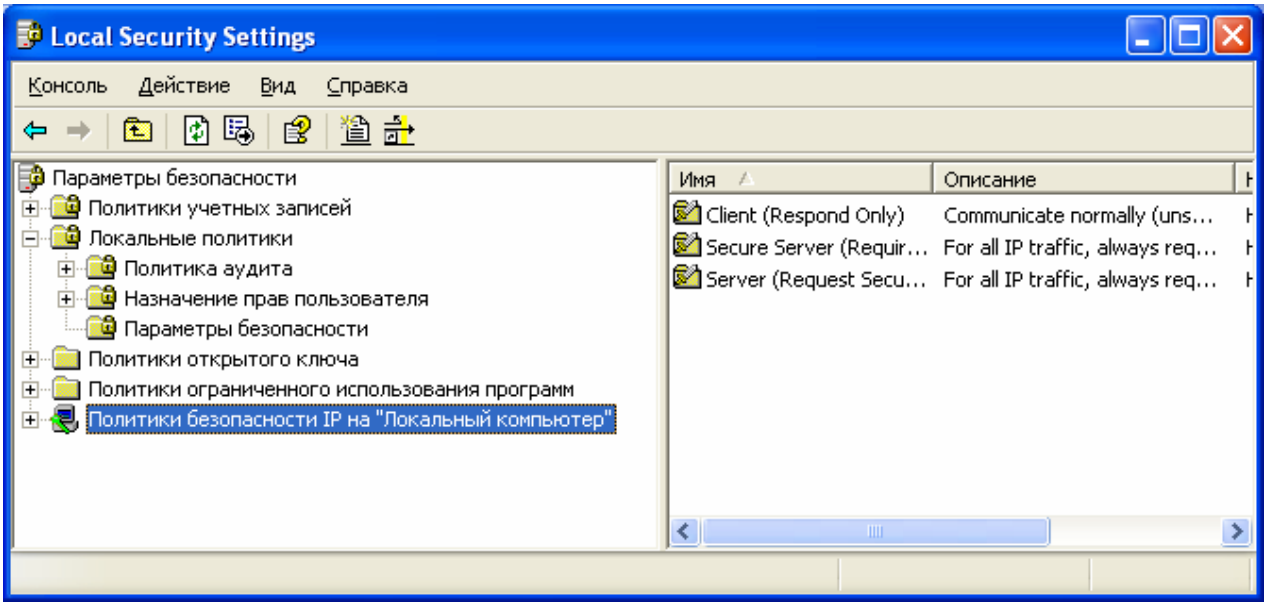


Рис. 6.6. Окно «Локальная политика безопасности»

Применение специализированных сертифицированных средств защиты, реализующих средства межсетевого экранирования, в частности СЗИ VipNet, является, безусловно, более предпочтительным, чем использование штатных средств ОС Windows.

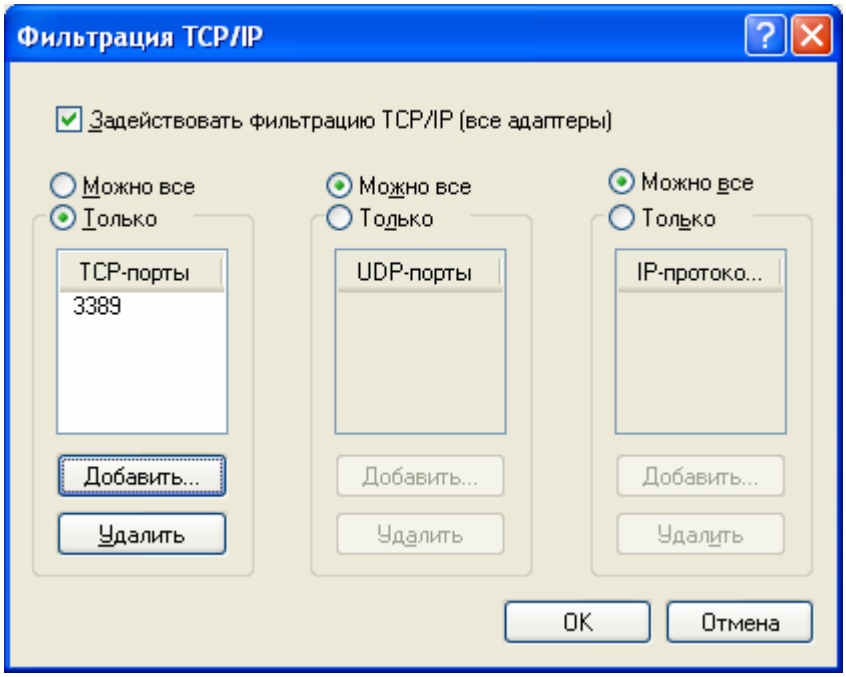


Рис. 6.7. Установка фильтра, разрешающего соединение

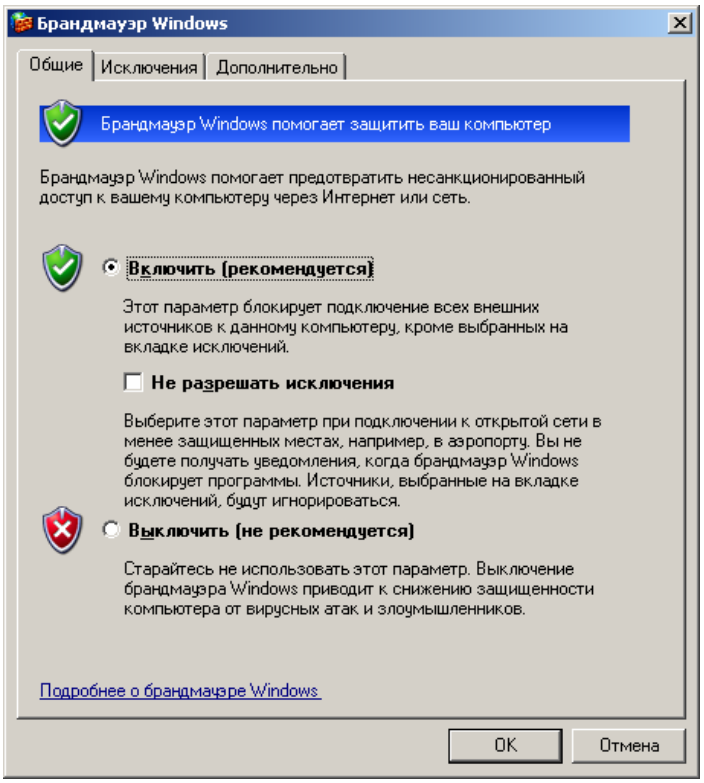


Рис. 6.8. Окно штатного межсетевого экрана «Брандмауэр Windows»

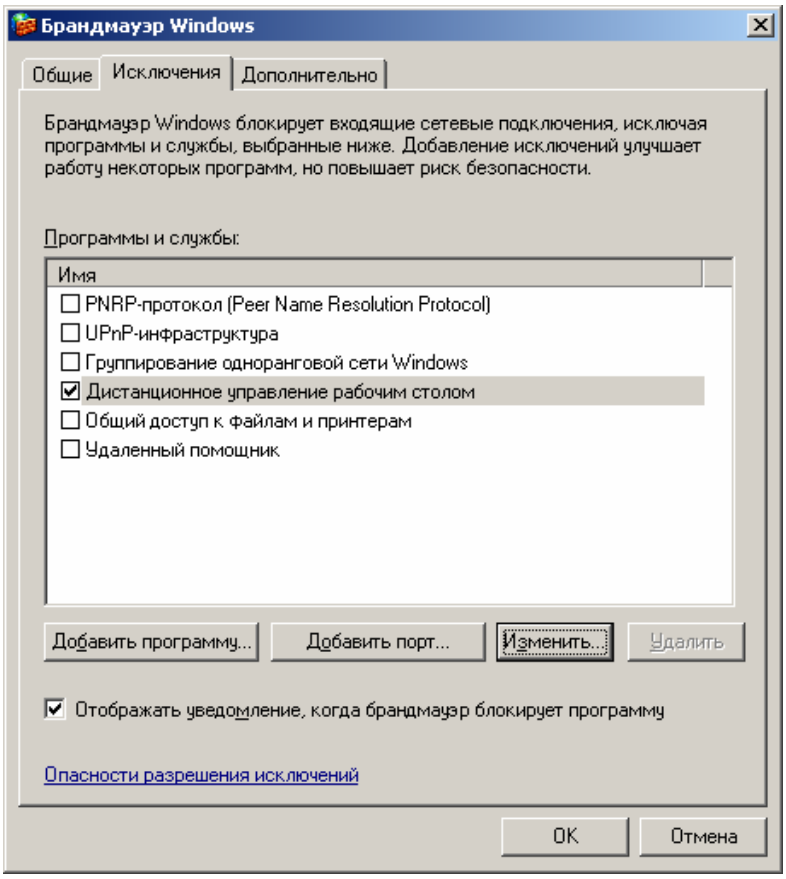


Рис. 6.9. Окно настроек «Брандмауэра Windows»

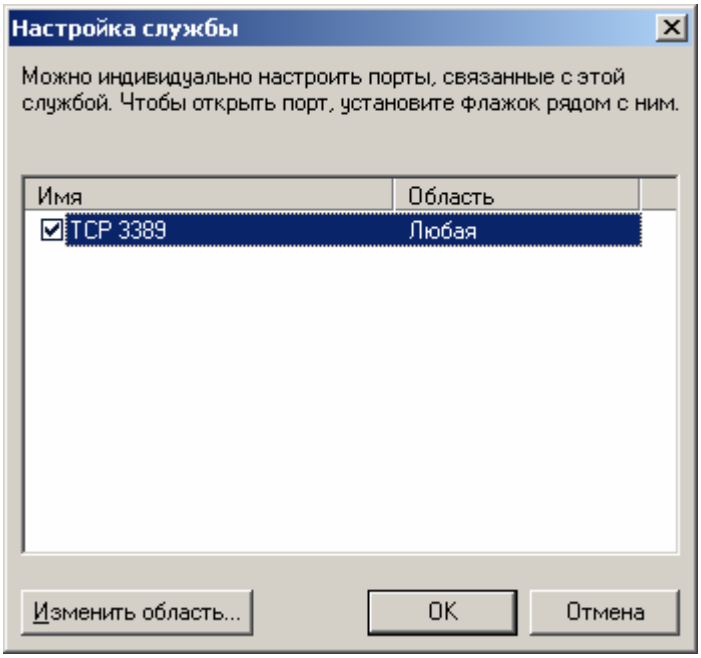


Рис. 6.10. Окно настройки службы «Дистанционное управление рабочим столом»

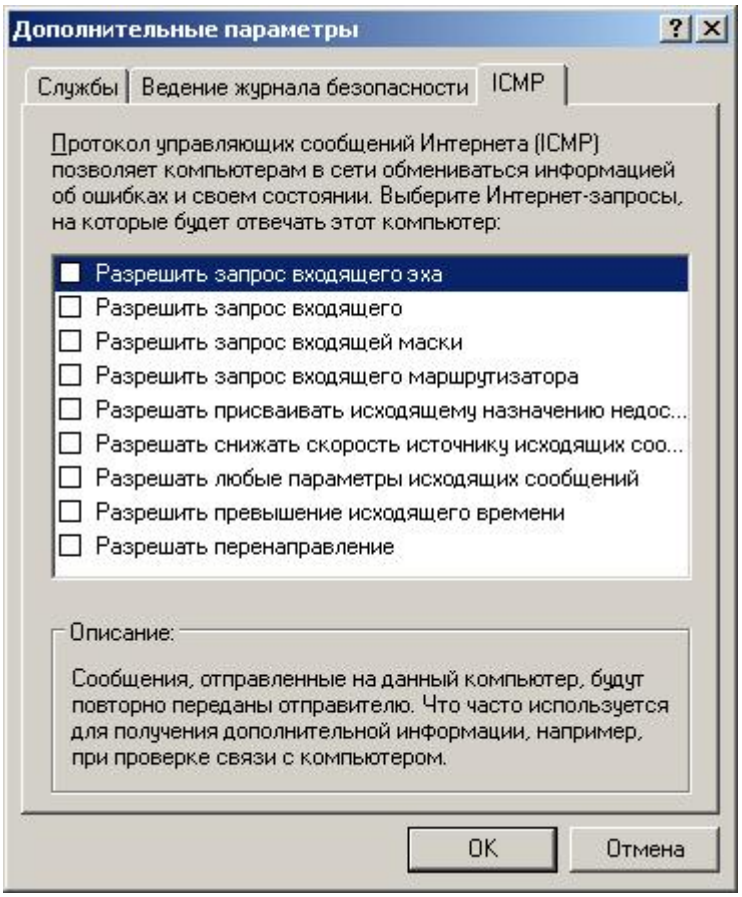


Рис. 6.11. Окно отключения протокола ICMP

6.2.2. Ограничение возможности расширения полномочий при осуществлении локального доступа

Несанкционированное повышение полномочий пользователей до уровня администратора возможно в ОС Windows Server 2003 благодаря существованию уязвимостей в реализации некоторых служб. Основным способом атаки является запуск рядовым пользователем утилиты, использующей ту или иную уязвимость в реализации ОС. Так, известны утилиты PipeUpAdmin, netddemsg, getadmin, позволяющие рядовому пользователю ОС Windows 2000 Server добавить свою учетную запись в состав группы администраторов. В настоящее время в реализации ОС Windows Server 2003 эти уязвимости закрыты и названные утилиты не функционируют. Однако в связи с невозможностью прогнозирования выявления новых уязвимостей в ОС должны быть предприняты меры по защите от воздействия подобных утилит. Заметим, что предполагаемая в схеме защиты конфигурация предусматривает единственно возможный узел — сервер терминального доступа, в котором могут использоваться внешние носители (дисководы, CD-ROM-приводы, USB-порты). Администратор системы, единственный из всех пользователей имеющий право локальной регистрации, перед помещением съемного носителя в накопитель должен убе-

диться в отсутствии на нем опасных программ, позволяющих атаковать систему.

Необходимо использовать меры по организации для пользователей замкнутой программной среды, исключающей запуск любых приложений, кроме приложений, обеспечивающих работу с защищаемыми данными:

- удаление потенциально опасных исполняемых файлов;
- установка NTFS-разрешений, запрещающих пользователям терминального сервера запуск потенциально опасных приложений, необходимых для работы администратора, которые невозможно удалить;
- применение списков программ, доступных для запуска пользователям терминального сервера.

6.3. Настройки сервера MSTS

Настройки сервера MSTS включают в себя настройки политики безопасности ОС Windows Server 2003. Основная задача, решаемая данными настройками, состоит в разрешении возможности доступа к ресурсам терминального сервера только для учетных записей, зарегистрированных в созданной по умолчанию группе «Remote Desktop Users». Кроме того, применяются дополнительные меры по запрету терминального доступа для администратора и пользователей, входящих в состав группы Administrators. Эти меры требуются для защиты учетных записей администраторов от подбора: даже при известном пароле администратора зарегистрироваться с данной учетной записью будет невозможно.

По умолчанию параметр политики безопасности «Allow log on through Terminal Server» (раздел «Назначение прав пользователя» оснастки «Локальная политика безопасности») разрешает терминальный доступ для группы Administrators. Необходимо исключить из перечня учетных записей, имеющих данную привилегию, все группы за исключением «Remote Desktop Users» (рис. 6.12).

Параметр политики безопасности «Deny log on through Terminal Server» (раздел «Назначение прав пользователя» оснастки «Локальная политика безопасности») по умолчанию не установлен. Необходимо установить специальный запрет терминального доступа для группы Administrators, а также для остальных учетных записей, имеющих права администратора, добавив в перечень требуемые учетные записи (рис. 6.13).

Все учетные записи, которые предназначены для терминального доступа, должны быть включены в состав группы «Remote Desktop Users», участие этих учетных записей в составе иных групп должно быть исключено (рис. 6.14).

Для сервера (свойства объекта «Мой компьютер», раздел «Remote») необходимо установить возможность терминального доступа (включить параметр «Enable Remote Desktop on this computer»).

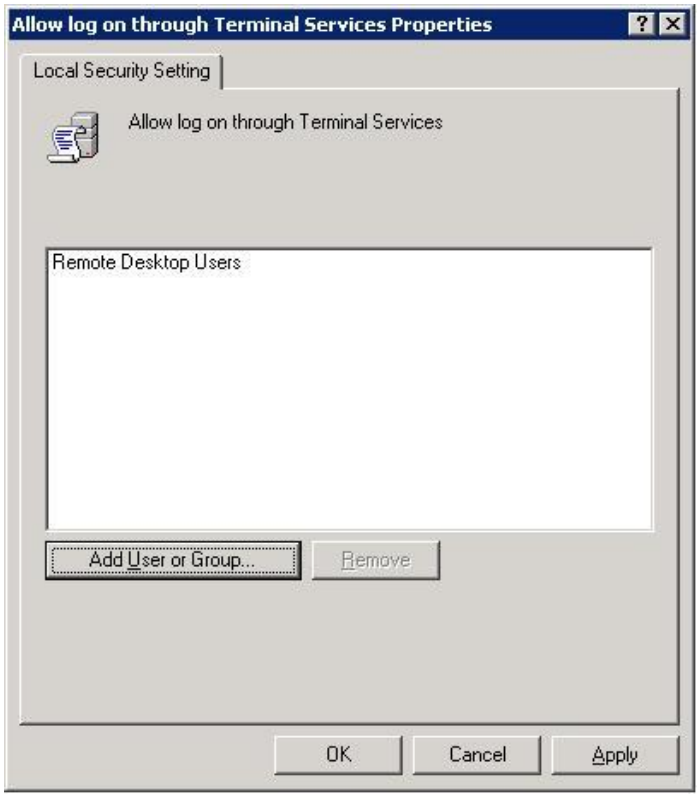


Рис. 6.12. Установка права терминального доступа

Рис. 6.13. Установка специального запрета терминального доступа для учетной записи Администратор

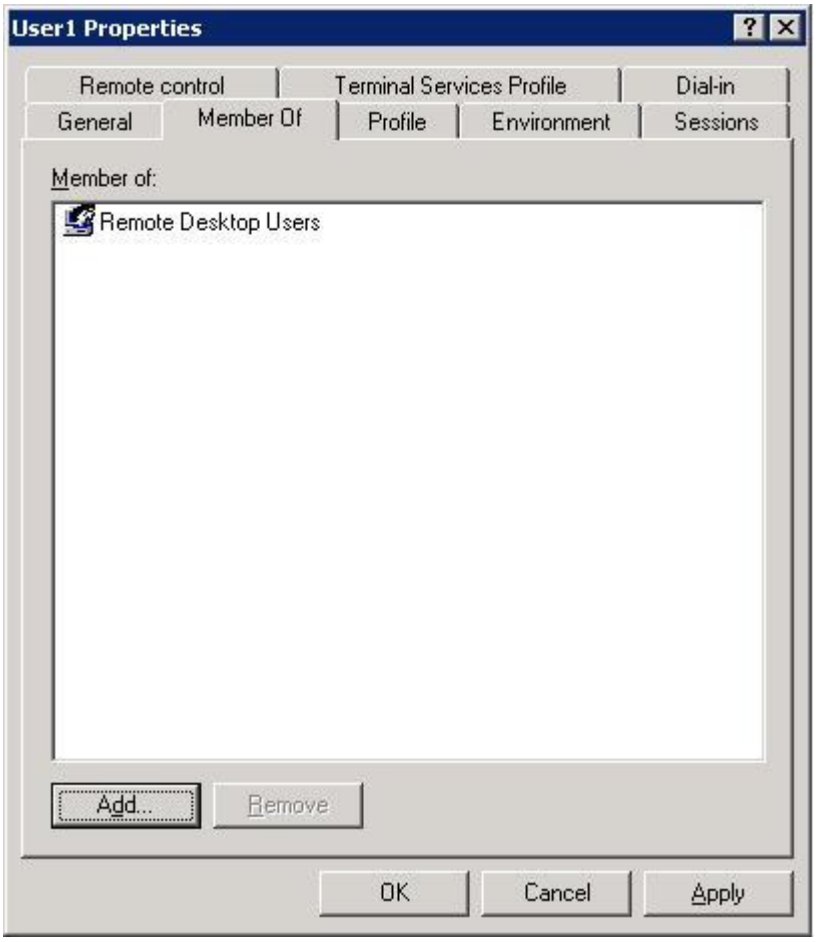


Рис. 6.14. Добавление пользователей терминального доступа в группу «Remote Desktop Users»

6.4. Настройки протокола RDP

Настройка протокола RDP осуществляется в оснастке «Terminal Services Configuration» (рис. 6.15). Основная цель данных настроек состоит в установке требования ввода пароля при регистрации и запрещении использования ресурсов рабочей станции, включая буфер обмена, принтеры и накопители.

Как уже отмечалось выше, терминальный сервер поддерживает шифрование трафика. За настройки шифрования отвечает раздел «General» (рис. 6.16), все настройки в этом разделе можно оставить по умолчанию.

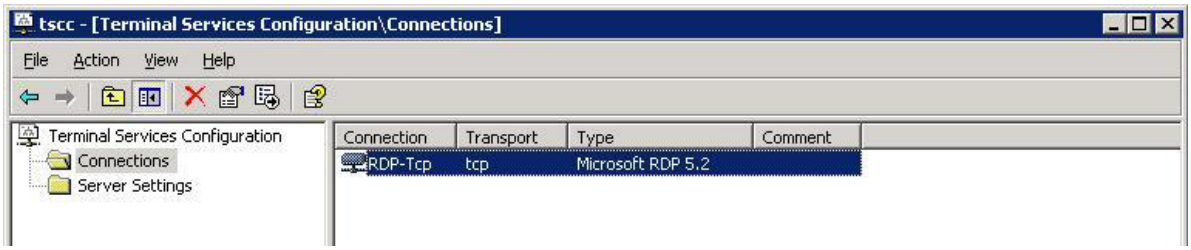


Рис. 6.15. Фрагмент окна настройки протокола RDP

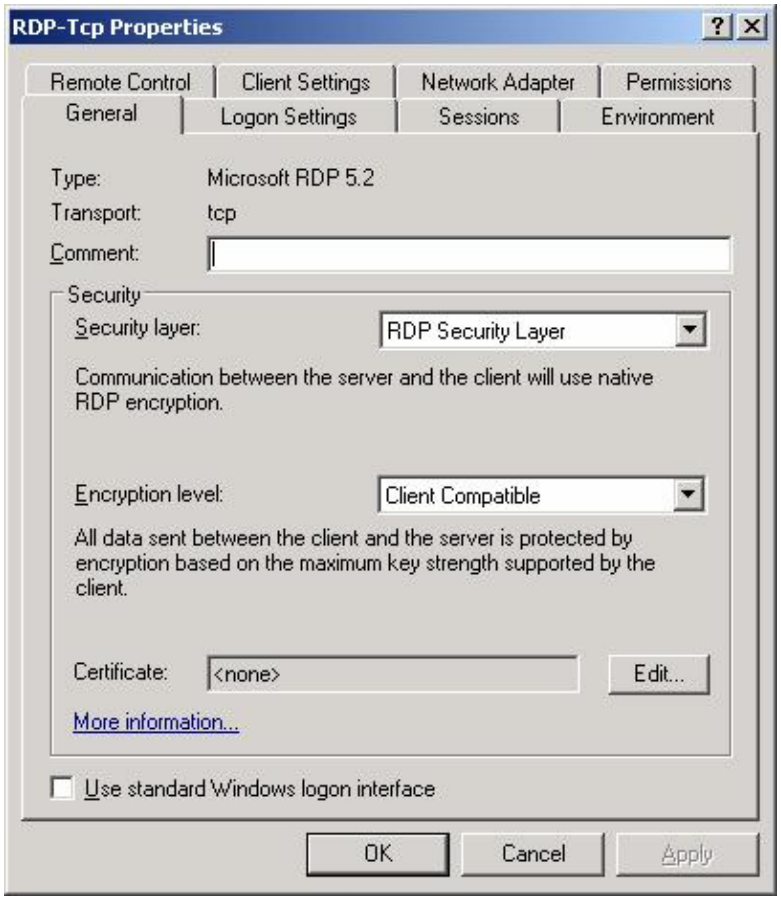


Рис. 6.16. Раздел «General»

В разделе «Logon Settings» необходимо включить требование ввода пароля при регистрации — «Always prompt for password». Исходя из предположения, что все пользователи будут работать со своими учетными записями, включается параметр «Use client-provided logon information».

В связи с тем, что не предполагается ограничений по длительности сеанса терминального доступа, в разделе «Sessions» следует оставить отключенные по умолчанию параметры «Override user settings» (рис. 6.17).

Для каждого пользователя терминального доступа возможна установка собственной программы для автоматического запуска. Это может быть выполнено путем изменения свойств пользователя в оснастке «Пользователи и группы». В тех случаях, когда в свойствах пользователя какая-либо определенная программа не будет назначена, в окне терминала пользователю будет доступен его Рабочий стол (Desktop). В разделе «Environment» рекомендуется установить пункт «Run initial program specified by user profile and Remote Desktop Connections or Terminal Services client».

В связи с тем, что в окне терминального доступа не требуется установка дополнительного контроля действий пользователя, в разделе «Remote Control» рекомендуется установить пункт «Do not allow remote control».

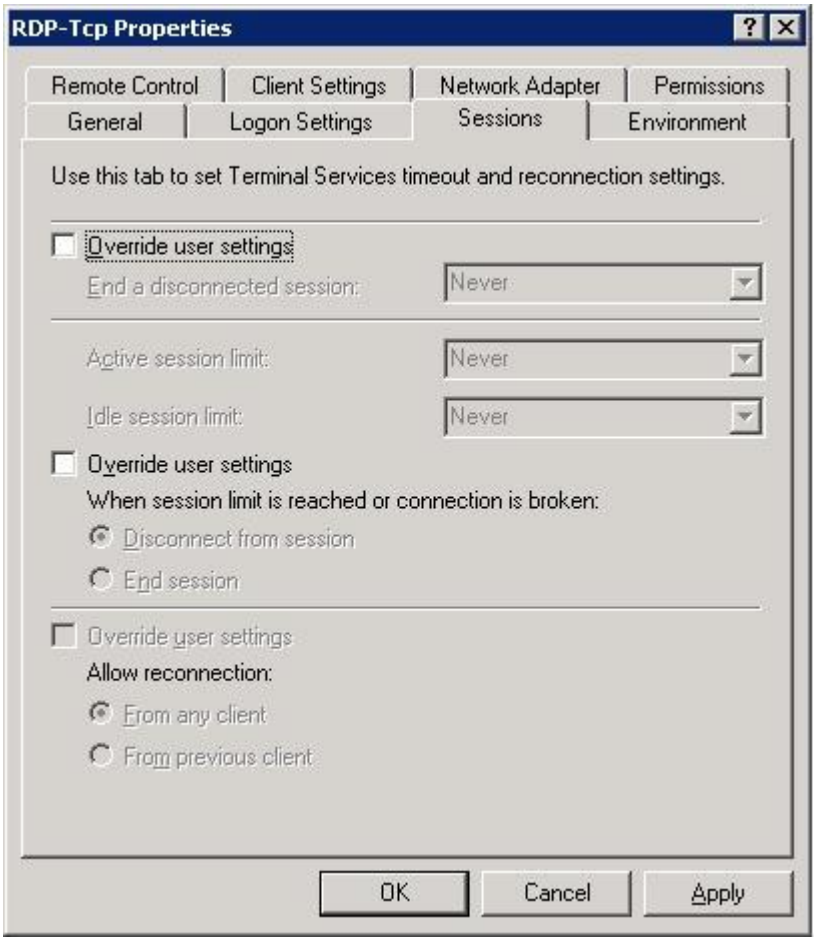


Рис. 6.17. Рекомендуемые установки раздела «Sessions»

В разделе «Client Settings» (рис. 6.18) обязательно должны быть включены запреты использования ресурсов рабочей станции, так как при их отключении будет нарушена вся настраиваемая политика безопасности. Кроме того, при их отключении появляется возможность внедрения на сервер вредоносных программ. Обязательно должны быть установлены запреты на:

- использование совместного буфера обмена (параметр «Clipboard mapping»);
- подключение локальных дисков рабочей станции (параметр «Drive mapping»);
- использование принтеров рабочей станции (параметры «Windows printer mapping» и «LPT port mapping»);
- использование звуковой карты рабочей станции (параметр «Audio mapping») не влияет на политику безопасности, однако может существенно увеличить сетевой трафик и загруженность сервера при прослушивании пользователями звуковых файлов, поэтому в целях повышения производительности сервера терминального доступа данный параметр также рекомендуется отключить.

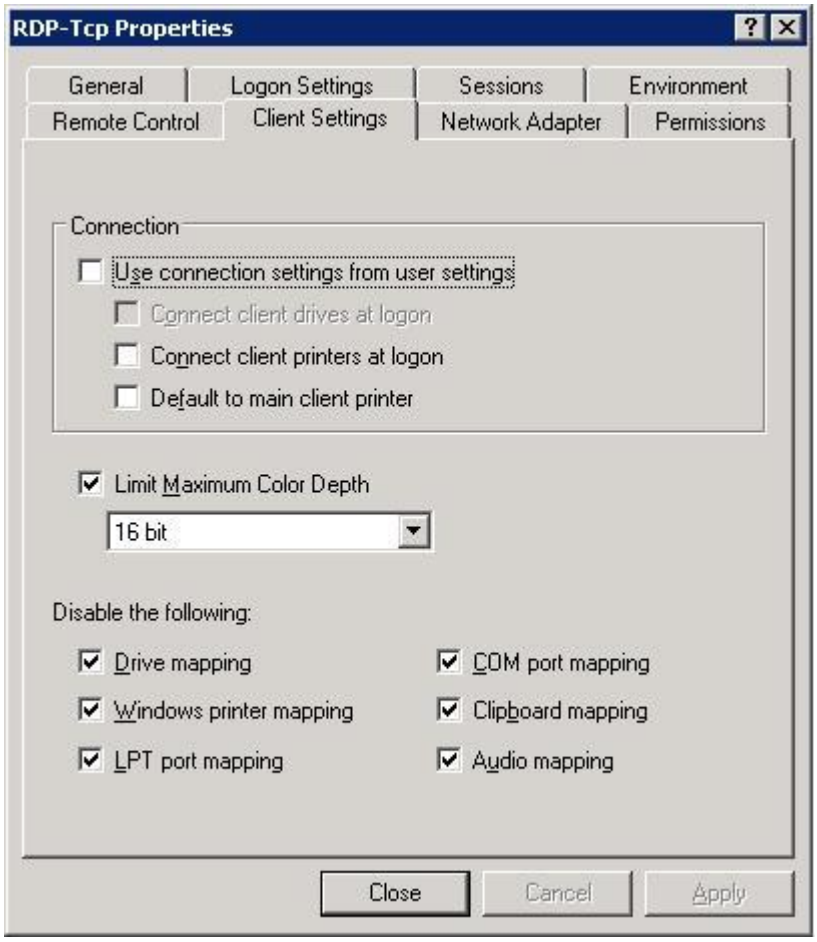


Рис. 6.18. Рекомендуемые установки раздела «Client Settings»

Дополнительно должен быть отключен параметр, позволяющий клиенту терминального доступа самостоятельно настраивать возможность подключения локальных принтеров (должен быть снят параметр «Use connection seings from user settings»).

В качестве дополнительной меры, повышающей производительность, предлагается ограничить параметры видеоизображения (параметр «Limit Maxium Color Depth») глубиной 16 бит.

При количестве пользователей, не превышающем 30, применяется малая часть пропускной способности канала, и с этой точки зрения количество разрешенных подключений не имеет принципиального значения. Однако под каждое соединение система выделяет некоторые ресурсы, кроме того, система для ускорения подключения пользователей к терминальному серверу поддерживает два резервных соединения (т.е. инициализированную среду). Поэтому в целях экономии системных ресурсов желательно установить максимальное количество подключений, равное числу машин, с которых осуществляется работа с терминальным сервером. Это значение устанавливается в разделе «Network Adapter» (рис. 6.19).

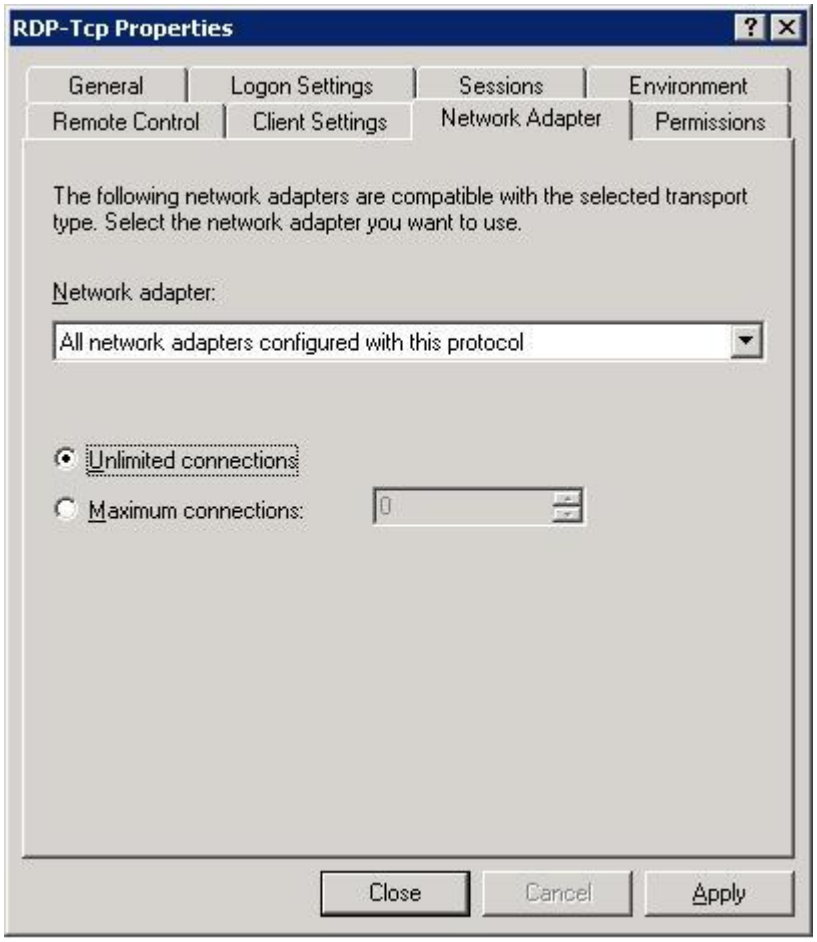


Рис. 6.19. Рекомендуемые установки раздела «Network Adapter»

Установленные по умолчанию разрешения доступа в разделе «Permissions» (рис. 6.20), позволяющие осуществлять доступ к терминальной службе для группы администраторов, необходимо изменить, разрешив доступ лишь для группы «Remote Desktop Users» с правом User access. В списке доступа необходимо установить две записи: для группы «Remote Desktop Users» и для учетной записи «SYSTEM» (рис. 6.21). Для группы «Remote Desktop Users» необходимо установить минимально необходимые права: Query Information, Logon, Connect (рис. 6.22). Для учетной записи «SYSTEM» необходимо установить минимально необходимые права: Query Information, Set Information, Remote Control (рис. 6.23).

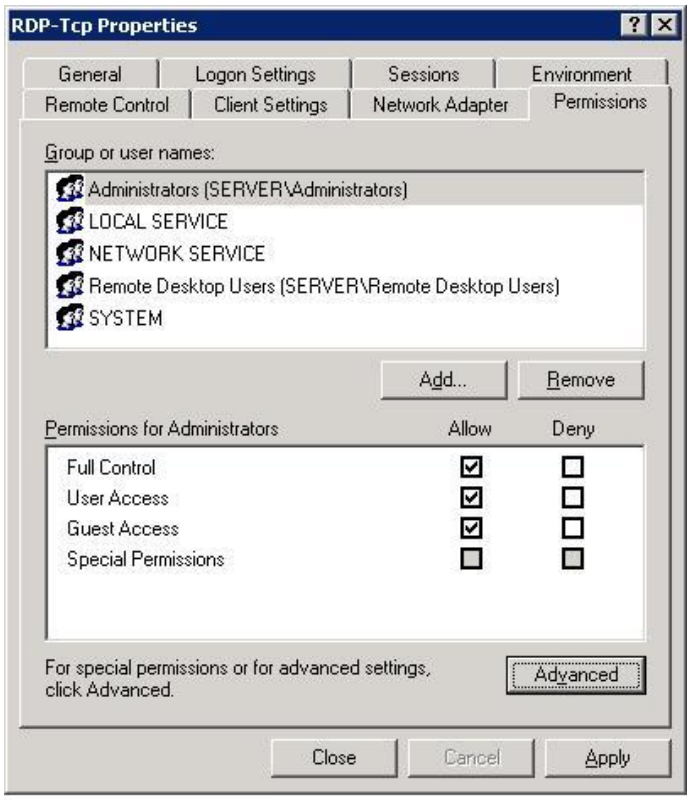


Рис. 6.20. Установки раздела «Permissions» по умолчанию

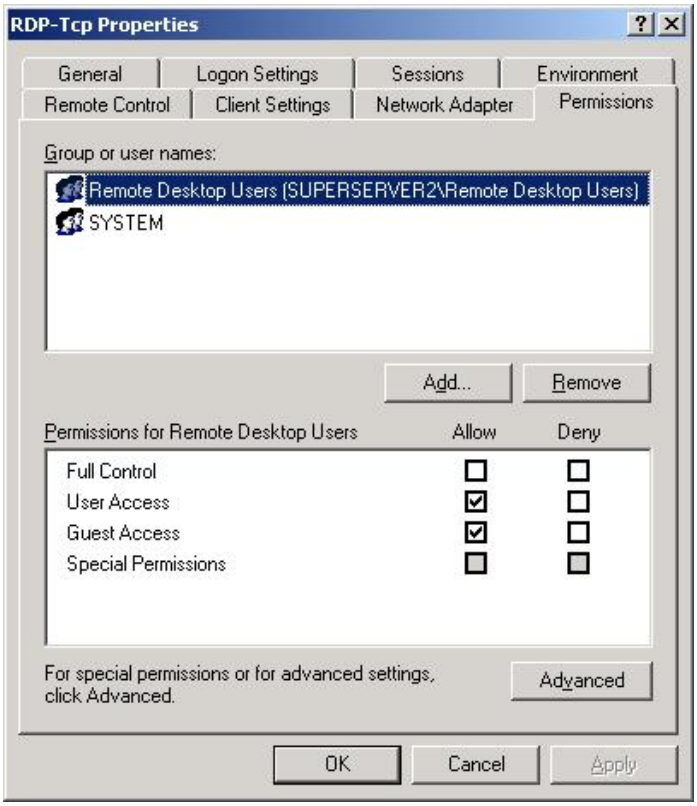


Рис. 6.21. Рекомендуемый список доступа

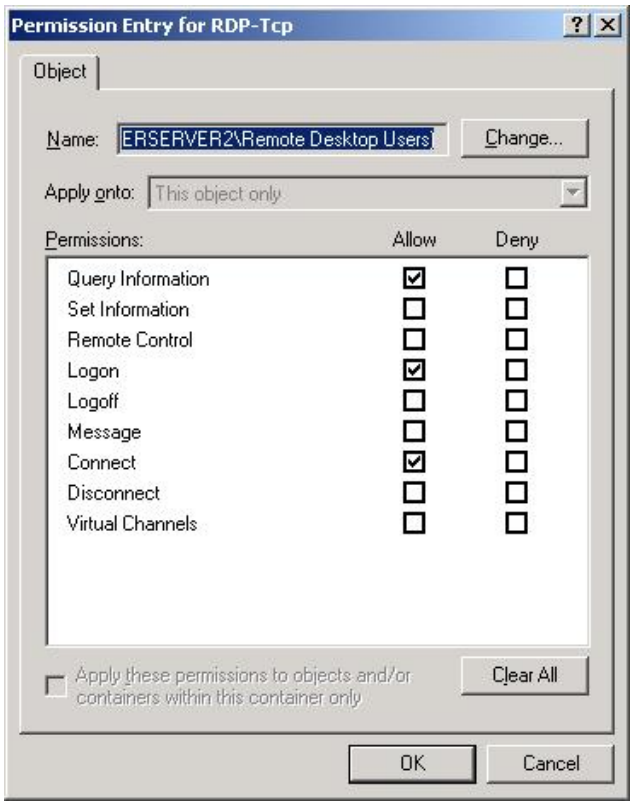


Рис. 6.22. Разрешения доступа группы «Remote Desktop Users»

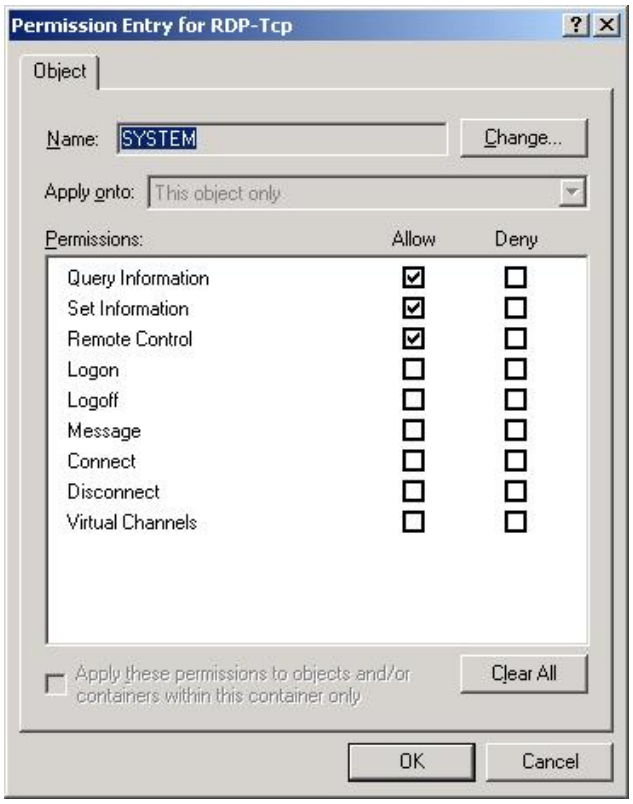


Рис. 6.23. Разрешения доступа учетной записи «SYSTEM»

В разделе «Server Settings» оснастки «Terminal Services Configuration» рекомендуется установить значения параметров, приведенные на рис. 6.24. В частности, рекомендуется ограничить каждого пользователя единственным сеансом работы, установив параметр «Restrict each user to one session». Указанное ограничение не даст возможности подключаться к серверу от имени уже работающего пользователя.

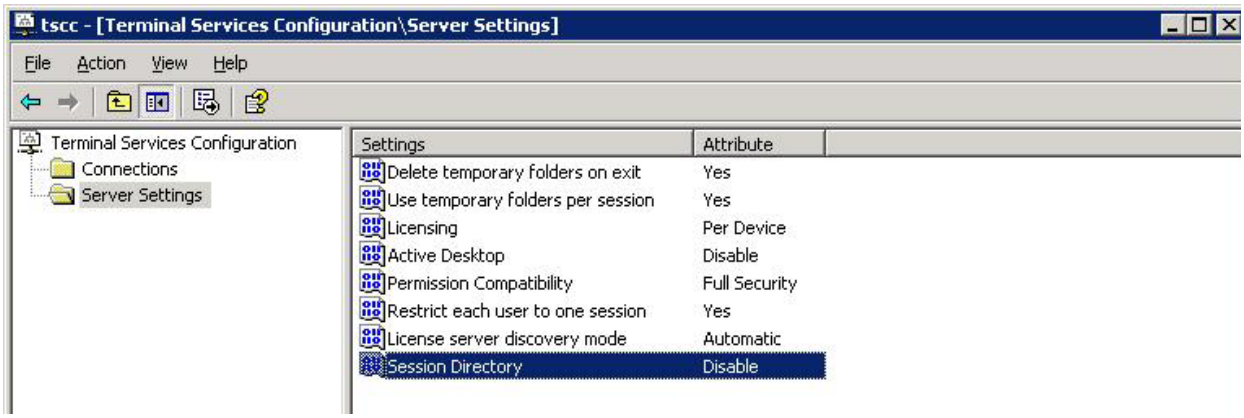


Рис. 6.24. Рекомендуемые установки раздела «Server Settings»

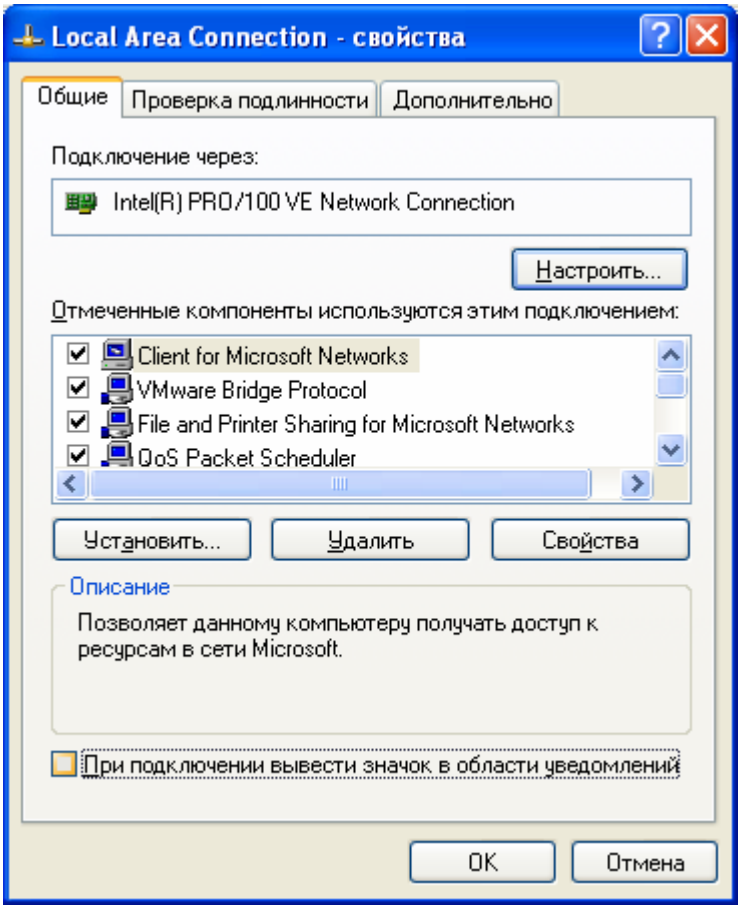


Рис. 6.25. Отключение пиктограммы свойств сетевого подключения

В связи с тем, что изображение обновленного экрана передается сервером MSTS каждый раз после изменений содержимого окон и рабочего стола пользователей, то с рабочего стола и из панели задач рекомендуется удалить все пиктограммы, изображение которых меняется с течением времени. В частности, обязательно необходимо удалить пиктограмму, отображающую состояние сетевого подключения. Данная пиктограмма обновляется при получении/отправке сетевых пакетов. Отправка изменения содержимого экрана, выполняемая MSTS, является причиной обновления данной пиктограммы, что, в свою очередь, приводит к отправке изменения содержимого окна. Таким образом, наличие данной пиктограммы приводит к генерации ненужного сетевого трафика. Для удаления пиктограммы необходимо в свойствах подключения по локальной сети отключить параметр «При подключении вывести значок в области уведомлений» (рис. 6.25).

ВЫПОЛНИТЬ!

1. Настроить виртуальную сеть между основной ОС и виртуальной машиной Windows Server 2003 таким образом, чтобы существовала возможность сетевого доступа к ресурсам Windows Server 2003. Для этого в свойствах сетевого адаптера виртуальной машины установить возможность прямого соединения (Bridged).
2. Добавить в ОС Windows Server 2003 службу MSTS. Для этого установить компоненты Terminal Server и Terminal Server Licensing (***Control Panel ⇒ Add or Remove Programs ⇒ Add/Remove Windows Components***). В процессе установки понадобится дистрибутив ОС Windows Server 2003.
3. Разрешить сетевой доступ к каталогу «C:\Windows\System32\clients\tsclient», содержащему установочный комплект клиента MSTS. В основной ОС установить программу-клиента MSTS из данного каталога.
4. С использованием установленного клиента MSTS выполнить подключение к серверу терминального доступа, указав в параметрах подключения его IP-адрес, имя и пароль учетной записи администратора сервера.
5. Открыть в терминальном окне произвольный текстовый документ, убедиться в возможности копирования его содержимого на диски основной ОС.
6. Разорвать терминальное соединение, выполнив в терминальном окне команду ***Start ⇒ Logoff…***
7. Выполнив команду netstat -aon, получить перечень открытых сервером сетевых портов.
8. Выполнить настройки ОС Windows Server 2003, последовательно отключив сетевые службы, функционирующие по умолчанию (см. п. 6.2.1), за исключением службы Terminal Services.
9. Повторно выполнив команду netstat -aon, проанализировать перечень оставшихся открытых сетевых портов.

10. Активизировать в ОС Windows Server 2003 программу «Брандмауэр Windows» (Windows Firewall). Выполнить настройки, запрещающие использование всех портов за исключением TCP-порта 3389 (см. п. 6.2.1).
11. Из основной ОС с использованием сетевого сканера nmap убедиться в невозможности подключения к сетевым ресурсам ОС Windows Server 2003 за исключением TCP-порта 3389.
12. В ОС Windows Server 2003 создать учетную запись пользователя терминального доступа, включив его в состав группы «Remote Desktop Users» (см. п. 6.3).
13. Выполнить настройки службы MSTS в соответствии с п. 6.3.
14. Выполнить настройки протокола RDP в соответствии с п. 6.4.
15. В основной ОС выполнить попытку подключения к серверу терминального доступа с учетной записью администратора. Убедиться в невозможности подключения.
16. Выполнить подключение к серверу терминального доступа от имени созданной учетной записи. Убедиться в невозможности копирования информации из терминального окна на носители основной ОС.
17. С использованием анализатора сетевого трафика выяснить объем и содержание сетевых пакетов, циркулирующих между клиентом и сервером в режиме терминального доступа.
18. Проанализировать вычислительные ресурсы (объем оперативной памяти, загрузка процессора, загрузка сети) сервера терминального доступа, выделяемые MSTS при подключении одного терминального клиента.

7. СЛУЖБЫ КАТАЛОГОВ

7.1. Общие сведения о службах каталогов

На заре компьютеризации предприятий нагрузка на администраторов компьютерных систем была невелика. Дело было не столько в количественных характеристиках (общее число серверов, сетевого оборудования, рабочих станций и т.п. было значительно ниже в те времена), сколько в качественных — до появления достаточно мощных и дешевых персональных компьютеров работа в системе велась через *терминалы.* С точки зрения администратора это означало, что все управление пользователями сводилось к администрированию одного единственного сервера. Со временем ситуация стала меняться, во-первых, предприятия приобретали все большее количество серверов, во-вторых, вопросам безопасности стало уделяться все больше внимания, что потребовало большего контроля каждого действия пользователя (как следствие, введения строгой аутентификации для каждого значимого для системы действия).

Со временем это привело к тому, что администратор был вынужден создавать учетную запись пользователя на каждом сервере в сети предприятия, а также на каждой рабочей станции, которой имеет право пользоваться сотрудник. Пользователь, в свою очередь, должен был постоянно предоставлять аутентифицирующую информацию (каждому сервису корпоративной сети).

Решением этой проблемы стало создание так называемых *служб каталогов* — систем централизованного хранения информации о пользователях. Международная организация по стандартизации (ISO) предложила стандарт X.500, который описывал функционирование такой системы. Однако протокол взаимодействия, описанный в рамках стандарта, оказался слишком перегруженным для сетей TCP/IP. По этой причине какое-то время службы каталога создавались производителями с использованием различных протоколов взаимодействия (NDS, NIS, NT4 domain). Такое разнообразие реализаций привело к тому, что независимые разработчики сетевых сервисов либо совсем не обеспечивали совместимости со службами каталогов, либо обеспечивали совместимость с какой-то конкретной реализацией. Как следствие, каждый производитель службы каталога должен был обеспечить клиентов и базовым набором служб (например, собственным файл-сервером).

Ситуация изменилась только тогда, когда Интернет-сообщество опубликовало «облегченный» вариант стандарта X.500 — протокол LDAP (Lightweight Directory Access Protocol[1], RFC 4510). Протокол обеспечивал простой доступ к каталогу в рамках TCP-соединения, касался также вопросов аутентификации и собственно структуры каталога. Позже стандарт претерпел

[1] Название так и переводится: облегченный протокол доступа к каталогу.

некоторое количество редакций и в настоящее время поддерживается практически любым сетевым приложением и реализован в любой службе каталога.

Развитие протокола продолжается и сегодня. Уже используется версия 3 данного протокола, а также опубликован целый ряд расширений (например, поддержка language tags, позволяющая хранить имя пользователя, записанное на нескольких языках, и отправлять клиенту имя на его родном языке).

Многие современные сетевые приложения также обладают встроенной поддержкой LDAP (почтовые клиенты способны производить поиск адреса по имени пользователя, web-серверы и СУБД могут извлекать список пользователей из каталога LDAP, распределенные приложения могут хранить свои настройки в каталоге LDAP и т. п.)

После того как все данные о пользователях, группах, в которые они входят, и их правах доступа переместились в централизованное хранилище, разработчики стали задумываться и о решении другой проблемы современных компьютерных систем — о постоянной необходимости пользователей в аутентификации. Производители стали выпускать системы с концепцией единого входа в сеть (так называемые системы SSO — single sign on), т. е. пользователь при однократном вводе пароля получал доступ ко всем ресурсам сети. На практике это работало только с сервисами самого производителя, поскольку слишком различались протоколы сетевой аутентификации у различных приложений. Те способы решения проблемы, которые пытались предложить конкретные производители, не были универсальны. Решения от Microsoft позволяли хранить пароль пользователя не в виде хэша, а в восстанавливаемом виде (т. е. практически в открытом), Novell позволял хранить пароль различными способами (зашифрованный хэшем пароля секретный ключ для сервисов Novell, NTLM-хэш для доступа к сервисам Microsoft). Такие решения позволяли пользователю не вводить свой пароль лишний раз, но не были ни безопасными, ни универсальными.

Ситуация вновь изменилась благодаря открытым стандартам. С небольшим промежутком времени появились описания методов, позволяющих разделить сам сетевой сервис и процесс аутентификации пользователя, что позволяло использовать любой сетевой сервис с нужным методом аутентификации. В настоящее время три этих метода распространены достаточно широко — это SASL (Simple Authentication and Security Layer, RFC 4422), GSSAPI (Generic Security Service Application Program Interface, RFC 2743) и SPNEGO (Simple and Protected GSSAPI Negotiation, RFC 2478). Все перечисленные методы реализуют примерно одну и ту же идею — существует четкий интерфейс взаимодействия сетевого сервиса с библиотекой, которая, как правило, является расширяемой при помощи модулей (опять же со строго зафиксированным интерфейсом). При этом в стандарт закладывается механизм согласования используемого модуля для аутентификации клиента и сервера.

Подобные механизмы часто встречались в рамках одного протокола (например, согласование диалекта SMB или метода аутентификации NFS),

однако они были уникальными для каждого протокола, что усложняло возможность использования одних и тех же методов аутентификации для разных сетевых сервисов.

Удобство заключается также в том, что универсальные методы аутентификации можно вкладывать друг в друга. Иными словами, если сервис был написан с поддержкой механизма GSSAPI, а у производителя службы каталога есть только модули SASL, реализующие необходимые методы аутентификации, то любое заинтересованное лицо может создать GSSAPI-модуль, который реализует аутентификацию по механизму SASL (рис. 7.1).

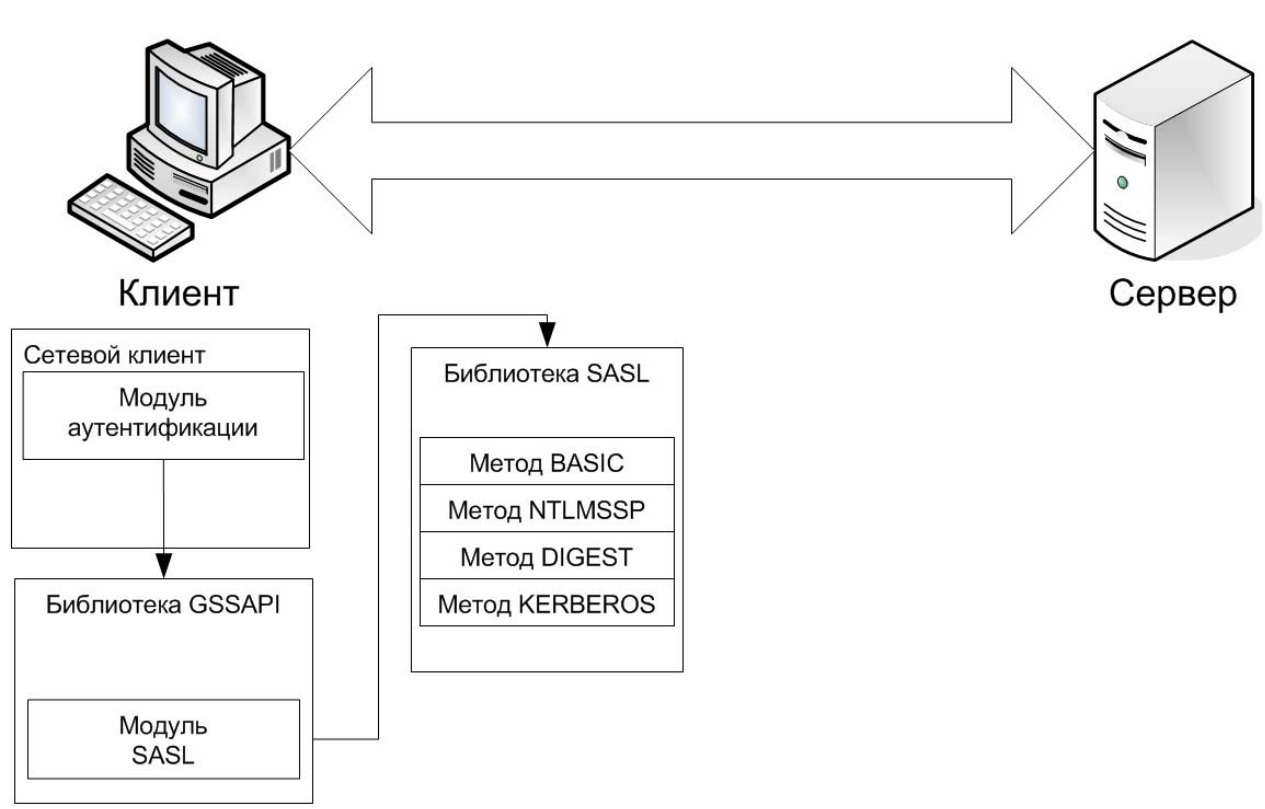


Рис. 7.1. Пример вложенности механизмов сетевой аутентификации

Однако использование универсальных механизмов лишь частично решает проблему единой аутентификации при входе в сеть. Необходим также некоторый безопасный протокол, который аутентифицирует клиента перед сервером (сервисом), с учетом того, что сам сервер ничего о пользователе не знает и должен использовать в качестве посредника сервер службы каталогов.

В качестве универсального протокола аутентификации можно использовать инфраструктуру открытых ключей (например, на базе сертификатов X.509) или протокол Kerberos (см. ниже).

В данной главе рассмотрена реализация службы каталога корпорации Microsoft — Active Directory (AD) на примере интеграции ее в среду сторонних сервисов. А именно, решение задачи аутентификации пользователей на web-сервере Apache при помощи протокола Kerberos, а также настройка меха-

низмов аутентификации и авторизации на рабочей станции под управлением ОС Linux с целью получения авторизующей информации из Active Directory.

Служба Active Directory состоит из целого набора сервисов, связанных между собой. Основой Active Directory является система разрешения имен, при помощи которой рабочие станции способны прозрачно обнаруживать серверы домена, например LDAP-серверы. В существующей реализации сервиса каталога в качестве службы разрешения имен выбрана система DNS (в отличие от предыдущих версий, которые использовали систему имен NetBIOS). Для хранения каталога LDAP, а также для обеспечения аутентификации по протоколу Kerberos в сети Microsoft выделяются специальные сервера, которые называются *контроллерами домена*. В целом контроллер домена — это выделенный сервер, предназначенный для обеспечения сервисов LDAP и KDC (см. ниже)

В качестве первого упражнения предлагается установить службу Active Directory на ОС Windows Server 2003. В качестве промежуточного шага требуется установить и настроить DNS-сервер.

ВЫПОЛНИТЬ!

1. Запустить виртуальную машину Windows Server 2003 (далее DC).
2. Установка и настройка DNS-сервера. Этот шаг состоит из двух основных действий: настройка клиента DNS и установка и настройка сервера DNS.
   a. Настроить имя компьютера (после внесения изменений потребуется перезагрузка): name: DC, DNS suffix: example.com.
   b. Установить DNS-сервер (Add/Remove programs ⇒ Windows components ⇒ Network services ⇒ DNS).
   c. Запустить оснастку администрирования DNS (Start ⇒ Administrative Tools ⇒ DNS).
   d. Создать зону прямого просмотра. Выбрать из контекстного меню объекта «Forward lookup zones» пункт «New zone». Выбрать тип зоны «Primary», имя зоны «example.com», разрешить динамические обновления для зоны. Эта зона будет хранить информацию о создаваемом нами домене Active Directory. Имя домена Active Directory совпадает с именем DNS домена, в котором и будет храниться информация о серверах и рабочих станциях Active Directory.
   e. Создать зону обратного просмотра. Выбрать из контекстного меню объекта «Reverse lookup zones» пункт «New zone». Выбрать тип зоны «Primary», network id 192.168.0.
   f. Создать запись для рабочей станции ws-linux (это пригодится позже, когда будет производиться настройка соответствующей виртуальной машины). Выбрать в списке зон прямого просмотра зону «example.com», выбрать из контекстного меню зоны пункт «New host (A)», указать имя хоста «ws-linux», IP-адрес 192.168.0.2, выбрать опцию «Create associated pointer (PTR) record».

3. Повышение роли сервера до контроллера домена. Данная процедура практически полностью автоматизирована, необходимо лишь указать некоторую вспомогательную информацию (например, о роли и месте нашего нового контроллера домена в существующей инфраструктуре). Здесь мы не будем подробно рассматривать сложные схемы построения доменов AD.
   a. Запустить утилиту dcpromo.
   b. Выбрать пункт «New domain controller in a new domain».
   c. Выбрать пункт «New domain in a new forest».
   d. Указать имя домена «example.com».
   e. NetBIOS-имя и расположение служебных файлов оставить по умолчанию.
   f. Убедиться, что тест динамических обновлений на DNS-сервере прошел успешно.
   g. Выбрать режим совместимости только с ОС Windows 2000 и 2003.
   h. Указать пароль режима восстановления «P@ssw0rd».
   i. Дождаться окончания работы мастера и перезагрузиться.
4. Установка дополнительных инструментов администрирования.
   a. Установить пакет «adminpak.msi», расположенный в каталоге «WINDOWS\System32». Данный пакет содержится во всех серверных версиях ОС Windows и содержит набор оснасток MMC для администрирования различных сервисов ОС Windows.
   b. Установить пакет «suptools.msi». Пакет Windows Support Tools поставляется вместе с дистрибутивом операционной системы (обычно располагается в каталоге SUPPORT установочного диска).
   c. Установить пакет «rktools.exe».

7.2. Структура каталога LDAP

Для дальнейшей работы с каталогом LDAP необходимо более подробно ознакомиться с его основными понятиями и компонентами, а также с набором утилит, которые позволяют производить манипуляции с ним.

7.2.1. Схема LDAP

Каталог LDAP имеет древовидную структуру. Узлы дерева называются *объектами*. Объекты делятся на *листья* и *контейнеры*. Каждый объект содержит набор *свойств* (различную информацию об объекте, например его имя, дата создания и т. п.). Список свойств каждого объекта зависит от *класса* этого объекта. По аналогии с объектно-ориентированным программированием классы выстраиваются в иерархическое отношение — предок-потомок. Класс-потомок содержит свойства своего предка, при этом поддерживается множественное наследование. Описание всех классов и их свойств хранится в самом каталоге LDAP и называется *схемой* каталога.

Схема содержит описания двух типов — описание атрибутов и описание классов. *Атрибут* — описание некоторого свойства, включающее в себя *синтаксис* (аналог типа данных, но кроме информации о типе он содержит также функцию сравнения, например есть синтаксис для строки чувствительной и нечувствительной к регистру). Кроме синтаксиса атрибут содержит информацию о своем имени, а также некоторые дополнительные данные (например, ограничения на синтаксис или указание, что атрибут хранит несколько значений). *Классы* бывают *абстрактными, структурными* и *вспомогательными*. Для каждого класса задается набор обязательных (значение этих атрибутов не может быть пустым) и необязательных (значение может отсутствовать) атрибутов. Кроме этого у класса указывается список возможных контейнеров для него, таким образом, объект является контейнером, если хотя бы один класс указывает его в качестве своего возможного контейнера.

Схема LDAP-каталога хранится в самом каталоге. Для описания схемы Active Directory существует два объекта: attributeSchema для атрибутов и classSchema для классов (оба эти класса, как и все их атрибуты также описаны внутри схемы).

7.2.2. Система имен LDAP

Для того чтобы LDAP-клиент имел возможность получать данные от LDAP-сервера, необходимо указать конкретный объект в каталоге. Для этого служат специальные атрибуты, для которых схемой задается требование уникальности в пределах одного контейнера. Такие атрибуты называются *относительным различимым именем* (relative distinguished name, rdn). Теперь для того, чтобы идентифицировать объект в пределах дерева, достаточно указать относительные различимые имена всех объектов в цепочке, начиная от корня дерева. Полученная последовательность различимых имен называется *полным различимым именем* (fully distinguished name, fdn). Для обозначения полного различимого имени принята запись

```
<имя атр.1>=<знач. атр.1>,
<имя атр.2>=<знач. атр.2>,..,
<имя атр.N>=<знач. атр.N>
```

Например, контейнер, хранящий схему каталога Active Directory (в домене example.com), имеет следующее полное различимое имя: CN=Schema, CN=Configuration, DC=example, DC=com. Как видно из примера, атрибут, являющийся относительным различимым именем у каждого объекта может быть свой (здесь CN — у объектов-контейнеров Schema и Configuration, DC — у объектов example и com). Указание того, какой атрибут может выступать в роли различимого имени, задается в схеме для каждого класса.

7.2.3. Инструментарий для работы с LDAP-каталогом

Для работы с произвольными LDAP-каталогами существует огромное количество программ как бесплатных, так и коммерческих. В данном разделе будут рассмотрены два основных инструмента: оснастка MMC ADSI Edit (которая обладает большим функционалом, если речь идет о LDAP-каталоге Active Directory), а также утилиты пакета ldap-utils (преимущество этих утилит в том, что они реализованы практически для каждой ОС).

Оснастка ADSI Edit входит в пакет Support Tools ОС Windows Server 2003. Для вызова этой утилиты необходимо в окне Start ⇒ Run набрать «adsiedit.msc». Загрузится консоль управления Microsoft с оснасткой ADSI Edit, подключенной к трем разделам каталога (рис. 7.2): Domain (собственно данные каталога Active Directory), Configuration (служебная информация) и Schema (схема каталога). При необходимости можно подключиться и к другим разделам каталога, выбрав соответствующий пункт контекстного меню корневого узла оснастки.

Для того чтобы просмотреть содержимое раздела, достаточно раскрывать соответствующие узлы (в роли узлов будут выступать объекты-контейнеры, обычные объекты будут отображаться в правом окне).

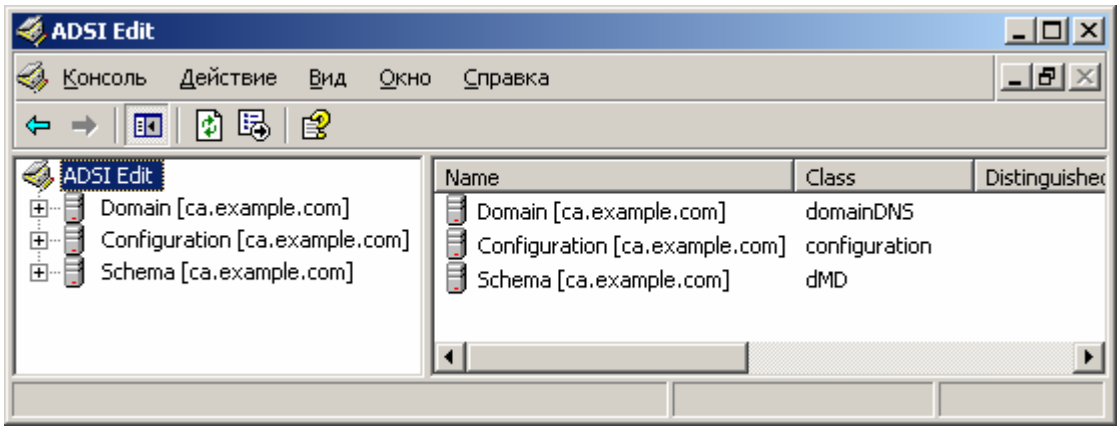


Рис. 7.2. Оснастка ADSI Edit

Для каждого объекта-контейнера из контекстного меню доступен следующий набор действий: переместить объект, переименовать объект, удалить объект, вызвать свойства объекта, а также создать внутри новый объект, допустимый по схеме. У обычных объектов доступно только перемещение, изменение имени, удаление и вызов свойств.

Пункт меню «Свойства» (Properties) позволяет редактировать свойства объекта при помощи редактора атрибутов (рис. 7.3), а также настраивать разрешения на практически любое действие над атрибутами, объектом и его содержимым (в случае объекта-контейнера).

Внешний вид редактора прав доступа представлен на рис. 7.4.

Основное применение оснастки ADSI Edit — детальное управление правами пользователей на объекты и атрибуты каталога, также ADSI Edit приме-

няется для редактирования тех атрибутов объектов, которые не редактируются стандартными средствами администрирования Active Directory (в данном пособии это будут атрибуты Unix пользователя).

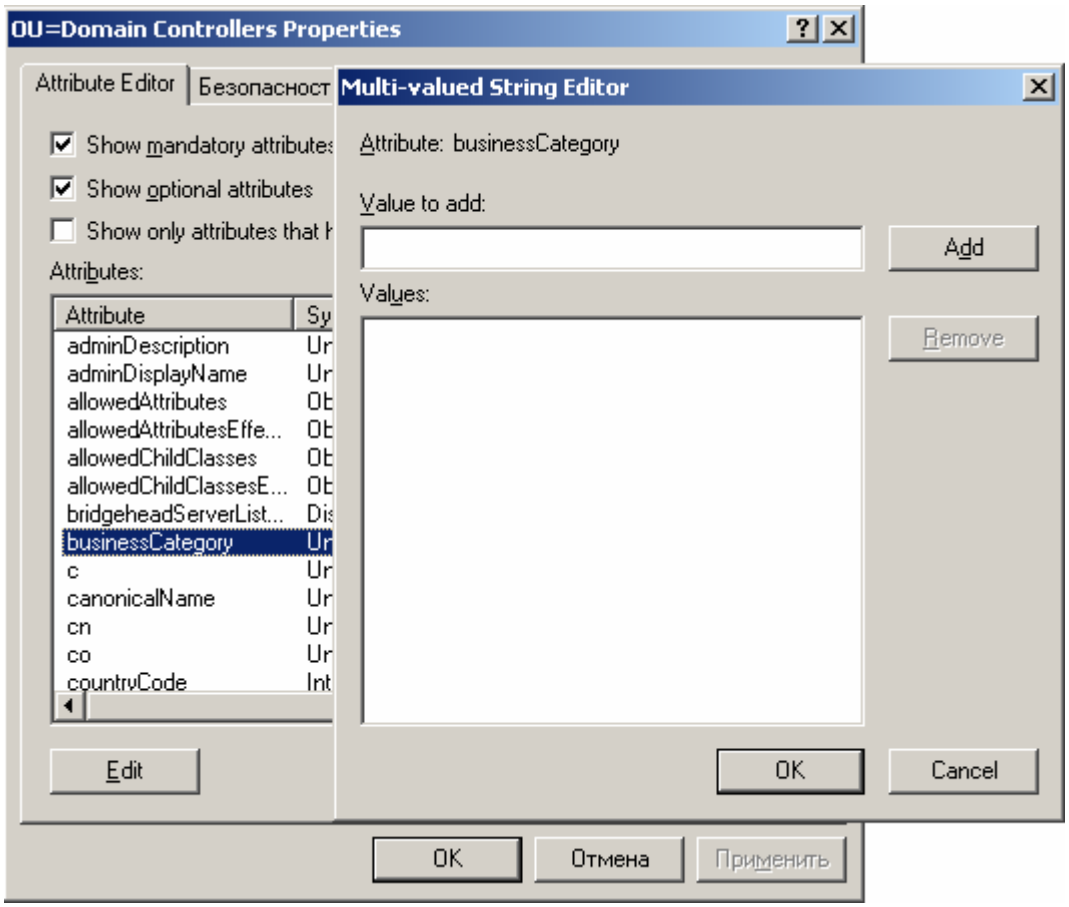


Рис. 7.3. Редактор атрибутов

Пакет ldap-utils включает в себя следующие основные утилиты:

- ldapsearch — служит для поиска объектов в каталоге от указанного узла и на указанную глубину,
- ldapmodify — служит для модификации объектов LDAP,
- ldapadd — то же, что и ldapmdify с опцией -a (добавление),
- ldapdelete — утилита для удаления объектов из каталога,
- ldapmodrdn — утилита для смены имени объекта.

При запуске каждой утилиты пакета необходимая информация берется из трех источников (каждый последующий перекрывает предыдущий): файла «/etc/ldap/ldap.conf», файла «~/.ldaprc» и опций командной строки. Основными параметрами являются:

- пользователь LDAP, от имени которого будет происходить подключение (вообще говоря, это может быть любой объект каталога),
- LDAP-сервер, к которому будет осуществляться подключение,
- метод проверки подлинности,

- количество выводимой отладочной информации,
- учетные данные.

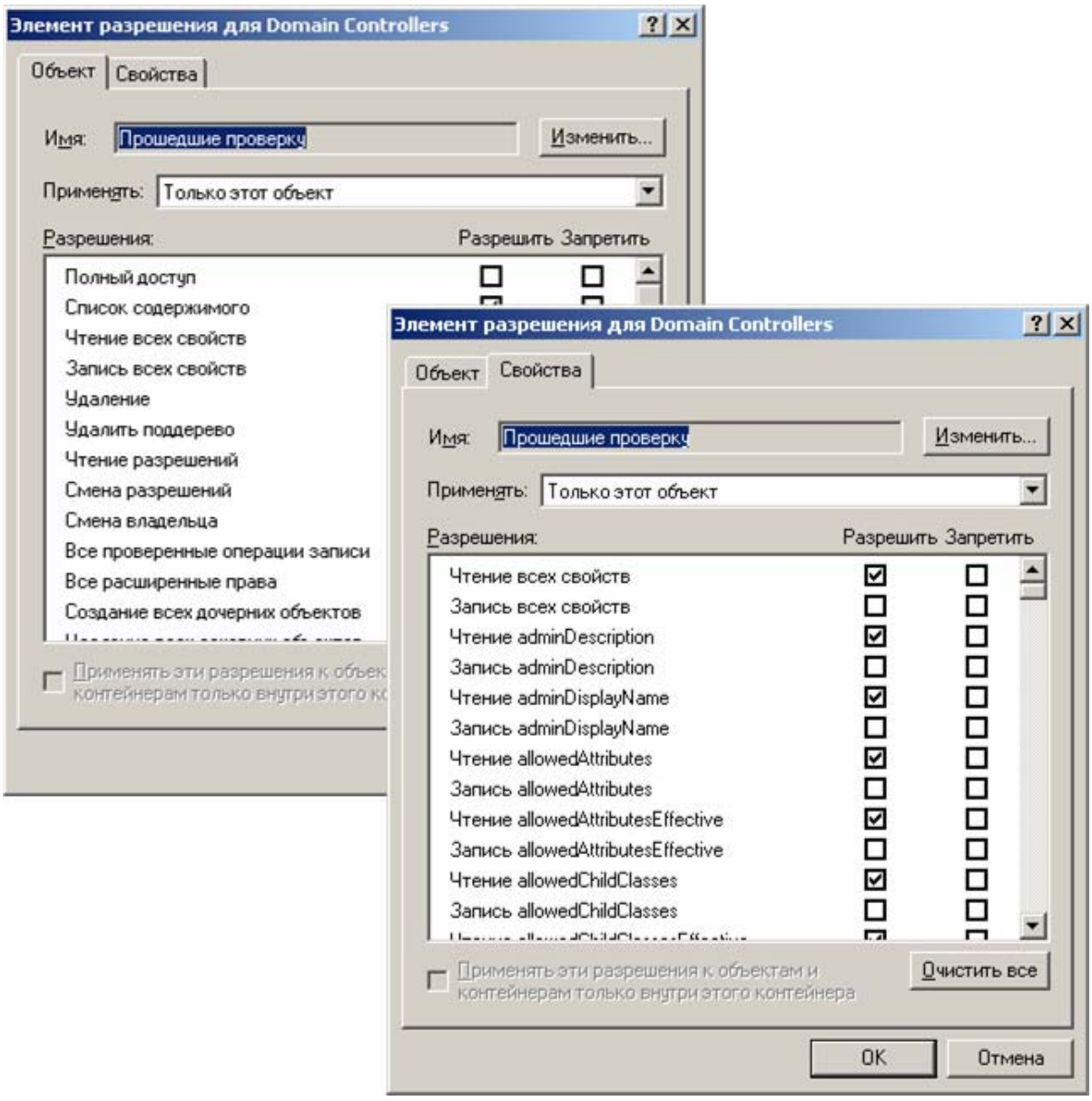


Рис. 7.4. Редактор прав доступа

Каждая из утилит имеет также собственные параметры. В случае с утилитами ldapmodify и ldapadd – это LDIF-файл, который описывает необходимые изменения. Информация о формате LDIF содержится в RFC 2849. Утилита ldapsearch ожидает параметры фильтра для запросов, а также список атрибутов, которые возвратит сервер для каждого объекта, удовлетворяющего фильтру (по умолчанию возвращаются все атрибуты).

Фильтры LDAP определены в самом протоколе и описаны в RFC 4515. Например, пусть необходимо найти в LDAP-каталоге объект-компьютер, у ко-

торого имя начинается на win-. Тогда соответствующий фильтр будет иметь вид: (&(objectClass=computer)(cn=win-*))[1].

Полностью запрос приведен на рис. 7.5. В данном примере утилита ldapsearch вызвана с наиболее часто употребляемыми опциями: -h <server> — задает LDAP-сервер (по имени или IP-адресу), -b <dn> — задает различимое имя объекта, с которого начнется поиск, -D <dn> — задает различимое имя объекта LDAP, от имени которого производятся запросы к серверу (с точки зрения стандарта LDAP это может быть абсолютно любой объект, в Active Directory только пользователь или компьютер), -x — задает режим аутентификации simple bind (пароль передается открытым текстом), -w <password> — задает пароль.

Также при вызове ldapsearch указано, что сервер должен возвратить только атрибуты cn у найденных объектов.

```
root@ws-linux:~# ldapsearch -h dc -b dc=example,dc=com -D cn=Administrator,cn=us
ers,dc=example,dc=com -x -w P@sswOrd "(&(ObjectClass=user)(cn=win-*))" cn
# requesting: cn
#

# win-test, Computers, example.com
dn: CN=win-test,CN=Computers,DC=example,DC=com
cn: win-test

# search reference
ref: ldap://ForestDnsZones.example.com/DC=ForestDnsZones,DC=example,DC=com

# search reference
ref: ldap://DomainDnsZones.example.com/DC=DomainDnsZones,DC=example,DC=com

# search reference
ref: ldap://example.com/CN=Configuration,DC=example,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
root@ws-linux:~#
```

Рис. 7.5. Использование утилиты ldapsearch

Рассмотрим ответ LDAP-сервера (рис. 7.5). Для удобства чтения, а также дальнейшего использования ldapsearch отображает информацию, полученную от сервера, в виде LDIF-файла (строки, начинающиеся с «#», являются комментариями). Сначала перечисляются объекты и их атрибуты, которые вернул сервер на посланный запрос. В нашем примере это компьютеры win-

[1] Важно отметить, что использование метасимволов в фильтре возможно не для каждого атрибута (в Active Directory метасимволы в запросах доступны для атрибутов с флагом indexed, который задается в схеме).

ws1 и win-ws2, затем идет несколько особых ответов, которые называются referral или external reference (внешние ссылки). Эти ссылки содержат указатели на другие *разделы* каталога (здесь раздел — это некоторая часть каталога, которая может храниться на другом сервере). Внешняя ссылка приведена в виде URL, имеющего формат ldap[s]://<сервер>/<dn>, где ldap или ldaps обычный или SSL-вариант протокола соответственно.

Конечной целью данной главы является интеграция рабочей станции под управлением ОС Linux в среду Active Directory, данная операция требует целого ряда настроек, в частности расширения схемы каталога для хранения Unix-атрибутов пользователя. Существует множество готовых расширений для схемы AD, поставляющихся с различными продуктами интеграции Unix систем в среду AD.

Кроме того, для дальнейшей работы создадим специального вспомогательного пользователя и группу для рабочих станций под управлением ОС Linux. Дело в том, что при авторизации рабочая станция должна иметь возможность определить следующие параметры пользователя: его идентификатор (uid), домашний каталог, командный интерпретатор, а также членство в группах. Эта информация по умолчанию недоступна для анонимного пользователя, поэтому необходимо создать специального непривилегированного пользователя, от имени которого рабочие станции будут осуществлять доступ.

ВЫПОЛНИТЬ!

5. Запустить виртуальную машину DC.
6. Добавить в схему каталога атрибуты для Unix-пользователей. Расширение схемы взято из пакета MS services for Unix:
    a. Запустить оболочку cmd.exe
    b. При помощи команды cd перейти в каталог «\schema»
    c. Схема сохранена в виде ldif-файла, чтобы ее добавить в каталог, необходимо воспользоваться стандартной утилитой для манипуляции с ldif файлами — ldifde. Для этого необходимо выполнить команду (здесь опция -i обозначает, что будет происходить импорт данных, -k позволяет выполнять импорт даже после появления некоторых некритических ошибок, опция -f указывает, что данные необходимо брать из файла):

        ```
        ldifde –i –k –f schema.ldf
        ```

7. Создать специального пользователя для рабочих станций под управлением ОС Linux:
    a. Запустить оснастку управления пользователями в Active Directory (Start ⇒ Programs ⇒ Administrative tools ⇒ Active Directory Users and Computers).
    b. В контекстном меню контейнера «Users» выбрать пункт «New», объект «User».

c. В качестве «First name» и «Users logon name» указать «proxyuser», задать пароль «P@ssw0rd», снять опцию «User must change password at next logon» и выставить опции «User cannot change password» и «Password never expires».
d. В контекстном меню контейнера «Users» выбрать пункт «New», объект «Group».
e. Указать имя группы «proxygroup», остальные опции оставить по умолчанию.
f. В окне свойств пользователя открыть закладку «Member Of», добавить группу «proxygroup», сделать эту группу основной (нажать кнопку «Set as primary»), удалить группу «Domain Users». Тем самым мы добились того, что пользователь «proxyuser» не обладает никакими правами на объекты каталога.

8. Настроить ограниченные права пользователя «proxyuser» на объекты каталога:
   a. Запустить оснастку ADSI Edit (Start ⇒ Run adsiedit.msc).
   b. Перейти на закладку «Security» свойств контейнера «Users», нажать кнопку «Advanced».
   c. В появившемся окне нажать кнопку «Add». Указать «proxygroup» в качестве объекта.
   d. В появившемся окне выбрать в разделе «Apply onto» вариант «This object and all child objects».
   e. Назначить права «List contents» и «Read all properties».
9. Запустить виртуальную машину WS-Linux.
10. Получить список групп, членом которых является пользователь «Administrator», зарегистрировавшись на LDAP-сервере с правами пользователя «proxyuser». Для чего на виртуальной машине WS-Linux запустить команду:

```
ldapsearch -h dc -b cn=Users,dc=example,dc=com -D
cn=proxyuser,cn=Users,dc=example,dc=com -w P@ssw0rd
-x "(&(objectClass=user)(cn=Administrator))"
memberOf
```

11. Убедиться, что список групп присутствует в выводимой на экран информации.

7.3. Система единого входа в сеть на основе протокола Kerberos

7.3.1. Общие сведения о протоколе Kerberos

Протокол Kerberos является универсальным протоколом аутентификации, основанным на распределении симметричных ключей шифрования неко-

торым доверенным сервером. Протокол Kerberos является открытым стандартом и описан в документе RFC 1510.

С помощью протокола Kerberos распределяются криптографические ключи в некоторой *области* (realm), которая имеет строковый идентификатор, как правило, совпадающий с именем DNS-домена. Например, для компьютеров DNS-домена example.com можно определить область Kerberos EXAMPLE.COM (имя области всегда заглавными буквами).

Каждая учетная запись в системе Kerberos называется *сущностью* (principal), причем учетные записи существуют не только для пользователей, но и для каждого сервиса. Имя учетной записи имеет следующий вид: «user@REALM» (где user — имя пользователя, а RELAM — имя области). Например, имя учетной записи пользователя root из домена «example.com» — «root@EXAMPLE.COM».

Учетные записи сервисов имеют вид <name1>/<name2>@REALM, где name1 — как правило, имя сервиса, а name2 — полное DNS-имя компьютера. Например, «HTTP/ws-linux.exampel.com@EXAMPLE.COM». Важно отметить, что имена сущностей Kerberos чувствительны к регистру.

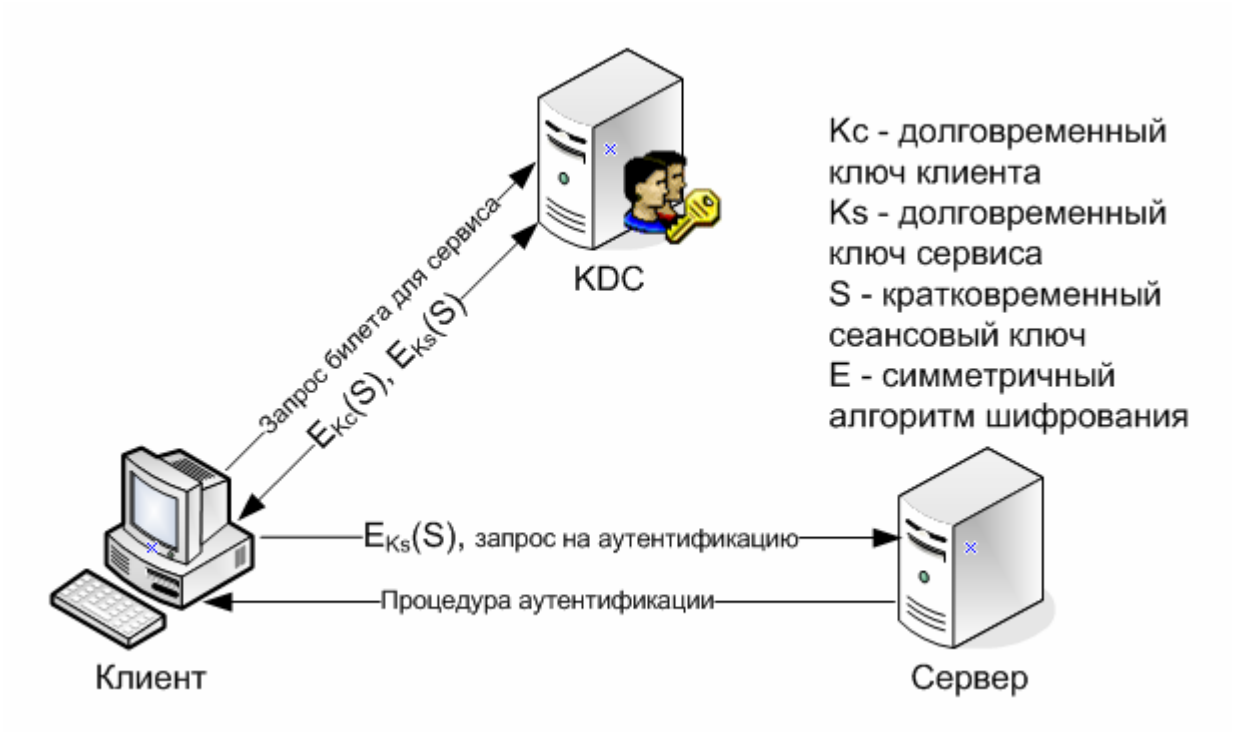


Рис. 7.6. Аутентификация по протоколу Kerberos

Информация обо всех учетных записях, а также их *долговременные ключи* (в случае пользователя это, например, хэш его пароля, т. е. ключ, который будет действителен не меньше месяца) хранятся на специальном сервере, называемом *центром распределения ключей* (Key Distribution Centre — KDC). Когда у пользователя появляется необходимость в использовании некоторого сервиса, то он обращается в центр распределения ключей с запросом кратко-

временного *сеансового ключа*. KDC возвращает ключ в двух вариантах — один зашифрован долговременным ключом пользователя, другой зашифрован долговременным ключом сервиса. Ответ KDC называется *билетом*, поскольку кроме ключа содержит и некоторую дополнительную информацию. После этого пользователь самостоятельно пересылает билет сервиса, и у них появляется некоторый общий секрет, который уже можно использовать как для аутентификации, так и для защиты канала (рис. 7.6).

Применение доверенного сервера, который хранит секреты всех своих пользователей, дает возможность обойтись без асимметричных алгоритмов шифрования, что делает протокол Kerberos более легким в реализации и управлении.

На приведенной схеме видно, что каждый сервис должен хранить собственный долговременный ключ. В Unix-системах ключи сервисов хранятся в так называемом *keytab*-файле (понятно, что данный файл не должен быть доступен для чтения всем пользователям), в ОС Windows ключ хранится в локальной базе учетных записей SAM (стандартными средствами просмотреть информацию о ключе невозможно).

Долговременный ключ пользователя, как уже упоминалось выше, — это, например, некоторая хэш-функция его пароля. Таким образом, в приведенной выше схеме пользователь все равно должен вводить свой пароль при обращении к очередному сервису. Для обеспечения реальной возможности однократной аутентификации схема взаимодействия клиента, сервера и KDC реально несколько отличается от приведенной на рис. 7.6. В процессе аутентификации при входе в сеть пользователь получает специальный билет, называемый *ticket granting ticket (TGT)*, использующийся для аутентификации пользователя в KDC (рис. 7.7).

Полученные пользователем билеты сохраняются в специальном защищенном кэше (в случае Unix-систем это файл, доступ к которому имеет только пользователь, получивший билет, в Windows — это защищенное хранилище модуля SSPI, т. е. оперативная память). При необходимости билет может переместиться на другой компьютер (например, если пользователь открыл сеанс по SSH или RDP), но при этом действительным он останется только в том случае, если в нем присутствует специальный флаг (соответственно, на KDC можно задавать, какие из пользователей будут получать этот флаг в билете, а какие нет).

Из изложенного выше механизма следует, что KDC не аутентифицирует клиента перед самим собой. Аутентификация есть только неявная — смог ли клиент воспользоваться тем сеансовым ключом, который он получил, или нет. Это создает две проблемы: во-первых, злоумышленник может получить билет пользователя и за какое-то время извлечь сеансовый ключ (например, если применялся слабый алгоритм шифрования); во-вторых, злоумышленник может запросить TGT и после пытаться взломать долговременный ключ пользователя.

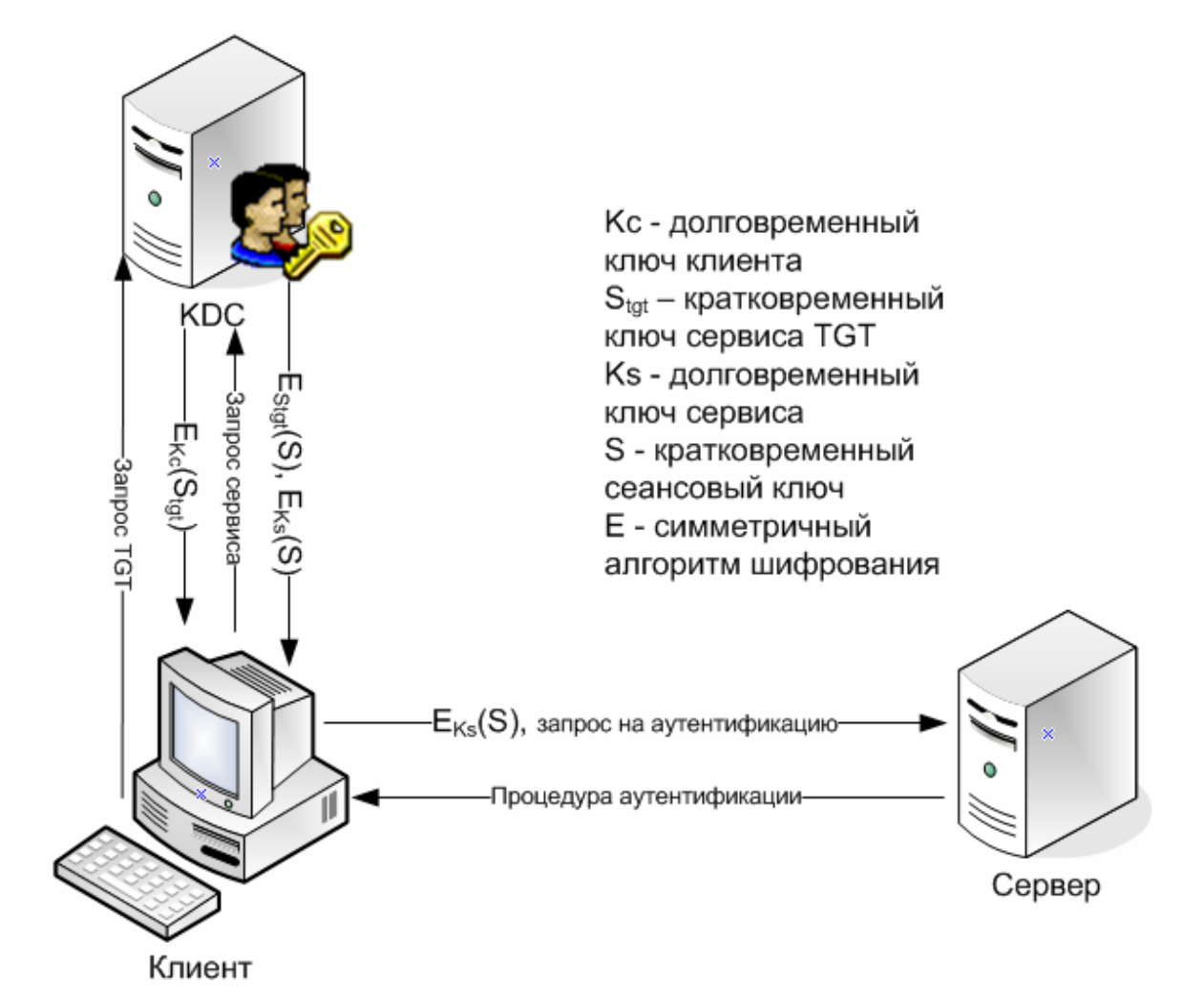


Рис. 7.7. Механизм однократной аутентификации при помощи вспомогательного билета TGT

Первой проблемы на самом деле не существует, поскольку билет Kerberos содержит информацию о своем времени жизни (по умолчанию 8 часов). Поэтому к тому времени, когда злоумышленник сможет извлечь сеансовый ключ, последний окажется уже недействительным.

Вторая проблема решается при помощи механизма *преаутентифкации.* Данный механизм не зафиксирован в RFC 1510, т. е. каждый производитель может добавлять свои механизмы, однако приведен один вариант, который наиболее широко распространен в настоящее время — это преаутентификация по меткам времени. Работает данный механизм следующим образом: клиент в своем запросе, в специальном разделе пакета Kerberos, отправляет зашифрованную своим долговременным ключом метку времени, сервер расшифровывает эту метку времени и сверяет ее со своим текущим временем. Если разница не превышает некоторого порогового значения (по умолчанию 3 минуты), то считается, что клиент прошел преаутентифкацию, и ему можно выдавать

билет. Понятно, что данный механизм требует синхронизации времени между всеми клиентами Kerberos и всеми KDC.

7.3.2. Реализация Kerberos для Unix-систем

В Unix-системах распространены две схожие реализации протокола Kerberos — это MIT Kerberos и Heimdal Kerberos. Реализации во многом схожи, поэтому рассмотрим канонический вариант[1] от MIT.

Все библиотеки и утилиты, имеющие отношение к Kerberos, распределены по нескольким пакетам, в случае Debian Linux это следующие пакеты:

- libkrb5 — библиотеки Kerberos (оригинальный интерфейс MIT и GSSAPI интерфейс),
- krb5-config — набор примеров конфигурационных файлов Kerberos,
- krb5-doc — документация по Kerberos,
- krb5-user — набор утилит для управления билетами пользователя (получение, уничтожение, просмотр),
- krb5-clients — некоторые клиенты классических сетевых сервисов: ftp, telnet, rsh, rcp и т.п.,
- krb5-servers — сервер KDC, сервис kadmind для удаленного управления базой KDC.

Основой всей системы MIT Kerberos являются, конечно же, библиотеки, которые должны быть правильно настроены для своего функционирования при помощи конфигурационного файла «krb5.conf». Данный файл имеет структуру типичного ini-файла, т.е. он поделен на разделы (каждый раздел начинается с имени, заключенного в прямые скобки), внутри раздела перечислены пары <опция> = <значение>. Основными разделами файла krb5.conf являются:

- libdefaults — различные значения по умолчанию,
- realms — раздел, описывающий информацию об областях Kerberos. Состоит из записей вида <имя области> = { набор опций в виде <опция> = значение },
- domain_realm — эта секция описывает принадлежность узла (или целого DNS-домена) к области Kerberos. Она содержит строки вида: name = REALM, где name — может быть имя хоста или имя домена, имя домена выделяется ведущей точкой (т.е., если указать example.com, то это обозначает одноименный узел, а если — .example.com, то весь DNS-домен example.com). Если имя области — это просто имя DNS-домена в верхнем регистре, то соответствующую запись можно опустить.

Опции раздела libdefaults:

- ticket_lifetime — время жизни билета в секундах (не может превышать максимального времени жизни, разрешенного на KDC);

[1] Дело в том, что Kerberos был разработан в Массачусетском Технологическом Институте (MIT)

- default_realm — имя области Kerberos по умолчанию;
- dns_lookup_kdc — разрешить поиск KDC при помощи SRV-записей dns (будет производиться поиск записи _kerberos._udp.<имя домена> или _kerberos._tcp.<имя домена>);
- default_tgs_enctype и default_tkt_enctype — какие алгоритмы шифрования будут запрашиваться для билетов и ключей;
- kdc_timesync — если значение отлично от 0, то библиотека Kerberos будет пытаться синхронизировать время локальной системы с временем на KDC, однако если время отличается слишком сильно, и на KDC применяется преаутентификация, то синхронизировать время не удастся;
- default_keytab_name — задает имя keytab-файла (файла, хранящего долговременные ключи сервисов). По умолчанию используется «/etc/krb5.keytab».

Опции раздела realms:

- kdc — IP-адрес или имя KDC (также допускается указание порта KDC через двоеточие, по умолчанию используется порт 88 TCP или UDP);
- admin_server — IP-адрес или имя сервера с сервисом kadmind, также может содержать номер порта.

Управление билетами пользователь осуществляет через 3 основные утилиты: kinit — для получения билетов, klist — для просмотра билетов в кэше и kdestroy — для уничтожения билетов.

Утилита kinit, вызванная без параметров, старается получить на KDC TGT для сущности <имя пользователя>@REALM, где REALM берется из параметра default_realm файла «krb5.conf». Кроме того, возможен вызов утилиты kinit следующим образом: `kinit user@REALM`. Такой вызов позволяет запросить TGT для произвольной сущности Kerberos. Опция -k, указывает, что получать билет для указанной сущности необходимо при помощи ключа, сохраненного в keytab-файле, опция -t позволяет задать расположение keytab-файла, если оно отличается от заданного в «krb5.conf».

Утилита klist выводит список всех билетов, полученных данным пользователем, или информацию о содержимом keytab-файла. Опция -e позволяет посмотреть типы шифрования для билета и ключа, опция -f выводит флаги каждого билета. Для просмотра содержимого keytab-файла применяется опция -k, если добавить опцию -K, то кроме информации о сущностях в keytab-файле будут выведены также и сами ключи.

Утилита kdestroy очищает кэш текущего пользователя.

Для того чтобы в дальнейшем обеспечить аутентификацию пользователей рабочей станции ws-linux в домене AD, необходимо произвести настройку клиента Kerberos.

ВЫПОЛНИТЬ!

12. На виртуальной машине WS-Linux открыть текстовым редактором файл «/etc/krb5.conf», удалить имеющуюся в нем информацию и внести в него следующую:

```
[libdefaults]
ticket_lifetime = 36000
default_realm = EXAMPLE.COM
kdc_timesync=1
[realms]
EXAMPLE.COM = {
kdc = 192.168.0.1:88
}
```

13. На виртуальной машине DC запустить оснастку «Active Directory Users & Computers» (Start ⇒ Programs ⇒ Administration Tools).
14. В контекстном меню контейнера «Users» выбрать пункт «New ⇒ User». Указать «First Name» и «User logon name» — «root», указать пароль «P@ssw0rd», убедиться, что не отмечен пункт «User must change password at next logon».
15. В консоли виртуальной машины WS-Linux выполнить команду kinit от имени пользователя «root», на запрос ввода пароля ввести «P@ssw0rd». После чего ввести команду klist и убедиться, что ее вывод аналогичен выводу, изображенному на рис. 7.8.

```
root@ws-linux:~# kinit
Password for root@EXAMPLE.COM:
root@ws-linux:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: root@EXAMPLE.COM

Valid starting     Expires            Service principal
01/22/07 18:34:43  01/23/07 04:35:32  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 01/23/07 18:34:43


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
root@ws-linux:~#
```

Рис. 7.8. Вывод билетов пользователя в ОС Linux

7.3.3. Реализация Kerberos в ОС Windows Server 2003

Клиент Kerberos встроен во все ОС Windows семейства NT5, а реализация KDC — во все серверные версии NT5. Однако изначально реализация Kerberos не рассматривалась Microsoft как самостоятельное решение (а лишь только как средство аутентификации в AD), поэтому в стандартной поставке Windows отсутствуют утилиты для работы с Kerberos напрямую. Некоторые возможности предоставляют инструменты из пакетов Support Tools и Resource Kit.

Служба KDC активизируется в серверной ОС только при повышении роли сервера до контроллера домена AD. Как уже отмечалось, контроллер до-

мена AD — это KDC и LDAP-серверы, плюс некоторые утилиты администрирования этих серверов. Сущность Kerberos создается параллельно с созданием учетной записи пользователя, при этом происходит прямое отображение имени пользователя в сущность Kerberos (пользователю user домена example.com будет сопоставлена сущность «user@EXAMPLE.COM»).

Сложнее обстоит ситуация с учетными записями сервисов. Поскольку имя пользователя Windows не может содержать символ «/», то прямого соответствия быть не может. Для разрешения этой проблемы в схеме каталога определен специальный многозначный атрибут — servicePrincipalName. Минус заключается в том, что все эти сущности Kerberos используют один и тот же ключ.

Для обеспечения взаимодействия с Unix-системами пакет Windows Support tools содержит утилиту для создания сущностей Kerberos и экспортирования их ключей в виде keytab-файла. Данная утилита обладает достаточно большим количеством параметров, рассмотрим те из них, которые необходимы при создании keytab-файла.

Вызов утилиты будет иметь вид:

```
ktpass –princ <сущность> -crypto des-cbc-md5 +desOnly
–out <keytab-файл> +rndPass -mapuser <user> -mapop set
```

Ниже приведены комментарии по каждому из параметров:
-princ <сущность> — задает сущность Kerberos;
-crypto — задает, для какой криптосистемы следует генерировать ключ (в настоящее время реализации от Microsoft и MIT пересекаются только по режиму шифрования des-cbc-md5);
+desOnly — указывает, что в базе KDC должен генерироваться ключ только для алгоритма DES;
-out <keytab файл> — имя keytab-файла;
+rndPass — генерация случайного ключа;
-mapuser <user> — указывает, с каким пользователем AD связать данную сущность Kerberos;
-mapop set — указывает, что необходимо заменить сущность по умолчанию (а не добавить к списку servicePrincipalName).

Для мониторинга кэша билетов Kerberos используется утилита kerbtray из пакета Windows Resource Kit. После запуска эта утилита размещает свою иконку в системном трее, и по двойному нажатию левой клавиши мыши выводит на экран окно со списком билетов пользователя и информации в них.

7.3.4. Пример реализации системы SSO

Применение протокола Kerberos для аутентификации достаточно распространено как в сервисах Microsoft, так и в сервисах других производителей. Например, Kerberos используется для аутентификации клиента перед

web-сервером. Рассмотрим ее на примере модуля mod_auth_kerb для web-сервера Apache.

Выполнение следующего задания создаст защищенную web-страницу на сервере Apache, при этом аутентификация для пользователей домена AD будет происходить прозрачно.

ВЫПОЛНИТЬ!

16. На виртуальной машине WS-Linux открыть текстовым редактором файл «/etc/apache2/sites-available/default», в разделе <Directory /var/www> указать следующие директивы (комментарии, указанные после символа «#», набирать не надо):

```
# подключить модуль mod_auth_kerb
AuthType Kerberos

# расположение keytab-файла
Krb5Keytab /etc/apache2/http.keytab

# разрешить аутентификацию
# при помощи сеансового ключа
KrbMethodNegotiate on

# запретить аутентификацию уровня Basic
KrbMethodK5Passwd off

# разрешить доступ только
# аутентифицированным пользователям
Require valid-user
```

17. На виртуальной машине DC запустить «Windows Support Tools Shell» (Start ⇒ Programs ⇒ Windows Support tools ⇒ Command Prompt)
18. Создать сущность Kerberos, соответствующую учетную запись и keytab-файл для web-сервера Apache. Для этого создать пользователя ws-linux_http при помощи оснастки «Active Directory Users and Computers». Создать файл «keytab» при помощи следующей команды:

```
ktpass –princ HTTP/ws-linux.example.com@EXAMPLE.COM
–crypto des-cbc-md5 +desOnly
–ptype KRB5_NT_PRINCIPAL
–out c:\http.keytab +rndPass
–mapuser ws-linux_http@example.com –mapop set
```

19. Скопировать файл «http.keytab» в каталог «/etc/apache2» виртуальной машины WS-Linux.
20. Перезапустить web-сервер Apache при помощи команды

```
/etc/init.d/apache2 force-reload
```

21. На виртуальной машине DC запустить и настроить обозреватель Internet Explorer. Для этого в меню Tools ⇒ Internet Options выбрать закладку «Security», выбрать зону «Local Intranet», нажать кнопку «Sites», нажать кнопку «Advanced», добавить к списку сайтов строку «*.example.com». В разделе «Security level for this zone» нажать кнопку «Custom level». Убедиться, что в подразделе «logon» раздела «User authentication» выбран пункт «Automatic logon only in Intranet zone». Перейти на закладку «Advanced». Убедиться, что в разделе «Security» активизирована опция «Enable Integrated Windows Authentication». Если она не была выбрана, то выбрать и перезагрузить виртуальную машину.
22. Запустить утилиту kerbtray.exe (Start ⇒ Programs ⇒ Windows Resource Kit Tools ⇒ Command Shell, набрать в консоли kerbtray). Просмотреть текущий список билетов пользователя, вызвав окно утилиты при помощи иконки в системном трее (иконка в виде билета зеленого цвета).
23. В адресной строке обозревателя набрать «http://ws-linux.example.com». Убедиться, что в обозревателе отобразилась стартовая страница web-сервера Apache.
24. Убедиться, что в списке билетов пользователя добавился билет «HTTP/ws-linux.example.com@EXAMPLE.COM».
25. На рабочей станции WS-Linux выполнить команду

```
tail /var/log/apache2/access.log
```

26. В выводе команды найти запись, аналогичную представленной на рис. 7.9.

```
192.168.0.1 - Administrator@EXAMPLE.COM [22/Jan/2007:18:58:43 +0500] "GET /apach
e2-default/ HTTP/1.1" 403 326 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
 5.2; SV1; .NET CLR 1.1.4322)"
```

Рис. 7.9. Запись в лог-файле web-сервера Apache о доступе Kerberos-сущности «Administrator@EXAMPLE.COM»

27. Попробовать зайти на web-страницу с основной рабочей станции (может понадобиться перенастройка IP-адреса виртуального сетевого подключения VMWare VMNet1, адрес должен быть из сети 192.168.0/24), убедиться, что в окне обозревателя выводится сообщение об ошибке 401 (Unauthorized).

7.4. Создание единого пространства безопасности на базе Active Directory

Рассмотренные выше технологии позволяют создать распределенную систему с единой базой аутентификационных и авторизующих данных. Аутентификационные данные — это сущности Kerberos, которые связаны с объектами LDAP, хранящими авторизующую информацию (идентификаторы пользователей, членство в группах и т. п.).

Операционные системы семейства Windows встраиваются в среду Active Directory при помощи стандартного инструментария, при этом на рабочей станции запускаются необходимые сервисы, которые переадресуют запросы к локальной базе SAM на контроллеры домена (так же активизируется SSPI-модуль, отвечающий за аутентификацию по протоколу Kerberos). Для того чтобы увидеть этот процесс более подробно, рассмотрим интеграцию рабочей станции под управлением ОС Linux в среду Active Directory.

7.4.1. Технология PAM

В первую очередь необходимо сменить режим аутентификации рабочей станции с традиционного (через файл «/etc/shadow»), на аутентификацию по протоколу Kerberos.

Для решения подобных задач уже довольно долгое время в Linux (и некоторых других Unix-системах) система аутентификации отделена от утилит, которым необходимо производить аутентификацию. Каждая такая утилита (например, login) запрашивает специальную библиотеку libpam для проведения аутентификации пользователя. Все, что должен обеспечить сервис, — это обратную связь между пользователем и библиотекой (для запроса пароля при необходимости, вывода служебных сообщений, например с просьбой приложить специальное устройство к считывателю и т. п.). Сама библиотека libpam тоже не содержит конкретного функционала по аутентификации, весь функционал содержится в модулях PAM[1]. О том, какие модули необходимо загружать, библиотека узнает из файлов конфигурации, расположенных в каталоге «/etc/pam.d» (по файлу конфигурации на каждый сервис).

Файл конфигурации может содержать 4 типа записей:

- auth — отвечают за процесс аутентификации;
- account — отвечают за то, что учетная запись актуальна (не заблокирована, не истек срок действия пароля и т. п.);
- password — отвечают за манипуляции с аутентификационными данными (например, смена пароля);
- session — выполняют некоторые действия, связанные с сеансом пользователя (установка переменных окружения, удаление кэша при выходе пользователя и т. п.).

Модуль аутентификации по протоколу Kerberos должен присутствовать в разделах auth и session (для создания кэша билетов и его удаления). Для того чтобы не было необходимости переписывать конфигурационные файлы для каждого из сервисов, в любом дистрибутиве Linux предусмотрены общие настройки PAM для всех сервисов. В дистрибутиве Debian — это файлы comon-auth, common-password, common-account и common-session, в каждом описаны умолчания для соответствующих разделов. Именно сюда и необходимо добавить модуль «pam_krb5.so».

[1] PAM (Pluggable Authentication Modules) – подключаемые модули аутентификации.

ВЫПОЛНИТЬ!

28. На виртуальной машине WS-Linux обязательно произвести вход от имени суперпользователя в две или более виртуальных консоли.
29. Открыть в текстовом редакторе файл «/etc/pam.d/common-auth».
30. Поставить символ «#» в начале строки, подключающей модуль pam_unix.so (традиционный механизм, основанный на файлах «/etc/passwd» и «/etc/shadow»).
31. Добавить строку «auth required pam_krb5.so».
32. На виртуальной машине DC создать пользователя «ws-linux_host» при помощи оснастки «Active Directory Users and Computers».
33. Создать keytab-файл при помощи следующей команды:

```
ktpass –princ host/ws-linux.example.com@EXAMPLE.COM
–crypto des-cbc-md5 +desOnly
–ptype KRB5_NT_PRINCIPAL
–out c:\http.keytab +rndPass
–mapuser ws-linux_host@example.com –mapop set
```

34. Скопировать keytab-файл на виртуальную машину WS-Linux под именем «/etc/krb5.keytab».
35. Сменить права доступа на keytab-файл командой:

```
chmod 400 /etc/krb5.keytab.
```

36. На одной из виртуальных консолей произвести выход из системы и вход в нее. Убедиться при помощи команды klist, что пользователь получил TGT.

После того как модуль pam_krb5 был подключен к системе аутентификации, получилось следующее:

- пользователь root способен осуществить вход в систему;
- пользователь student не способен войти в систему, поскольку он не обладает учетной записью Kerberos;
- пользователь Administrator способен пройти аутентификацию, но он отвергается нижележащими модулями, которые не могут получить про него авторизующую информацию (членство в группах, идентификатор, домашний каталог, login shell).

Для того чтобы обеспечить вход для пользователя student достаточно создать для него соответствующую учетную запись в Active Directory. Чтобы разрешить вход администратору, надо каким-то образом перенаправить остальные модули на получение информации о пользователе не из файла «/etc/passwd», а из LDAP каталога Active Directory.

7.4.2. Технология NSS

С целью универсального изменения источников различных данных для приложений (информация о пользователях, группах, компьютерах и т. п.) в Unix-системах используется технология Name Service Switch (NSS). Техноло-

гия эта похожа на PAM, т.е. при выполнении определенных стандартных функций специальная библиотека (здесь это библиотека языка C) переадресовывает вызов некоторому специальному модулю, который указывается в конфигурационном файле.

Настройка системы NSS производится через файл «/etc/nsswitch.conf», который содержит информацию об объектах и источниках данных о них. Например, в файле содержится строка hosts, которая задает поведение функции gethostbyname. В качестве источника данных для этой функции указаны file и dns, т. е. сначала будет произведен поиск в файле «/etc/hosts», а потом при помощи системы DNS.

Для того чтобы обеспечить получение авторизующей информации из Active Directory, необходимо изменить настройки для объектов users и groups, указав в качестве источника ldap. Однако соответствующая библиотека также требует определенной настройки, а именно ей необходимо указать, как интерпретировать объекты в нашей схеме LDAP.

ВЫПОЛНИТЬ!

37. Настройка модуля nss_ldap на виртуальной машине ws-linux
    a. Открыть в текстовом редакторе файл «/etc/libnss-ldap.conf»
    b. Указать следующие параметры

```
# адрес ldap сервера
host 192.168.0.1

# база поиска
base dc=example,dc=com

# версия протокола
ldap_version 3

# учетная запись для соединения с сервером
binddn cn=proxyuser,cn=users,dc=example,dc=com

# пароль учетной записи
bindpw P@ssw0rd

# глубина поиска
scope sub

# Настройки схемы:
# контейнер с объектами с информацией о пользователе
nss_base_passwd CN=users,DC=example,DC=com

# контейнер с информацией о группах
nss_base_group CN=users,DC=example,DC=com

# указать аналог класса posixAccount
# (класс posixAccount содержится в стандартной схеме,
# определенной в RFC 2307)
nss_map_objectclass posixAccount user

# аналог атрибута uid
nss_map_attribute uid sAMAccountName
```

```
# аналог атрибута homeDirectory
nss_map_attribute homeDirectory msSFUHomeDirectory

# указать аналог класса posixGroup
nss_map_objectclass posixGroup group

# аналог атрибута uniqueMember
nss_map_attribute uniqueMember member
```

Остальные параметры оставить по умолчанию.

c. Открыть текстовым редактором файл «/etc/nsswitch.conf». Добавить метод ldap в строках passwd и group.

38. Задать атрибуты для Unix систем пользователя root.
    a. Запустить оснастку ADSI Edit.
    b. Открыть редактор атрибутов для группы Domain Users. Задать атрибуту gidNumber значение 2000.
    c. Открыть редактор атрибутов для пользователя root. Задать следующие значения атрибутов: uid — 0, gidNumber — 2000, loginShell — «/bin/bash», msSFUHomeDirectory — «/root».
39. С помощью текстового редактора удалить запись о пользователе root из файла «/etc/passwd».
40. Проверить получение авторизующей информации из каталога AD.
    a. Выполнить команду getent passwd и убедиться, что пользователь root присутствует в выводе.
    b. Выполнить команду getent group и убедиться, что в выводе присутствует группа Domain Users.
    c. Выполнить команду id root и убедиться, что такой пользователь существует, его uid — 0, основная группа — Domain Users.
41. Конфигурация PAM.
    a. Открыть в текстовом редакторе файл «/etc/pam.d/common-auth» и поставить символ «#» в начале строки, описывающей модуль pam_unix.
    b. Перейти в новую виртуальную консоль и убедиться, что пользователь root по-прежнему способен войти в систему.
42. Сделать также учетные записи student и Administrator Unix-пользователями (учтите, что пользователю Administrator необходимо создать домашний каталог).

8. АУДИТ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

8.1. Понятие аудита информационной безопасности

Аудит информационной безопасности (ИБ) представляет собой одно из наиболее актуальных и динамично развивающихся направлений стратегического и оперативного менеджмента в области безопасности КС и вызывает постоянный интерес специалистов. Его основная задача — объективно оценить текущее состояние ИБ организации, а также ее адекватность поставленным целям и задачам бизнеса.

Под аудитом ИБ понимается системный процесс получения объективных качественных и количественных оценок текущего состояния ИБ организации в соответствии с определенными критериями и показателями на всех основных уровнях обеспечения безопасности: нормативно-методологическом, организационно-управленческом, процедурном и программно-техническом [19].

Результаты квалифицированно выполненного аудита ИБ организации позволяют построить оптимальную по эффективности и затратам систему обеспечения информационной безопасности (СОИБ), представляющую собой комплекс технических средств, а также процедурных, организационных и правовых мер, объединенных на основе модели управления ИБ.

В результате проведения аудита могут быть получены как качественные, так и количественные оценки. При качественном оценивании, например, может быть приведен перечень уязвимостей в программно-аппаратном обеспечении с их классификацией по трехуровневой шкале опасности: высокая, средняя и низкая. Количественные оценки чаще всего применяются при оценке риска для активов организации, создаваемого угрозами безопасности. В качестве количественных оценок могут выступать, например, цена риска, вероятность риска, размер риска и т. п.

Объективность аудита обеспечивается, в частности, тем, что оценка состояния ИБ производится специалистами на основе определенной методики, позволяющей объективно проанализировать все составляющие СОИБ.

Аудит ИБ может представлять собой услугу, которую предлагают специализированные фирмы, тем не менее в организации должен проводиться внутренний аудит ИБ, выполняемый, например, администраторами безопасности.

Традиционно выделяют три типа аудита ИБ, которые различаются перечнем анализируемых компонентов СОИБ и получаемыми результатами:

- активный аудит;
- экспертный аудит;
- аудит на соответствие стандартам ИБ.

8.1.1. Активный аудит

Активный аудит представляет собой обследование состояния защищенности определенных подсистем информационной безопасности (ПИБ), относящихся к программно-техническому уровню. Например, вариант активного аудита, называемый тестом на проникновение (Penetration test), предполагает обследование подсистемы защиты сетевых взаимодействий. Активный аудит включает:

- анализ текущей архитектуры и настроек элементов ПИБ;
- интервьюирование ответственных и заинтересованных лиц;
- проведение инструментальных проверок, охватывающих определенные ПИБ.

Анализ архитектуры и настроек элементов ПИБ проводится специалистами, обладающими знаниями по конкретным подсистемам, представленным в обследуемой системе (например, могут требоваться специалисты по активному сетевому оборудованию фирмы Cisco или по ОС семейства Microsoft), а также системными аналитиками, которые выявляют возможные изъяны в организации подсистем. Результатом этого анализа является набор опросных листов и инструментальных тестов.

Опросные листы используются в процессе интервьюирования лиц, отвечающих за администрирование АИС, для получения субъективных характеристик АИС, для уточнения полученных исходных данных и для идентификации некоторых мер, реализованных в рамках СОИБ. Например, опросные листы могут включать вопросы, связанные с политикой смены и назначения паролей, жизненным циклом АИС и степенью критичности отдельных ее подсистем для АИС и бизнес-процессов организации в целом.

Параллельно с интервьюированием проводятся инструментальные проверки (тесты), которые могут включать следующие мероприятия:

- визуальный осмотр помещений, обследование системы контроля доступа в помещения;
- получение конфигураций и версий устройств и ПО;
- проверка соответствия реальных конфигураций предоставленным исходным данным;
- получение карты сети специализированным ПО;
- использование сканеров защищенности (как универсальных, так и специализированных);
- моделирование атак, использующих уязвимости системы;
- проверка наличия реакции на действия, выявляемые механизмами обнаружения и реагирования на атаки.

Аудитор может исходить из следующих моделей, описывающих степень его знания исследуемой АИС (модель знания):

- модель «черного ящика» – аудитор не обладает никакими априорными знаниями об исследуемой АИС. Например, при проведении внешнего актив-

ного аудита (то есть в ситуации, когда моделируются действия злоумышленника, находящегося вне исследуемой сети), аудитор может, зная только имя или IP-адрес web-сервера, пытаться найти уязвимости в его защите;

− модель «белого ящика» – аудитор обладает полным знанием о структуре исследуемой сети. Например, аудитор может обладать картами и диаграммами сегментов исследуемой сети, списками ОС и приложений. Применение данной модели не в полной мере имитирует реальные действия злоумышленника, но позволяет, тем не менее, представить «худший» сценарий, когда атакующий обладает полным знанием о сети. Кроме того, это позволяет построить сценарий активного аудита таким образом, чтобы инструментальные тесты имели минимальные последствия для АИС и не нарушали ее нормальной работы;

− модель «серого ящика» или «хрустального ящика» – аудитор имитирует действия внутреннего пользователя АИС, обладающего учетной записью доступа в сеть с определенным уровнем полномочий. Данная модель позволяет оценить риски, связанные с внутренними угрозами, например от неблагонадежных сотрудников компании.

Аудиторы должны согласовывать каждый тест, модель знания, применяемую в тесте, и возможные негативные последствия теста с лицами, заинтересованными в непрерывной работе АИС (руководителями, администраторами системы и др.).

По результатам инструментальной проверки проводится пересмотр результатов предварительного анализа и, возможно, организуется дополнительное обследование (рис. 8.1).

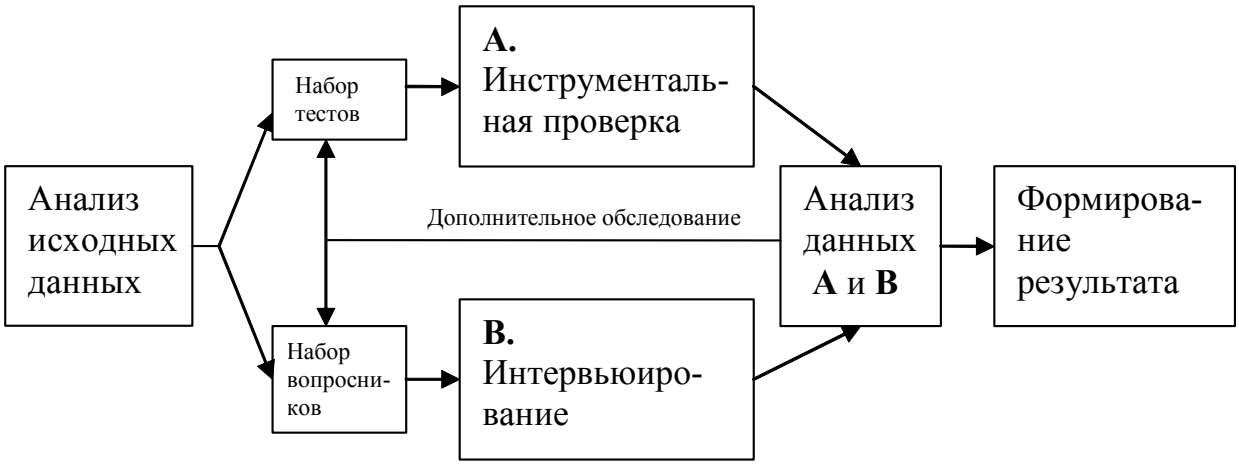


Рис. 8.1. Схема проведения активного аудита ИБ

По результатам активного аудита создается аналитический отчет, состоящий из описания текущего состояния технической части СОИБ, списка найденных уязвимостей АИС со степенью их критичности и результатов упрощенной оценки рисков, включающей модель нарушителя и модель угроз.

Дополнительно может быть разработан план работ по модернизации технической части СОИБ, состоящий из перечня рекомендаций по обработке рисков.

8.1.2. Экспертный аудит

Экспертный аудит предназначен для оценивания текущего состояния ИБ на нормативно-методологическом, организационно-управленческом и процедурном уровнях. Экспертный аудит проводится преимущественно внешними аудиторами, его выполняют силами специалистов по системному управлению. Сотрудники организации-аудитора совместно с представителями заказчика проводят следующие виды работ:

- сбор исходных данных об АИС, ее функциях и особенностях, используемых технологиях автоматизированной обработки и передачи информации (с учетом ближайших перспектив развития);
- сбор информации об имеющихся организационно-распорядительных документах по обеспечению ИБ и их анализ;
- определение защищаемых активов, ролей и процессов СОИБ.

Важнейшим инструментом экспертной оценки является сбор данных об АИС путем интервьюирования технических специалистов и руководства заказчика.

Основные цели интервьюирования руководящего состава организации:

- определение политики и стратегии руководства в вопросах обеспечения ИБ;
- выявление целей, которые ставятся перед СОИБ;
- выяснение требований, которые предъявляются к СОИБ;
- получение оценок критичности тех или иных подсистем обработки информации, оценок финансовых потерь при возникновении тех или иных инцидентов.

Основные цели интервьюирования технических специалистов:

- сбор информации о функционировании АИС;
- получение схемы информационных потоков в АИС;
- получение информации о технической части СОИБ;
- оценка эффективности работы СОИБ.

В рамках экспертного аудита проводится анализ организационно-распорядительных документов, таких как политика безопасности, план защиты, различного рода положения и инструкции. Организационно-распорядительные документы оцениваются на предмет достаточности и непротиворечивости декларируемым целям и мерам ИБ, а также на предмет соответствия стратегической политике руководства в вопросах ИБ.

Результаты экспертного аудита могут содержать рекомендации по совершенствованию нормативно-методологических, организационно-управленческих и процедурных компонентов СОИБ.

8.1.3. Аудит на соответствие стандартам ИБ

В ряде случаев проводится аудит на соответствие стандартам ИБ. Специально уполномоченные организации-аудиторы по результатам аудита принимают решение и выдают документальное подтверждение о соответствии СОИБ тому или иному эталонному стандарту (проводят сертификацию). Сертификация является показателем качества СОИБ и поднимает престиж и уровень доверия к организации.

Аудит на соответствие стандартам чаще всего подразумевает проведение активного и экспертного аудита. По результатам могут быть подготовлены отчеты, содержащие следующую информацию:

- степень соответствия проверяемой ИС выбранным стандартам;
- количество и категории полученных несоответствий и замечаний;
- рекомендации по построению или модификации СОИБ, позволяющие привести ее в соответствие с требованиями рассматриваемого стандарта.

8.2. Методика проведения инструментальных проверок

Инструментальные проверки (ИП) выполняются в процессе активного аудита ИБ. Как уже было отмечено, ИП состоят из набора заранее согласованных тестов, направленных на получение характеристик об уровне защищенности выбранных ПИБ. Для проведения инструментальных проверок может быть предложена следующая методика, предполагающая тестирование возможности несанкционированного доступа (НСД) к информации, обрабатываемой или хранящейся в АИС, как изнутри организации, так и из внешних сетей. Методика включает три этапа: анализ структуры АИС, внутренний аудит, внешний аудит.

На этапе анализа структуры АИС с позиций ИБ производится анализ и инвентаризация информационных ресурсов и СВТ: формируется перечень защищаемых сведений; описываются информационные потоки, структура и состав АИС; проводится категорирование ресурсов, подлежащих защите.

На втором этапе осуществляется внутренний аудит АИС, включающий анализ настроек АИС с точки зрения ИБ. На данном этапе с учетом известных изъянов ОС и специализированных СЗИ осуществляется анализ защищенности от опасных внутренних воздействий. Исследуется возможность несанкционированных действий легальных пользователей компьютерной сети, которые могут привести к модификации, копированию или разрушению конфиденциальных данных. Анализ осуществляется путем детального изучения настроек безопасности средств защиты с использованием как общеупотребимых (в том числе входящих в арсенал хакеров), так и специально разработанных автоматизированных средств исследования уязвимости АИС. Анализируются следующие компоненты АИС:

− средства защиты ПК — возможность отключения программно-аппаратных систем защиты при физическом доступе к выключенным станциям; использование и надежность встроенных средств парольной защиты BIOS;

− состояние антивирусной защиты – наличие в АИС вредоносных программ, возможность их внедрения через машинные носители, сеть Интернет;

− ОС — наличие требуемых настроек безопасности;

− парольная защита в ОС — возможность получения файлов с зашифрованными паролями и их последующего дешифрования; возможность подключения с пустыми паролями, подбора паролей, в том числе по сети;

− система разграничения доступа пользователей АИС к ресурсам — формирование матрицы доступа; анализ дублирования и избыточности в предоставлении прав доступа; определение наиболее осведомленных пользователей и уровней защищенности конкретных ресурсов; оптимальность формирования рабочих групп;

− сетевая инфраструктура — возможность подключения к сетевому оборудованию для получения защищаемой информации путем перехвата и анализа сетевого трафика; настройки сетевых протоколов и служб;

− аудит событий безопасности — настройка и реализация политики аудита;

− прикладное ПО — надежность элементов защиты используемых АРМ; возможные каналы утечки информации; анализ версий используемого программного обеспечения на наличие уязвимых мест;

− СЗИ: надежность и функциональность используемых СЗИ; наличие уязвимых мест в защите; настройка СЗИ.

На третьем этапе осуществляется внешний аудит АИС, оценивающий состояние защищенности информационных ресурсов организации от НСД, осуществляемого из внешних сетей, в том числе из Интернет. Последовательно анализируются следующие возможности проникновения извне:

− получение данных о внутренней структуре АИС — наличие на web-серверах информации конфиденциального характера; выявление настроек DNS- и почтового серверов, позволяющих получить информацию о внутренней структуре АИС;

− выявление компьютеров, подключенных к сети и достижимых из Интернет — сканирование по протоколам ICMP, TCP, UDP; определение степени доступности информации об используемом в АИС ПО и его версиях; выявление активных сетевых служб; определение типа и версии ОС, сетевых приложений и служб;

− получение информации об учетных записях, зарегистрированных в АИС с применением утилит, специфичных для конкретной ОС.

− подключение к доступным сетевым ресурсам — определение наличия доступных сетевых ресурсов и возможности подключения к ним;

− использование известных уязвимостей в программном обеспечении МЭ, выявление неверной конфигурации МЭ.

− выявление версий ОС и сетевых приложений, подверженных атакам типа «отказ в обслуживании»;

− тестирование возможности атак на сетевые приложения — анализ защищенности web-серверов, тестирование стойкости систем удаленного управления, анализ возможности проникновения через имеющиеся модемные соединения.

По результатам тестирования оформляется экспертное заключение, описывающее реальное состояние защищенности АИС от внутренних и внешних угроз, содержащее перечень найденных изъянов в настройках систем безопасности. На основании полученного заключения разрабатываются рекомендации по повышению степени защищенности АИС, по администрированию систем, по применению СЗИ.

Реализация методики требует постоянного обновления знаний об обнаруживаемых изъянах в системах защиты. Не все этапы методики могут быть автоматизированы. Во многих случаях требуется участие эксперта, обладающего соответствующей квалификацией.

8.3. Постановка задачи для проведения инструментальных проверок

Проведение инструментальной проверки предлагается отработать на основе модели компьютерной сети в режиме «черного ящика». В моделируемой сети (рис. 8.2) используются:

− сервер ОС Windows Server 2003 с развернутыми web-сервером IIS, контроллером домена и DNS-сервером,

− сервер ОС Debian Linux с установленными web-сервером Apache и почтовым сервером,

− две рабочие станции ОС Windows XP,

− одна рабочая станция ОС Windows 2000.

Все программное обеспечение применяется без дополнительных настроек безопасности. При развертывании модели сети в компьютерном классе применяются два рабочих места с ОС Windows XP, на один из которых с применением технологии виртуальных машин разворачиваются образы ОС Windows Server 2003 и ОС Windows 2000, на другой — образ ОС Debian Linux. DNS-сервер, функционирующий на сервере, должен разрешать IP-адреса моделируемой сети.

Таким образом, моделируемая сеть (192.168.10.0/24) будет содержать пять сетевых узлов (табл. 8.1).

Для проведения проверки аудитору предлагается диапазон, состоящий из IP-адресов узлов моделируемой сети — 192.168.10.0/24. Считается, что количество узлов, тип и версии ОС и приложений не известны.

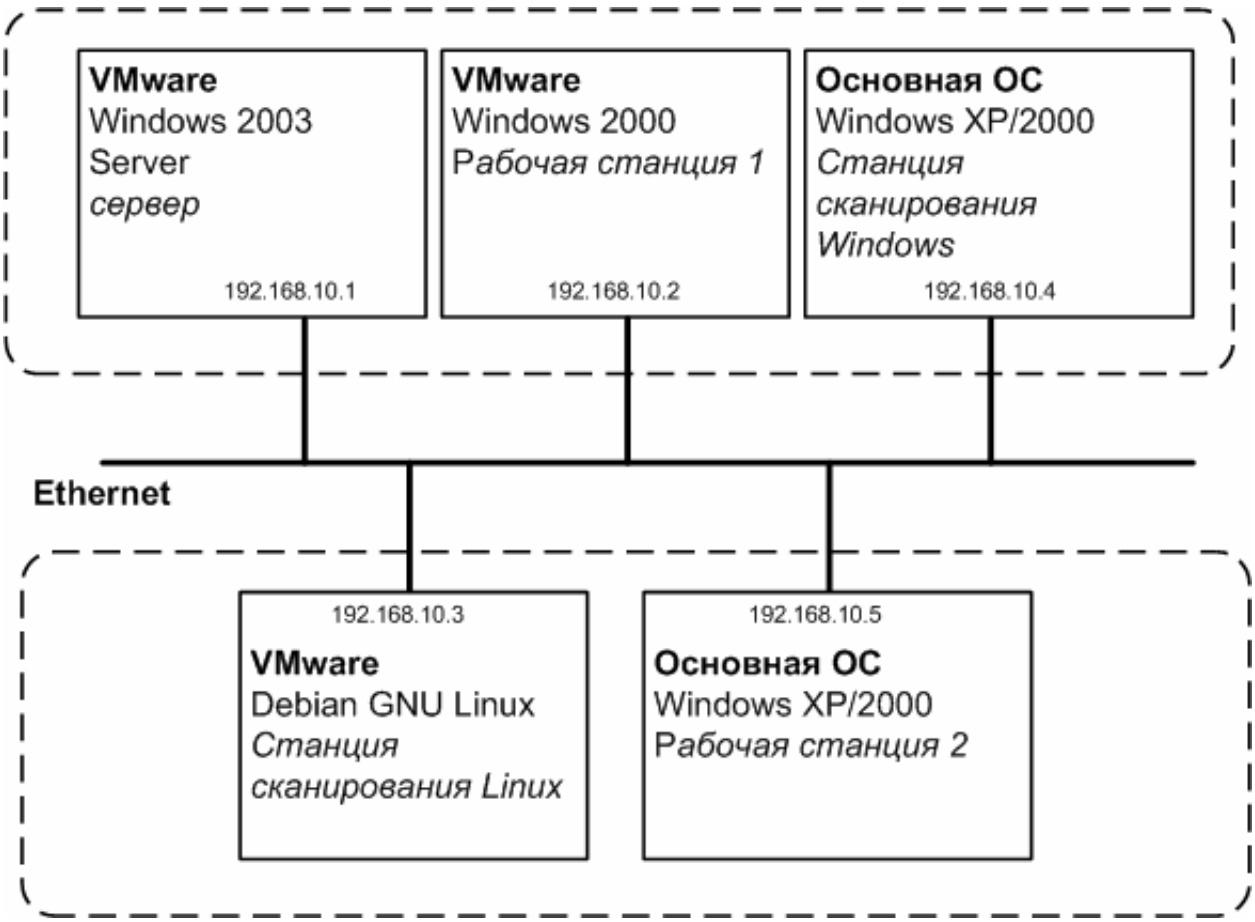


Рис. 8.2. Схема моделируемой сети

Таблица 8.1

Перечень сетевых узлов моделируемой сети

Наименование сетевого узла	ОС	IP – адрес	Сетевые сервисы и дополнительные утилиты
Сервер	Windows Server 2003	192.168.10.1	IIS, контроллер домена, DNS-сервер
Рабочая станция 1	Windows 2000 Professional	192.168.10.2	
Станция сканирования Linux	Debian GNU Linux	192.168.10.3	Apache, fping, icmpush, hping3, arping, tethereal, nmap, nessusd, и nikto
Станция сканирования Windows	Windows XP	192.168.10.4	nessuswx, cgichk, AdRem Netcrunch
Рабочая станция 2	Windows XP	192.168.10.5	

Задачей является построение карты, инвентаризация ресурсов исследуемой сети, а также выявление уязвимостей в программном обеспечении уз-

лов. В результате необходимо сформировать документ табличной формы, где должны быть приведены полученные сведения и характеристика степени защищенности сети.

Инструментальная проверка проводится в два этапа: на первом этапе выполняется идентификация сетевых ресурсов, на втором выявляются уязвимости в ПО сетевых узлов с применением сканеров безопасности.

Выполнение этапа обнаружения и получения информации о сетевых узлах позволяет понять, насколько актуальны данные о состоянии сети, полученные на этапе предварительного анализа, а также оценить, насколько быстро злоумышленник сможет произвести атаки, направленные на идентификацию сетевых ресурсов. На данном этапе аудитор должен получить следующую информацию:

- IP-адреса сетевых узлов и подсетей;
- открытые TCP- и UDP-порты на обнаруженных узлах;
- версии ОС и сетевых сервисов, работающих на обнаруженных сетевых узлах.

Эта информация может быть получена проведением пассивной и активной сетевой разведки. Пассивная разведка предполагает получение информации из общедоступных источников (web-сайт организации и иные web-сайты, новостные группы, базы whois и др.) и с применением методов социальной инженерии. Активная разведка включает:

- обнаружение сетевых узлов;
- обнаружение открытых TCP- и UDP-портов на обнаруженных узлах путем сканирования портов;
- получение DNS-имен сетевых узлов;
- получение списка сетевых узлов организации, а также информации об их назначении с применением переноса зоны DNS;
- построение карты сети;
- идентификация ОС и ПО путем получения «отпечатков ОС».

На втором этапе применяются сканеры безопасности, позволяющие по известной им базе уязвимостей определить степень подверженности исследуемых узлов сетевым атакам.

Результаты обследования оформляются в виде таблиц 8.2 и 8.3.

ВЫПОЛНИТЬ!

1. Включите все сетевые узлы моделируемой сети. Убедитесь в корректности настройки IP-адресов и DNS-серверов. По необходимости произведите дополнительную настройку.

Примечание: Для задания сетевых параметров интерфейсов в ОС Linux можно воспользоваться командой:

```
ifconfig eth0 192.168.10.3 netmask 255.255.255.0 up
```

где 192.168.10.3 — IP-адрес, который будет установлен на сетевом интерфейсе, ключевое слово netmask используется для задания маски сети, ключевое слово up принуждает ОС активизировать интерфейс. Для задания адреса DNS-сервера необходимо отредактировать файл «/etc/resolv» (параметр search установить в alpha.local, nameserver в 192.168.10.1)

Таблица 8.2

Перечень сетевых узлов исследуемой сети

№	Наименование	Выводимые результаты			
1	Обследуемый сегмент сети	Сканируемый диапазон IP-адресов			
2	Характер обнаруженных узлов в сегменте	Рабочие станции, web-серверы, контроллеры домена и т.п.			
3	Возможность идентификации сетевых узлов	Результаты использования Ping-разведки. Результаты, полученные с использованием других ICMP сообщений. Результаты, полученные при использовании переноса зоны DNS.			
4	Выявленные узлы	IP	Назначение узла	Тип и версия ОС	Представленные сетевые сервисы и их версии, открытые порты
5	Карта сетевого сегмента и его подключения к другим сетям	Карта сети в графическом или табличном варианте			

2. На станцию сканирования Windows установите ПО AdRem Netcrunch (демонстрационную версию программы можно загрузить по адресу «http://www.adremsoft.com/netcrunch/»). Утилиты cgichk и nessuswx не имеют автоматического установщика, их необходимо скопировать в какую-либо папку на жёсткий диск станции сканирования Windows.

8.4. Обнаружение сетевых узлов

Стандартным способом обнаружения сетевых узлов при заданном диапазоне IP-адресов является применение утилиты Ping, которая входит в состав практически любой ОС. Такой способ обнаружения активных сетевых узлов

называется Ping-разведкой (Ping sweep). Утилита Ping использует протокол ICMP для проверки доступности сетевого узла: на исследуемый сетевой узел посылается ICMP-запрос (тип 8), в случае доступности сетевого узла будет получен ICMP-ответ (тип 0). Существует большое количество утилит, которые позволяют ускорить и автоматизировать этот процесс, например fping, hping3 и IP-Tools.

Сканирование подсети можно выполнять с помощью утилиты fping, используя команду:

```
fping -g 192.168.10.0 192.168.10.254
   2>/dev/null | grep 'is alive'
```

где ключ -g позволяет указать список тестируемых узлов. Использование `2>/dev/null` позволяет не выводить на экран служебные сообщения утилиты fping. Дополнительная обработка утилитой grep (поиск текста с использованием регулярных выражений Perl) позволяет вывести на консоль только доступные сетевые узлы, так как будут выведены только строки, содержащие подстроку «is alive»:

```
192.168.10.1 is alive
192.168.10.2 is alive
192.168.10.3 is alive
192.168.10.4 is alive
192.168.10.5 is alive
```

В приведенном примере в результате выполнения команды fping, запущенной со станции сканирования Linux, получены сведения о том, что в заданном диапазоне адресов тестировались все возможные адреса узлов сети 192.168.10.0/24, и среди них только узлы с адресами 192.168.10.1,…, 192.168.10.5 присутствуют в сети и отвечают на запросы.

Метод Ping-разведки является наиболее универсальным методом обнаружения сетевых узлов, однако по ряду причин данный метод может оказаться неэффективным. Например, в случае, когда обследуемая сеть защищена межсетевыми экранами, блокирующими ICMP-сообщения, или если пользователи защищены персональными межсетевыми экранами.

Кроме Ping-разведки существуют другие методы получения списка сетевых узлов. Рассмотрим некоторые из них более подробно.

Использование широковещательной посылки ICMP-запроса. Данный метод заключается в использовании широковещательных адресов назначения при осуществлении Ping-разведки. Протокол ICMP предполагает, что при получении пакета, содержащего ICMP-запрос с широковещательным адресом назначения, требуется ответ. Однако, в соответствии с RFC 1122, на приходящий ICMP-пакет с широковещательным адресом назначения можно не реагировать, что и делают многие современные ОС в целях безопасности.

Для использования этого метода можно воспользоваться стандартной утилитой ping (в ОС Linux необходимо дополнительно использовать ключ -b) Пример широковещательной посылки ICMP-запроса:

```
[bvv@srv181 bvv]$ ping -b 10.110.18.255
WARNING: pinging broadcast address
PING 10.110.18.255 (10.110.18.255) 56(84) bytes of data.
64 bytes from 10.110.18.2: icmp_seq=0 ttl=64 time=0.025 ms
64 bytes from 10.110.18.10: icmp_seq=0 ttl=255 time=0.355 ms (DUP!)
64 bytes from 10.110.18.11: icmp_seq=0 ttl=255 time=0.440 ms (DUP!)
64 bytes from 10.110.18.1: icmp_seq=0 ttl=255 time=0.904 ms (DUP!)
64 bytes from 10.110.18.9: icmp_seq=0 ttl=64 time=0.960 ms (DUP!)
64 bytes from 10.110.18.239: icmp_seq=0 ttl=255 time=0.991 ms (DUP!)
64 bytes from 10.110.18.3: icmp_seq=0 ttl=255 time=1.33 ms (DUP!)
--- 10.110.18.255 ping statistics ---
1 packets transmitted, 1 received, +6 duplicates, 0% packet loss,
time 0ms
rtt min/avg/max/mdev = 0.025/0.716/1.337/0.420 ms, pipe 2
```

В приведенном примере показано, что при сканировании сети 10.110.18.0/24 широковещательными ICMP-сообщениями ответы были получены от семи узлов. Следовательно, эти узлы присутствуют в сети. Кроме того, можно сделать вывод, что указанные узлы используют ОС, отличную от ОС семейства Windows, так как эти ОС не отвечают на широковещательные ICMP-запросы.

Применение ICMP-пакетов, отличных от ECHO-запросов. Данный метод заключается также в использовании протокола ICMP, но применяются не ECHO-запросы, а иные типы пакетов (например, тип 13 — TIMESTAMP или 17 — ADDRESS MASK REQUEST). ICMP-пакет TIMESTAMP позволяет запрашивать метку системного времени удаленной системы. Запрос и ответ ADDRESS MASK предназначен для получения сетевой маски бездисковыми системами (тонкими клиентами) в процессе загрузки. Данный тип запроса может быть использован для получения сетевой маски конкретного устройства. Метод обнаружения сетевых узлов может быть реализован, например, с помощью утилиты icmpush. Запуск утилиты icmpush с ключом -tstamp позволяет послать на выбранный сетевой узел ICMP-сообщение TIMESTAMP.

```
v-3:/home/bvv# icmpush -tstamp server.alpha.local
server.alpha.local -> 21:58:52
```

Из приведенного примера видно, что сетевой узел «server.alpha.local» доступен, кроме того, можно определить время, которое показывают системные часы на удаленной системе. При недоступности сетевого узла или при отключенной на удаленном узле функции ответа на запросы TIMESTAMP утилита icmpush ответ не возвращает.

Использование многоадресных рассылок IP-пакетов (IP Multicast). Многоадресные рассылки представляют собой технологию доставки трафика нескольким потребителям с экономией полосы пропускания. Существует ряд

множественных адресов, на которые по умолчанию должны отвечать сетевые узлы, которые поддерживают многоадресные рассылки. Среди них адрес 224.0.0.1 — все системы в текущей подсети, 224.0.0.2 — все маршрутизаторы в текущей подсети. Другие интересные множественные адреса можно найти по адресу «http://www.iana.org/assignments/multicast-addresses».
Метод можно применить с помощью команды Ping.

```
[bvv@srv181 bvv]$ ping 224.0.0.1
PING 224.0.0.1 (224.0.0.1) 56(84) bytes of data.
64 bytes from 10.101.18.2: icmp_seq=0 ttl=64 time=0.031 ms
64 bytes from 10.101.18.241: icmp_seq=0 ttl=64 time=0.113 ms (DUP!)
64 bytes from 10.101.18.10: icmp_seq=0 ttl=255 time=0.357 ms (DUP!)
64 bytes from 10.101.18.9: icmp_seq=0 ttl=64 time=1.03 ms (DUP!)
--- 224.0.0.1 ping statistics ---
1 packets transmitted, 1 received, +3 duplicates, 0% packet loss,
time 0ms
rtt min/avg/max/mdev = 0.031/0.383/1.033/0.394 ms, pipe 2
```

В приведенном примере показано, что при сканировании сети многоадресными ICMP-сообщениями ответы были получены от четырех узлов, присутствующих в сети и поддерживающих многоадресные рассылки.

При отсутствии многоадресных адресов в системной таблице маршрутизации можно воспользоваться ключом -i утилиты Ping для указания интерфейса, с которого будет происходить отправка пакетов.

ARP-разведка. Для обнаружения сетевых узлов, находящихся в одном сетевом сегменте (в одном широковещательном домене), можно использовать протокол ARP. Метод заключается в посылке ARP-запроса на получение MAC-адреса узла по известному IP-адресу. В случае доступности сетевого узла будет получен ответ.

Рассмотрим реализацию данного метода на примере утилиты arping. Используя команду `arping –i eth1 –c 1 192.168.10.1`, можно проверить доступность сетевого узла с IP-адресом 192.168.10.1, ключ -i позволяет указать сетевой интерфейс для выполнения ARP-запроса, ключ -c определяет количество посылаемых запросов.

```
v-3:/home/bvv# arping –i eth1 –c 1 192.168.10.1
ARPING 192.168.10.1
60 bytes from 00:40:05:05:a4:0b (192.168.10.1): index=0
time=66.996 usec
--- 192.168.10.1 statistics ---
1 packets transmitted, 1 packets received,   0% unanswered
```

Из результата работы приведенной команды можно сделать вывод, что узел с IP-адресом 192.168.10.1 доступен и MAC-адрес соответствующего интерфейса имеет вид 00:40:05:05:a4:0b.

С использованием утилиты tethereal, которая предназначена для перехвата сетевого трафика (здесь термин перехват сетевого трафика понимается как отображение или копирование трафика, проходящего через сетевой ин-

терфейс), можно отследить соответствующую сетевую активность. В примере используется ключ -i для указания сетевого интерфейса, с которого будет происходить перехват трафика. Ключевое слово arp указывает на то, что будут сниматься только данные протокола ARP.

```
v-3:/home/bvv# tethereal -ieth1 arp 2>/dev/null
  0.000000     192.168.10.3-> Broadcast    ARP Who has
192.168.10.1?  Tell 10.0.0.4
  0.000249 server.alpha.local -> 192.168.10.3    ARP 192.168.10.1
is at 00:40:05:05:a4:0b
```

Из приведенного примера видно, что с интерфейса с IP-адресом 192.168.10.3 было послано широковещательное ARP-сообщение с целью определения MAC-адреса узла 192.168.10.1 и был получен ответ от узла server.alpha.local.

Данный метод можно использовать не только при нахождении в одном сегменте с обследуемыми сетевыми узлами, но и при включенной на граничном маршрутизаторе сетевого сегмента функции proxy-arp.

TCP- и UDP-разведка представляют собой методы, при которых доступность сетевого узла определяется на основании доступности соответствующих TCP- либо UDP-портов. Заметим, что данный метод отличается от сканирования портов, так как достаточно получить любой ответ от любой службы обследуемого сетевого узла.

Для реализации этого метода обнаружения можно воспользоваться утилитой hping3. Указанная утилита представляет собой многофункциональный сетевой отладчик, и, в частности, может использоваться для реализации описанного метода. Для запуска утилиты воспользуемся командой `hping3 -p 81 192.168.10.1 -c 3`, ключ -p указывает на номер порта, на который будет послан запрос (по умолчанию используется TCP-порт), ключ -c определяет количество отправляемых пакетов.

```
# hping3 -p 81 192.168.10.1 -c 3
HPING 192.168.10.1 (eth0 192.168.10.1): NO FLAGS are set, 40
headers + 0 data bytes
len=46 ip=192.168.10.1 ttl=128 id=17590 sport=81 flags=RA seq=0
win=0 rtt=0.8 ms
len=46 ip=192.168.10.1 ttl=128 id=17591 sport=81 flags=RA seq=1
win=0 rtt=0.5 ms
len=46 ip=192.168.10.1 ttl=128 id=17592 sport=81 flags=RA seq=2
win=0 rtt=0.6 ms

--- 192.168.10.1 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.6/0.8 ms
```

В приведенном примере в результате выполнения команды hping3 путем посылки TCP-пакета на порт 81 (без установленных флагов) и анализа ответного пакета (установлены флаги RST и ACK) получены сведения о том, что

сетевой узел 192.168.10.1 присутствует в сети. После завершения работы утилита hping3 выводит статистику — количество отосланных и полученных пакетов, а также минимальное, среднее и максимальное время доставки пакетов.

С помощью этой же утилиты можно реализовать описанный метод с использованием протокола UDP. Для этого можно использовать команду `hping3 -2 –n -p 81 192.168.10.1 -c 3`, ключ -2 указывает на использование протокола UDP, -n отключает разрешение DNS-имени.

```
# hping3 -2 -n -p 81 192.168.10.1 -c 3
HPING 192.168.10.1 (eth0 192.168.10.1): udp mode set, 28 headers
+ 0 data bytes
ICMP Port Unreachable from ip=192.168.10.1
ICMP Port Unreachable from ip=192.168.10.1
ICMP Port Unreachable from ip=192.168.10.1

--- 192.168.10.1 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

В приведенном примере в результате выполнения команды hping3 получены сведения о том, что сетевой узел 192.168.10.1 присутствует в сети. Данная информация была получена путем отсылки UDP-пакета на 81 порт и анализа ответного пакета (было получено ICMP-сообщение «порт не доступен»). Обратите внимание, что если соответствующий UDP-порт будет открыт, то метод функционировать не будет.

Аудитор может получить информацию об имеющихся в сети устройствах из ARP-кэшей серверов и активного сетевого оборудования. Получение этой информации возможно, если аудитор находится в рамках модели «белого ящика». Обычно для этого необходимо выполнить команды `arp –a` на сервере и `show arp` в консоли активного сетевого устройства (на примере оборудования Cisco).

CDP-разведка. Данный метод позволяет обнаружить и получить информацию об устройствах, работающих по протоколу CDP (Cisco Discovery Protocol — протокол обнаружения Cisco) и находящихся в одном сетевом сегменте. Информацию о CDP-узлах можно получать, непосредственно подключаясь к консоли активного сетевого устройства производства фирмы Cisco Systems либо перехватывая CDP-объявления (например, с использованием утилиты cdpr от MonkeyMental.com).

Прослушивание сети. Данный метод заключается в перехвате сетевых пакетов из некоторого сетевого сегмента и в последующем анализе используемых IP-адресов. Для реализации данного метода можно использовать утилиту Ethereal.

ВЫПОЛНИТЬ!

3. Выявите сетевые узлы в локальном сетевом сегменте с использованием:

- утилиты fping;
- утилиты ping и широковещательной ICMP-посылки;
- утилиты icmpush (тип ICMP-пакетов13 и 17);
- утилиты ping и многоадресной рассылки;
- утилиты arping;
- утилиты hping3 и методов TCP- и UDP-разведки;
- утилиты arp и метода ARP-кэша;
- утилиты Ethereal и метода прослушивания сети. Напишите сценарий, выводящий список IP-адресов сетевых узлов, извлеченных из перехваченных с сетевого интерфейса пакетов (рекомендуется воспользоваться консольным вариантом утилиты Ethereal – tethereal).

4. По результатам сделайте вывод о возможности идентификации сетевых узлов в исследуемой сети различными методами, заполните строку 3 таблицы 8.2.

8.5. Сканирование портов и идентификация ОС

Для сканирования портов и для идентификации версий ОС и сервисов можно использовать утилиту nmap, которая реализует большое количество методов и техник сканирования портов и идентификации ресурсов. Утилита nmap позволяет формировать результаты сканирования в формате XML для просмотра web-обозревателем с использованием XSL-преобразования.

Для решения задачи обнаружения открытых TCP- и UDP-портов, а также ОС, сервисов и их версий на удаленном сетевом узле 192.168.10.1 можно воспользоваться командой `nmap –sS –sU –sV –O --osscan-guess 192.168.10.1`, выполняемой на станции сканирования Linux. Ключ -sS указывает на сканирование TCP-портов, ключ -sU — на сканирование UDP-портов, ключ -sV указывает на идентификацию сетевых сервисов, ключ -O — на идентификацию ОС, использование ключа -ossccan-guess предполагает «агрессивную» идентификацию ОС.

По умолчанию применяется техника случайного сканирования, то есть тестирование портов происходит в случайном порядке.

```
# nmap -sS -sU -sV -O --osscan-guess 192.168.10.1
Starting Nmap 4.20RC1 ( http://insecure.org ) at 2006-12-01 22:16 YEKT
Interesting ports on 192.168.10.1:
Not shown: 3144 closed ports
PORT     STATE         SERVICE       VERSION
21/tcp   open          tcpwrapped
25/tcp   open          smtp          Microsoft ESMTP
                                     6.0.3790.1830
53/tcp   open          domain        Microsoft DNS
80/tcp   open          http          Microsoft IIS webserver 6.0
88/tcp   open          kerberos-sec  Microsoft Windows kerberos-sec
110/tcp  open          pop3          Microsoft Windows 2003 POP3 Service 1.0
135/tcp  open          msrpc         Microsoft Windows RPC
139/tcp  open          netbios-ssn
389/tcp  open          ldap          Microsoft LDAP server
443/tcp  open          ssl/http      Microsoft IIS webserver 6.0
```

```
445/tcp  open          microsoft-ds  Microsoft Windows 2003 microsoft-ds
464/tcp  open          kpasswd5?
593/tcp  open          ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open          ssl/ldap      Microsoft LDAP server
1026/tcp open          msrpc         Microsoft Windows RPC
1027/tcp open          ncacn_http    Microsoft Windows RPC over HTTP 1.0
1248/tcp open          msrpc         Microsoft Windows RPC
1723/tcp open          pptp?
3268/tcp open          ldap          Microsoft LDAP server
3269/tcp open          ssl/ldap      Microsoft LDAP server
53/udp   open          domain?
67/udp   open|filtered dhcps
68/udp   open|filtered dhcpc
88/udp   open|filtered kerberos-sec
123/udp  open          ntp           NTP v3
137/udp  open          netbios-ns    Microsoft Windows NT netbios-ssn
                                     (workgroup:ALPHA)
138/udp  open|filtered netbios-dgm
389/udp  open|filtered ldap
445/udp  open|filtered microsoft-ds
464/udp  open|filtered kpasswd5
500/udp  open|filtered isakmp
1701/udp open|filtered L2TP
3456/udp open|filtered IISrpc-or-vat
4500/udp open|filtered sae-urn
1 service unrecognized despite returning data. If you know the ser-
vice/version, please submit the following fingerprint at
http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port53-UDP:V=4.20 … (часть вывода вырезана)
MAC Address: 00:0C:29:37:7B:86 (VMware)
Device type: general purpose
Running (JUST GUESSING) : Microsoft Windows 2003|XP|2000 (94%)
Aggressive OS guesses: Microsoft Windows 2003 Server SP1 (94%), Microsoft
Windows XP SP2 (92%), Microsoft Windows XP SP2 (firewall disabled) (91%),
Microsoft Windows 2000 Server SP4 (90%), Microsoft Windows 2000 SP3 (90%),
Microsoft Windows 2000, SP0, SP1, or SP2 (89%), Microsoft Windows 2000 SP4
(88%), Microsoft Windows Server 2003 Enterprise Edition 64-Bit SP1 (87%)
No exact OS matches for host (If you know what OS is running on it, see
http://insecure.org/nmap/submit/).
TCP/IP fingerprint:
OS:SCAN(V=4.20RC1%D=12/1%OT=21%CT=1%CU=1%PV=Y%DS=1%G=Y%M=000C29%...%DLI=S)
(часть вывода удалена)
Network Distance: 1 hop
Service Info: Host: server.alpha.local; OSs: Windows, Windows 2000
OS and Service detection performed. Please report any incorrect results at
http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 151.541 seconds
```

Из приведенных результатов работы утилиты nmap можно сделать вывод, что на узле 192.168.10.1 открыты TCP-порты 21, 25, 53, 80, 88, 110 и т.д., причём были идентифицированы следующие сервисы, использующие эти порты:

- порт 25 — почтовый сервер ESMTP, производитель Microsoft;
- порт 53 — DNS-сервер, производитель Microsoft;
- порт 80 — web-сервер IIS, версия 6.0, производитель Microsoft;
- порт 88 — сервер Kerberos для Microsoft Windows, производитель Microsoft;

– порт 110 — почтовый сервер POP3 для Microsoft Windows 2003, производитель Microsoft
– …

UDP-порты 53, 67, 68 и др. открыты, причем если указано состояние «open», то порт явно открыт. Если состояние «open|filtered», то порт может быть открыт, а может быть отфильтрован межсетевым экраном. Различить эти состояния не представляется возможным вследствие особенностей реализованного метода UDP-сканирования (при тестировании порта не было получено ICMP-сообщение «порт недоступен»).

Была идентифицирована большая часть сервисов, использующих перечисленные UDP-порты. Однако идентификация сервиса, функционирующего на UDP-порту 53, не прошла корректно вследствие большого количества запросов к серверу. Тем не менее, если провести повторное сканирование с помощью команды `nmap -sU -p U:53 -sV 192.168.10.1`, где опция -p U:53 указывает, что обследоваться должен только один UDP-порт 53, то можно идентифицировать версию сервиса.

```
# nmap -sU -p U:53 -sV 192.168.10.1
…
Interesting ports on 192.168.10.1:
PORT   STATE SERVICE VERSION
53/udp open  domain  Microsoft DNS
…
```

После перечня открытых портов nmap возвращает следующую информацию:

– MAC-адрес производителя сетевого адаптера (в соответствии с префиксом MAC-адреса). Обратите внимание, что если бы станция сканирования Linux и сетевой узел 192.168.10.3 находились в различных IP-сетях (были связаны через маршрутизатор), то этот адрес был бы неизвестен, так как для его определения используется протокол ARP:

```
MAC Address: 00:0C:29:37:7B:86 (VMware)
```

– выявленный тип обследуемого устройства, в данном случае указано, что это устройство общего назначения (другие возможные значения router – маршрутизатор, switch – коммутатор, game console – игровая консоль и др.):

```
Device type: general purpose
```

Утилита nmap в примере не смогла точно идентифицировать тип и версию ОС. Тем не менее указано, что вероятней всего (с уровнем доверия 94 %) это версия Microsoft Windows 2003, XP либо 2000. Проанализировав тип почтового сервиса POP3, можно сделать вывод, что на обследуемой системе запущена ОС Windows Server 2003. В выводе также указываются степени доверия относительно других ОС, полученные путём «агрессивной» идентификации ОС.

Далее в выводе указывается «сетевое расстояние» (network distance), то есть количество промежуточных IP-сетей. В приведённом примере этот параметр равен единице, так как станция сканирования Linux и сетевой узел 192.168.10.1 находятся в одной IP-подсети:

```
Network Distance: 1 hop
Service Info: Host: server.alpha.local;
OSs: Windows, Windows 2000
```

Параметр «Service Info» содержит краткую информацию об имени узла, а также предполагаемый тип и версию ОС.

В конце вывода команды приводится информация о количестве просканированных IP-адресов и о суммарном времени сканирования.

С использованием ключа -oX <имя xml-файла> результаты сканирования можно сохранить в формате XML для дальнейшего удобного изучения с помощью web-обозревателя или для дальнейшей автоматической обработки.

ВЫПОЛНИТЬ!

5. С помощью утилиты nmap проведите сканирование портов сетевых узлов, найденных на предыдущем шаге. Сформируйте списки открытых TCP- и UDP-портов, идентифицируйте версии ОС и запущенных сервисов. По результатам сделайте вывод о возможности обнаружения открытых портов и идентификации типа и версии ОС, а также сетевых сервисов на сетевых узлах исследуемой сети, заполните строку 4 табл. 8.2.

8.6. Использование DNS для обнаружения и выяснения назначения сетевых узлов

Использование DNS-серверов позволяет аудитору (в рамках модели «черного ящика») получить информацию о назначении сетевых узлов исследуемых сегментов ЛВС и внешних сетевых узлов организации. Например, наличие в базе DNS записи типа MX для узла «smtp.example.ru» дает аудитору информацию о том, что указанный узел является почтовым сервером. Обратный DNS (записи типа PTR) позволяет получить одно из имен сетевого узла по IP-адресу.

Для взаимодействия с DNS-сервером могут быть использованы стандартные системные утилиты nslookup (ОС Windows и UNIX), host или dig (ОС UNIX).

Непрерывный набор пространства имен DNS называется зоной DNS. Для получения зоны DNS можно использовать протокол переноса зоны DNS (DNS zone transfer известен так же, как AXFR).

Рассмотрим пример переноса зоны DNS с использованием утилиты dig. Для этого необходимо указать тип запроса – axfr и параметр «@имя_сервера», указывающий на сервер, с которого будет произведен перенос зоны DNS.

```
$ dig axfr examp1.si.ru @ns.examp1.si.ru
; <<>> DiG 9.2.4 <<>> axfr examp1.si.ru @ns.examp1.si.ru
;; global options:  printcmd
examp1.si.ru.               86400   IN      SOA     examp1.si.ru.
n.examp1.si.ru. 2006102601 86400 7200

2419200 604800
examp1.si.ru.               86400   IN      NS      ex1.pp.ru.
examp1.si.ru.               86400   IN      NS      ns.secondary.net.ua.
examp1.si.ru.               86400   IN      A       145.23.211.2
examp1.si.ru.               86400   IN      MX      10 ex.miti.ru.
icq.examp1.si.ru.           86400   IN      CNAME   examp1.si.ru.
deltaplan.examp1.si.ru.     86400   IN      A       232.44.22.77
deltaplan.examp1.si.ru.     86400   IN      MX      10 mail1.delta-plan.ru.
webcam.examp1.si.ru.        86400   IN      A       195.64.22.33
home.examp1.si.ru.          86400   IN      A       83.44.44.44
ex1.examp1.si.ru.           86400   IN      A       34.12.34.22
conference.examp1.si.ru.    86400   IN      CNAME   examp1.si.ru.
squish.examp1.si.ru.        86400   IN      MX      10 squish.examp1.si.ru.
squish.examp1.si.ru.        86400   IN      A       230.252.141.7
vjud.examp1.si.ru.          86400   IN      CNAME   examp1.si.ru.
examp1.si.ru.               86400   IN      SOA     examp1.si.ru.
n.examp1.si.ru. 2006102601 86400 7200

2419200 604800
;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(localhost)
;; WHEN: Thu Nov  2 13:01:11 2006
;; XFR size: 19 records
```

В приведенном примере с помощью утилиты dig был произведен перенос зоны DNS с DNS-сервера «ns.examp1.si.ru».

Вывод команды представляется в табличной форме, которая повторяет структуру базы DNS. Первая строка таблицы является записью типа SOA (Start of Authority) которая, в частности, идентифицирует имя домена или зоны DNS: «examp1.si.ru.n.examp1.si.ru». Все имена, которые не заканчиваются точкой, дополняются именем домена для зоны DNS. Например, если бы в примере встретилось имя «example.ru» (без заключительной точки), то оно бы трансформировалось в имя «example.ru.example1.si.ru». Затем указан серийный номер, интервал времени, через который происходит обновление первичного DNS-сервера, и интервал, через который необходимо проводить обновление, если первая попытка обновления была неудачной.

Две последующие строки указывают на имена DNS-серверов «ex1.pp.ru» и «ns.secondary.net.ua», которые являются авторитетными для данного домена (то есть такими DNS-серверами, которые содержат информацию об этой зоне).

Следующая запись типа A отображает имя сетевого узла examp1.si.ru на его IP-адрес 145.23.211.2.

Запись типа MX идентифицирует почтовый сервер. Основываясь на проведенном листинге можно сделать вывод, что для домена «examp1.si.ru» почтовым сервером является «ex.miti.ru» с числом предпочтения 10 (используемым для выбора того либо иного сервера при маршрутизации почты). Дру-

гими словами, если адрес получателя письма будет иметь вид «user@exam1.si.ru», то оно будет передано на почтовый сервер «ex.miti.ru».

Запись типа CNAME отображает псевдоним сетевого узла на его каноническое имя. Основываясь на листинге, можно сделать вывод, что «icq.examp1.si.ru» является псевдонимом для «examp1.si.ru» и, следовательно, соответствует серверу с IP-адресом 145.23.211.2.

Последней записью в таблице является запись типа SOA, что соответствует строгому определению зоны DNS: зона DNS — это серия записей DNS, начинающаяся с записи типа SOA для запрашиваемого имени, продолжающаяся любым количеством записей, отличных от SOA, заканчивающаяся повторным вхождением записи SOA.

Приведенная команда возвращает также время выполнения запроса, адрес DNS-сервера, к которому был произведен запрос, время, когда был произведен запрос, и количество записей, которые были возвращены. Ключевое слово «IN» в строках вывода указывает на то, что используется адресация и схема именования, применяемая в Интернет.

Одной из возможных записей в таблице DNS может быть запись типа SRV. Записи типа SRV предназначены для идентификации сетевых узлов, на которых присутствует заданный сервис. Эта запись позволяет пользователям, зная имя домена и имя нужного сервиса, получать имя узла и порт, на котором этот сервис запущен.

Например, наличие записи «_http._tcp.example.com. IN SRV 0 5 80 www.example.com.» указывает на то, что сервис HTTP использует порт 80 сервера «www.example.com». Элемент «_http» указывает на тип сервиса – HTTP. Элементы «_ftp» и «_ldap» — другие часто встречающиеся значения, указывающие соответственно на FTP- и LDAP-сервисы. Элемент «_tcp» указывает, что в качестве транспортного протокола для этого сервиса используется TCP. Значения 0, 5 и 80 указывают соответственно на приоритет, вес и номер порта, который используется сервисом.

Таким образом, осуществляя анализ информации, полученной путем переноса зоны DNS, можно получить подробную информацию о системном ландшафте (т.е. о составе и назначении серверов) организации.

В целях затруднения инвентаризации ресурсов системного ландшафта злоумышленником возможность переноса зоны DNS на недоверенные узлы должна быть запрещена.

ВЫПОЛНИТЬ!

6. Убедитесь, что настройки DNS-сервера, функционирующего на узле «Сервер», разрешают передачу зоны DNS. Для проверки и включения этого параметра выполните команду `dnsmgmt.msc /s` (можно через меню Пуск ⇒ Администрирование ⇒ DNS), в левой панели появившегося окна раскройте список «Forward Lookup Zones», из контекстного меню зоны alpha.local выберите пункт Свойства (Properties), перейдите в закладку

«Передача зоны» (Zone Transfers), убедитесь что пункт «Разрешить передачу зоны» (Allow zone transfers) включен. В случае, если параметр выключен, его необходимо включить (рис. 8.3).

7. С помощью утилиты dig, установленной на станции сканирования Linux, получите описание зоны DNS. Проанализируйте полученные данные. Сделайте выводы о том, какую информацию о сети можно получить с использованием передачи зоны DNS. Дополните строку 3 таблицы 8.2.

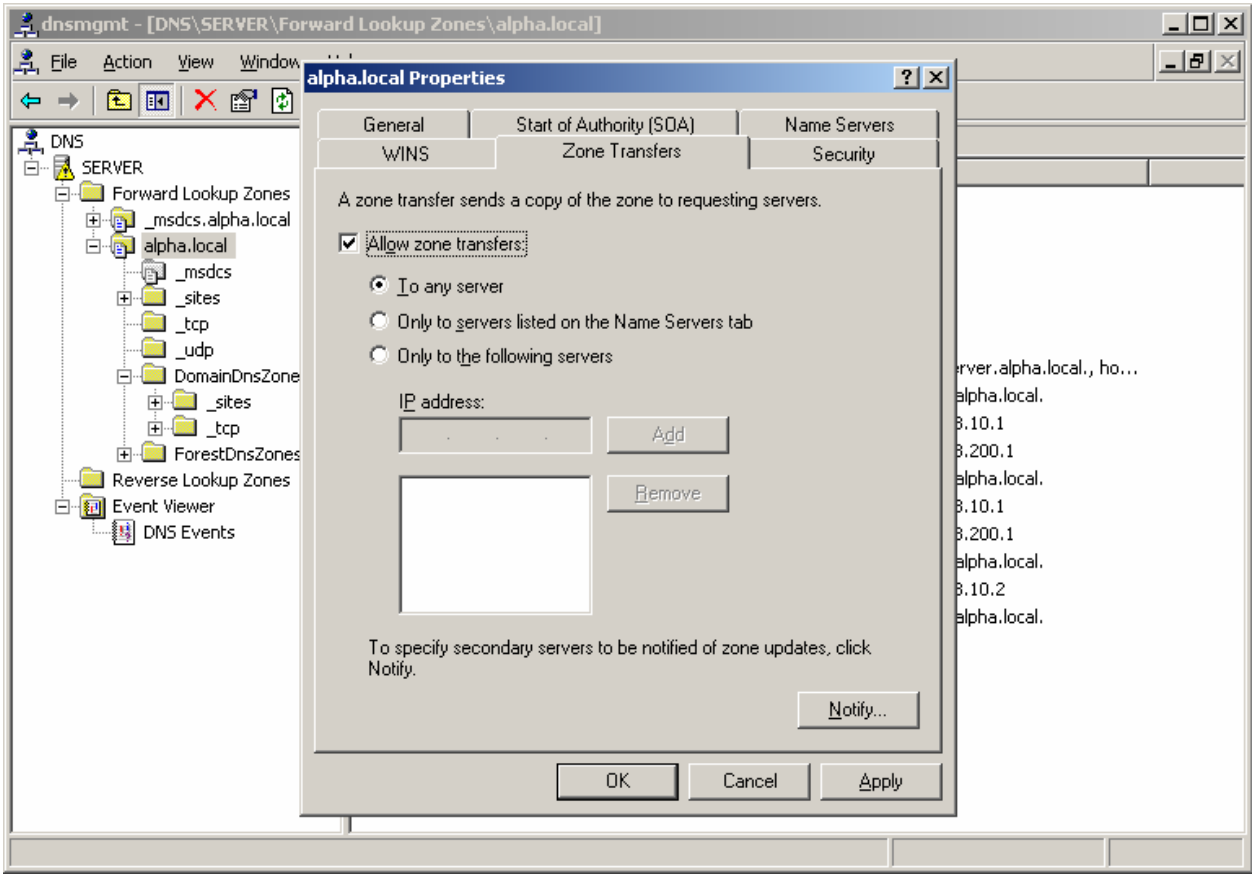


Рис. 8.3. Разрешение передачи зоны DNS

8.7. Создание карт сети

Карта сети позволяет получить представление об архитектуре и структуре сети. Обычно карта сети представляет изображение сетевой топологии в рамках физического (физическое расположение сетевых узлов и каналов, рис. 8.4), канального (способ коммутации сетевых узлов, рис. 8.5) и сетевого (схема взаимодействия между IP-подсетями, рис. 8.6) уровней модели OSI.

Существуют способы автоматического получения карты сети. Например, программы LAN MapShot и NetCrunch позволяют построить карты сети канального и сетевого уровней.

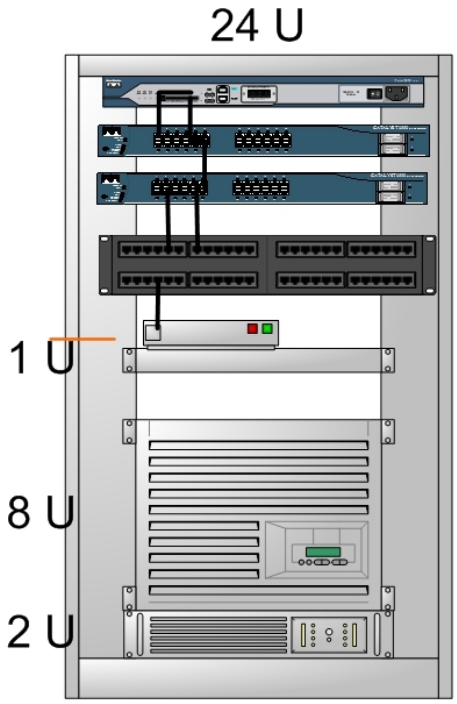


Рис. 8.4. Карта сети на физическом уровне

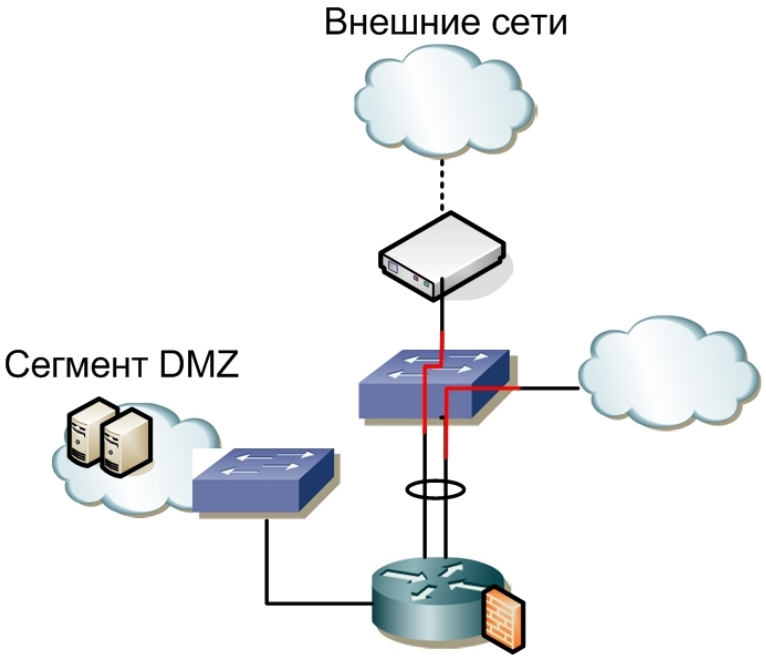


Рис. 8.5. Карта сети на канальном уровне

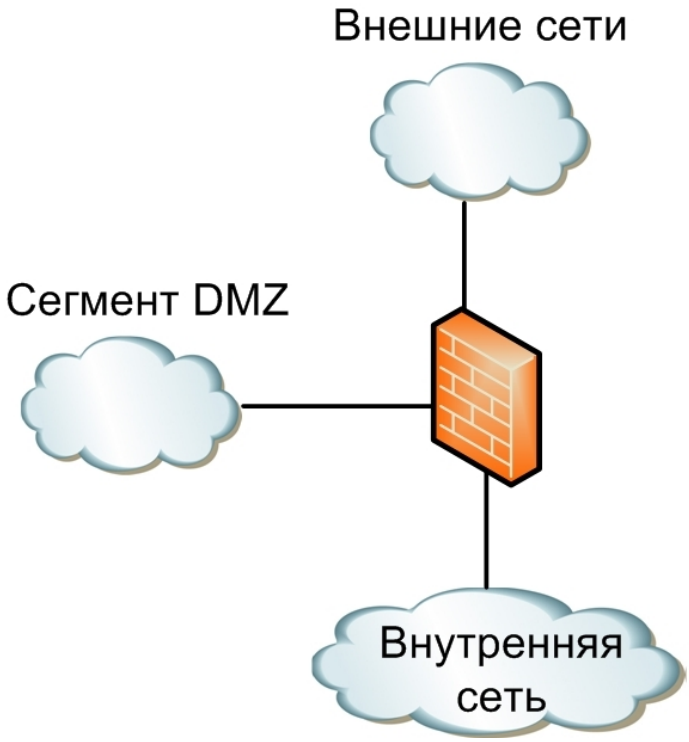


Рис. 8.6. Карта сети на сетевом уровне

Рассмотрим пример карты сети, построенной с использованием утилиты NetCrunch (рис. 8.7). Все найденные сетевые узлы соединены линией, обозначающей, что они находятся в одном сетевом сегменте.

Для построения карты сети на сетевом уровне можно воспользоваться утилитой traceroute (tracert в ОС Windows). Указанная утилита позволяет определить путь, пройденный пакетом от отправителя к получателю. Комбинируя эту информацию, можно создать карту сети. Утилиты NetCrunch и Visual Route используют указанный метод для автоматического построения карт сети на сетевом уровне.

ВЫПОЛНИТЬ!

8. С помощью демонстрационной версии программы NetCrunch, установленной на станции сканирования «Windows», постройте карту моделируемой сети. Для построения карты сети воспользуйтесь мастером построения карты «Create New Atlas», которого можно запустить при старте утилиты NetCrunch. Выберите «Yes» в окне подтверждения автоматического поиска устройств. В качестве адреса сети и маски укажите 192.168.10.0 и 255.255.255.0 соответственно. Выберите автоматический способ обнаружения сетевых узлов. Нажмите кнопку «next», выберите пункт «Все сетевые узлы» (All nodes). После обнаружения сетевых узлов карту сети можно просмотреть, выбрав адрес сети в пункте ***IP Networks*** $\Rightarrow$ ***Local*** левой панели. С помощью команды ***File*** $\Rightarrow$ ***Export*** $\Rightarrow$ ***Export Map*** карту сети можно сохранить в графическом файле.

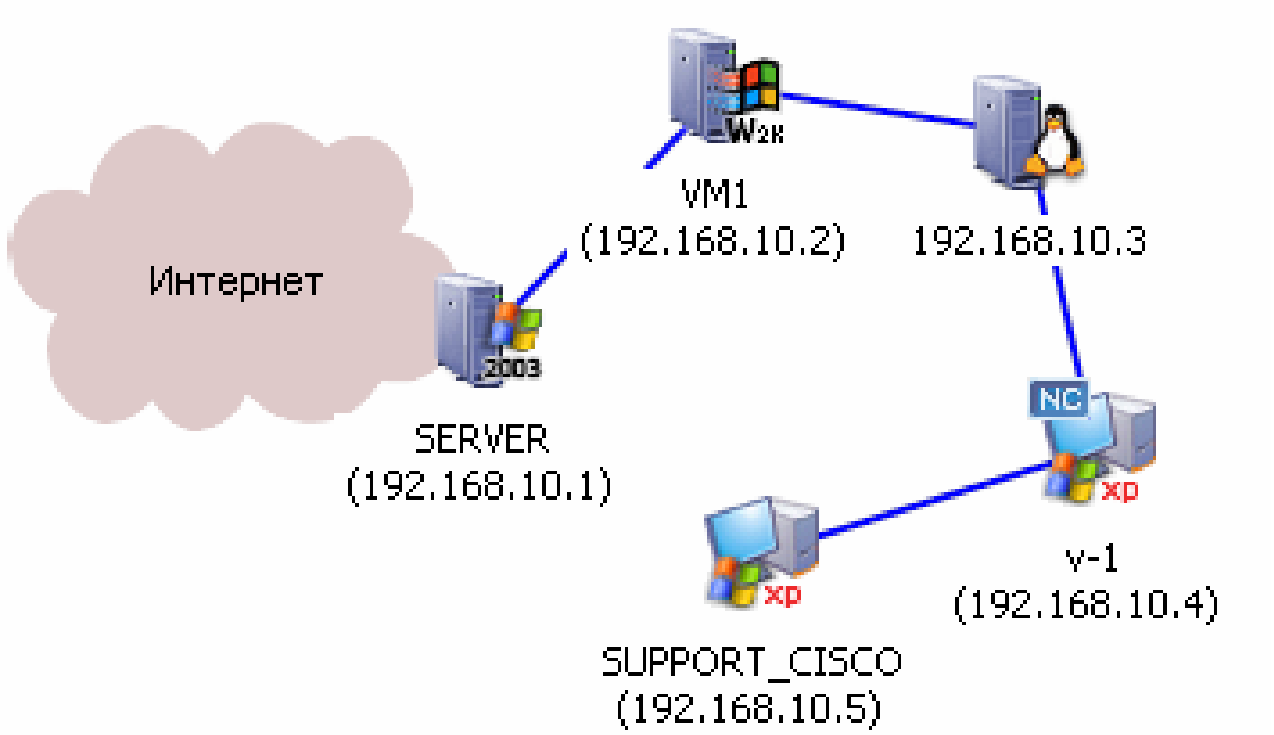


Рис. 8.7. Пример карты, построенной с помощью утилиты NetCrunch

8.8. Использование сканера безопасности Nessus

Сканер безопасности — это программное или программно-аппаратное средство, предназначенное для автоматизации процедуры выявления уязвимостей компьютерных систем. Его главной функцией является выяснение версий установленного программного обеспечения и ошибок конфигурации, в том числе в парольной политике. Для этого сканер безопасности обнаруживает доступные на узле сетевые службы, пытается подключиться к ним, а после этого — произвести соответствующий набор тестов.

Алгоритм работы сканера безопасности заключается в следующем: оператор задает некоторый набор IP-адресов или DNS-имен узлов, которые необходимо просканировать. После этого сканер производит проверку доступности данного узла, затем идентифицирует открытые порты и определяет запущенные сетевые сервисы.

Основным компонентом сканера безопасности является база уязвимостей. Используя ее, сканер пытается проверить уязвимости сетевых сервисов, поочередно применяя тесты, подходящие для данного выбранного сервиса. Сканеры безопасности могут проводить обнаружение уязвимостей не только в сетевых сервисах, но и в ОС, в локальных сервисах и приложениях. После завершения сканирования все собранные данные объединяются в отчеты различной формы. Аудитор может включать данные отчеты в документы, описывающие результаты инструментальной проверки.

При использовании сканеров безопасности аудитор должен соблюдать повышенную осторожность, так как при тестировании они могут реализовывать атаки на уязвимые системы, что может спровоцировать нарушение нормальной работоспособности системы.

Сканер безопасности не пытается «взломать» обследуемый узел, тем не менее производимые тесты могут быть опасными в том плане, что способны вызвать отказ в обслуживании. Кроме того, некоторые сканеры, такие как LANguard Network Security Scanner, позволяют выполнять атаку «удаленный подбор пароля» для доступа к общим файлам и папкам (в ОС семейства Windows NT это эквивалентно атаке на учетную запись пользователя).

В качестве иллюстрации рассмотрим широко известный и популярный сканер Nessus. Программная часть Nessus версии 3.0 является свободно распространяемой, но для пользователей, которые не приобрели лицензию, обновление баз данных уязвимостей производится только через неделю с момента их выпуска. Кроме того, свободно распространяемая версия может применяться лишь для сканирования узлов в подсетях класса C.

Структурно Nessus состоит из серверной части, клиентской части и набора подключаемых модулей (plug-ins). Серверная часть обеспечивает взаимодействие с сетевой средой, запуск выбранных тестов, а также получение и первичную обработку их результатов. Подключаемые модули — это сценарии тестов, написанные на специально разработанном для этого интерпретируемом языке NASL (Nessus Attack Scripting Language). Клиентская часть обеспечивает взаимодействие пользователя с сервером, выбор и настройку тестов, а также генерацию отчетов о сканировании. Обмен между клиентской и серверной частями ведется по прикладному протоколу NTP (Nessus Transport Protocol) и может быть как открытым (без шифрования передаваемого трафика), так и закрытым (с шифрованием по одному из протоколов SSL или TLS). По умолчанию сервер использует порт 1241.

Главной особенностью сканера безопасности Nessus является открытость сценариев тестирования и возможность написания пользователем своих собственных сценариев или доработки существующих. Этим Nessus кардинально отличается от подавляющего большинства коммерческих сканеров, программный код которых является на 100 % закрытым.

Рассмотрим применение сканера безопасности Nessus на примере версии 3.0.3 для ОС семейства Microsoft Windows. Установка данного сканера производится стандартным образом. Обязательным условием его нормальной работы является наличие *функционирующей* службы Tenable Nessus (рис. 8.8), которая устанавливается вместе со сканером.

Чтобы начать сканирование, необходимо нажать кнопку «Start Scan Task» в меню программы. Исходными данными для сканирования узла являются его IP-адрес (рис. 8.9), а также совокупность тестов, которые необходимо выполнить (рис. 8.10).

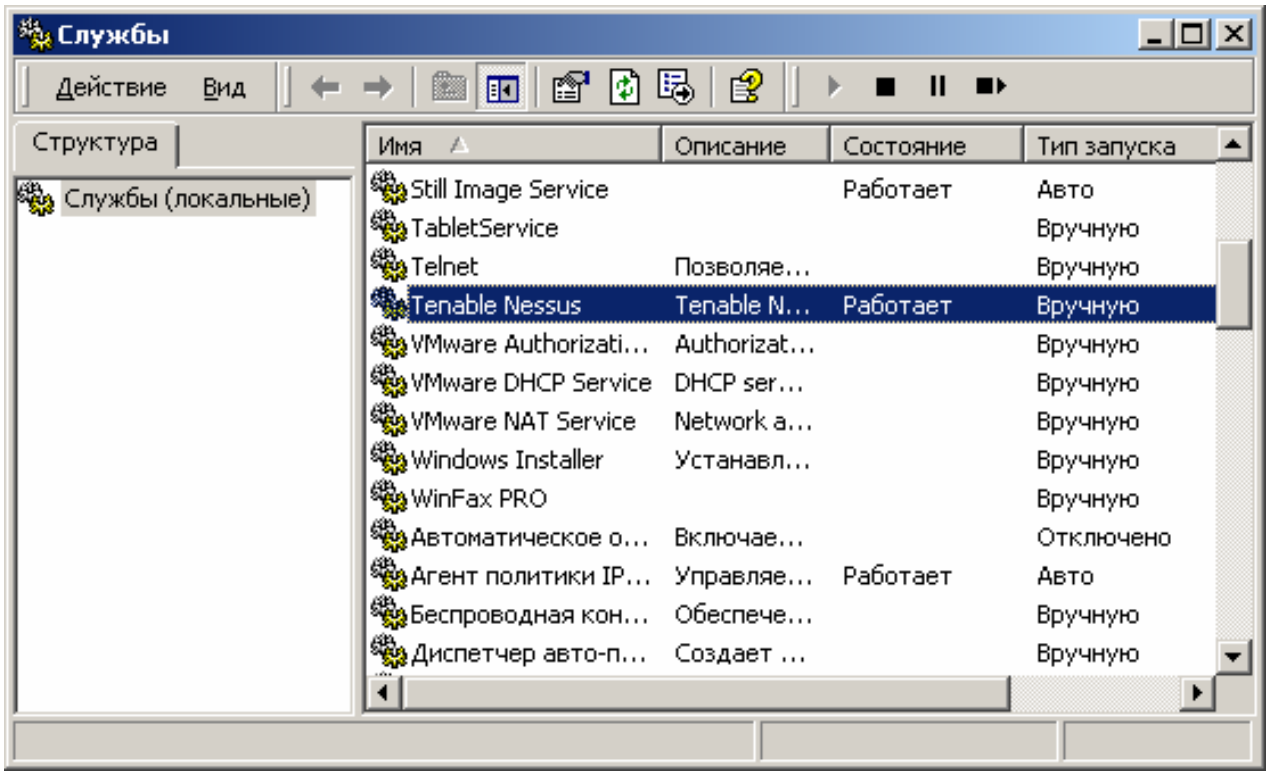


Рис. 8.8. Служба Tenable Nessus в перечне служб

Чтобы выбрать все тесты, за исключением «опасных» (тех, что могут вывести узел из работоспособного состояния), и использовать при этом настройки «по умолчанию», необходимо выбрать пункт «Enable all but dangerous plugins with default settings». При наличии уверенности в том, что временный выход узла из строя не нанесет никакого ущерба, можно выбрать пункт «Enable all plugins with default settings». При необходимости определить конкретный список проводимых тестов, а также потребности изменить настройки программы нужно выбирать пункт «Define my policy». Следует заметить, что в этом случае сделанные настройки и выбранный список тестов будут действовать лишь в ходе одной операции сканирования. Поэтому более разумным является предварительное создание собственной политики сканирования с использованием диалогового окна «Manage Policies» (рис. 8.11), которое открывается из меню программы. Под политикой понимается набор тестов и настроек программы.

Чтобы создать новую политику, необходимо нажать кнопку «Add a new policy», а затем ввести ее имя. Для изменения настроек необходимо нажать «Edit Settings», а для выбора списка тестов — «Edit Plugins». Диалоговое окно с настройками программы показано на рис. 8.12.

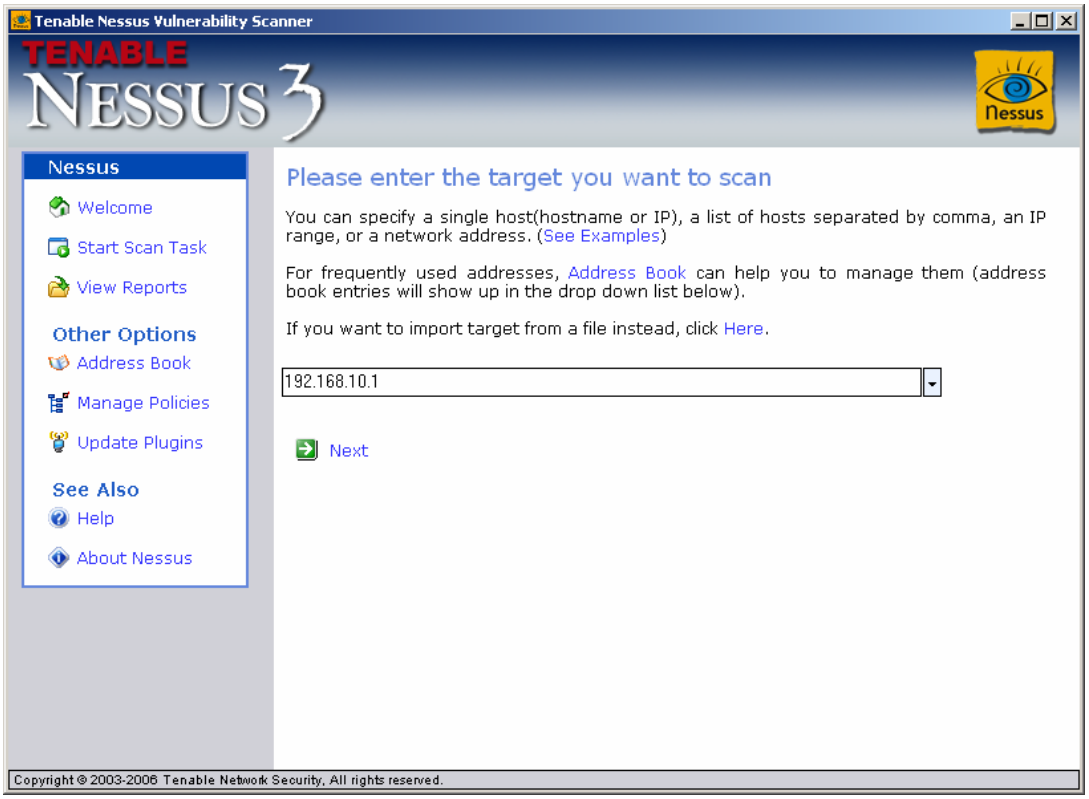


Рис. 8.9. Указание IP-адреса сканируемого узла

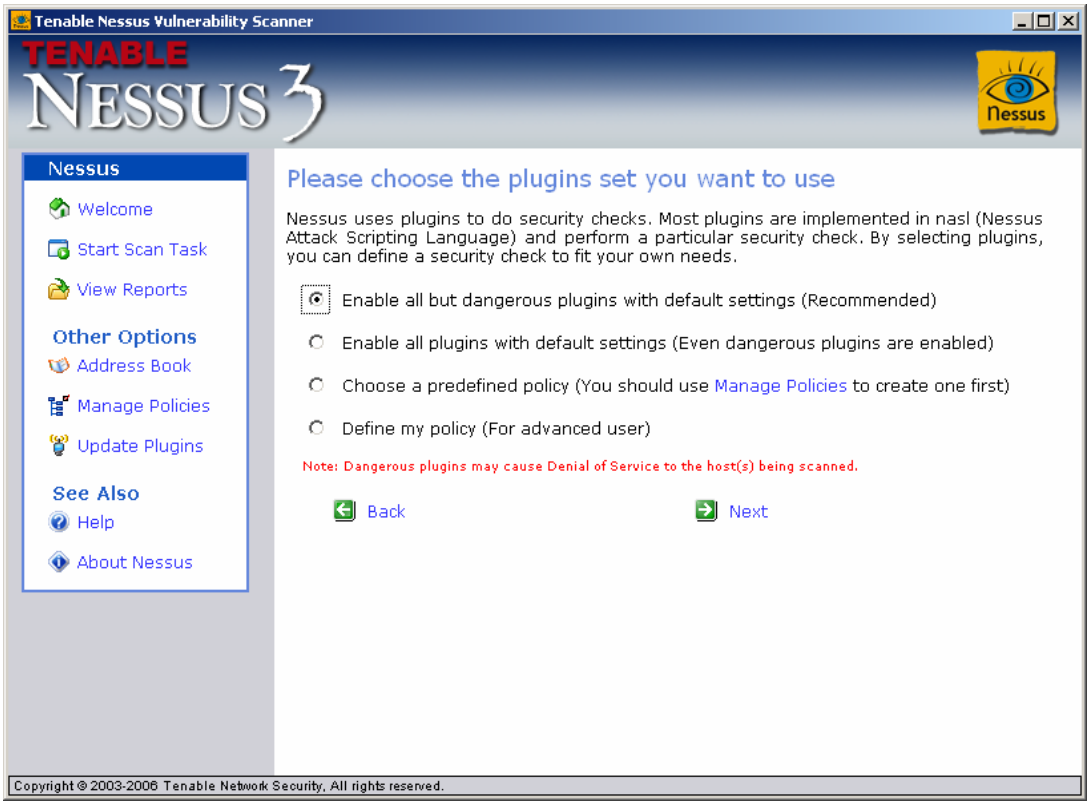


Рис. 8.10. Отключение опасных тестов

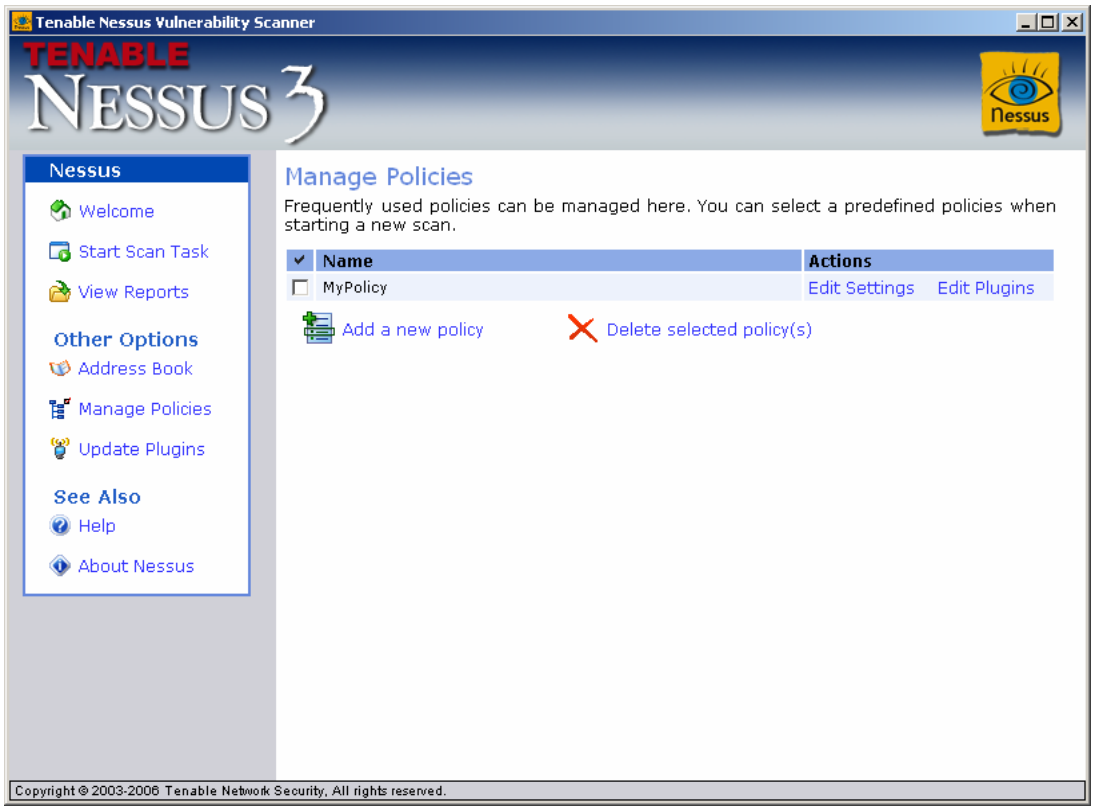


Рис. 8.11. Создание политики сканирования

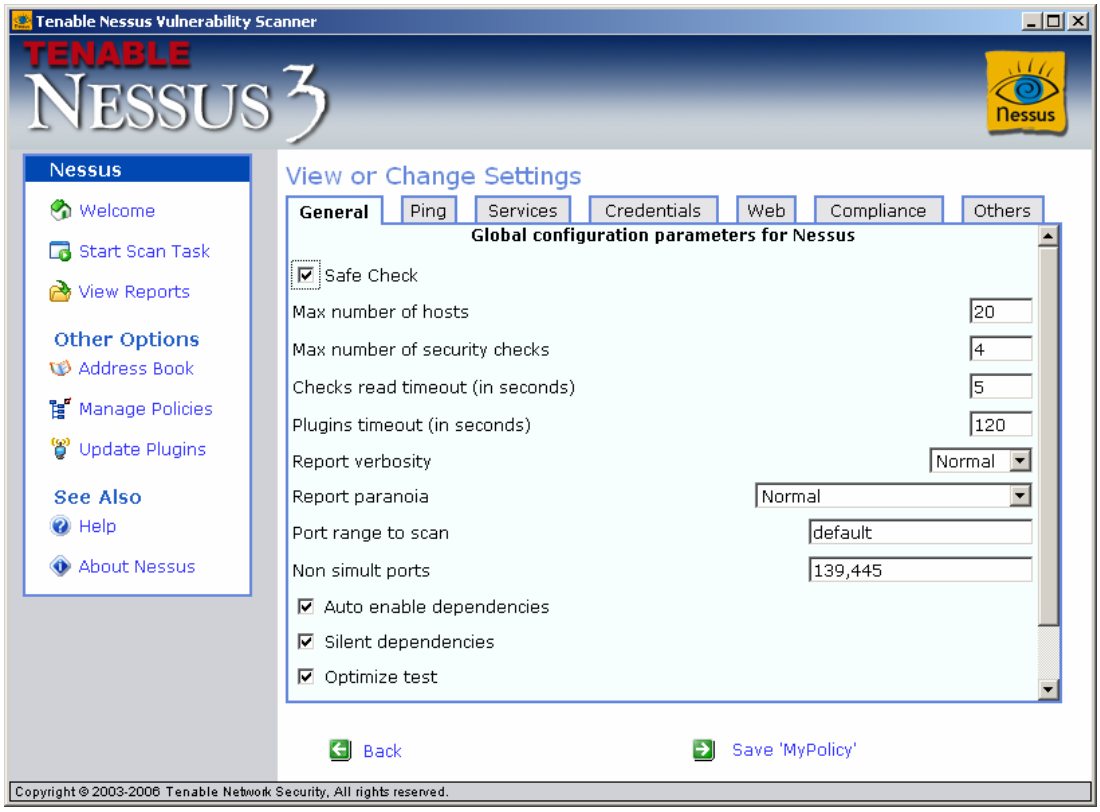


Рис. 8.12. Изменение настроек политики сканирования

Все настройки снабжены пояснениями, поэтому детально рассматриваться не будут. Чтобы вызвать соответствующее пояснение, необходимо щелкнуть на названии настройки (курсор при этом примет форму стрелки с вопросительным знаком). Тем не менее следует отдельно остановиться на настройке «Safe Check». Включение данной опции отменяет использование тестов, которые могут вызвать отказ узла, подвергающегося сканированию. В незарегистрированной версии Nessus 3.0.3 для Windows эта опция не действует, и «опасные» тесты вообще не проводятся.

Диалоговое окно, в котором осуществляется выбор тестов безопасности (подключаемых модулей), представлено на рис. 8.13. В левой части экрана отображаются группы, на которые поделены все тесты, а в правой, при выборе соответствующей группы, — собственно список тестов. Подключаемые модули сгруппированы по типам уязвимостей и используемому на сканируемом узле программному обеспечению (тип операционной системы, наличие web-сервера, FTP-сервера и пр.).

Если известно, какая операционная система установлена на сканируемом узле, то можно ускорить процедуру сканирования выбором только актуальных для этой системы тестов. В остальных случаях рекомендуется выполнять максимальное число тестов. Если создана политика сканирования с необходимыми настройками, то при запуске сканирования можно выбрать пункт «Use a predefined policy» и далее — требуемую политику.

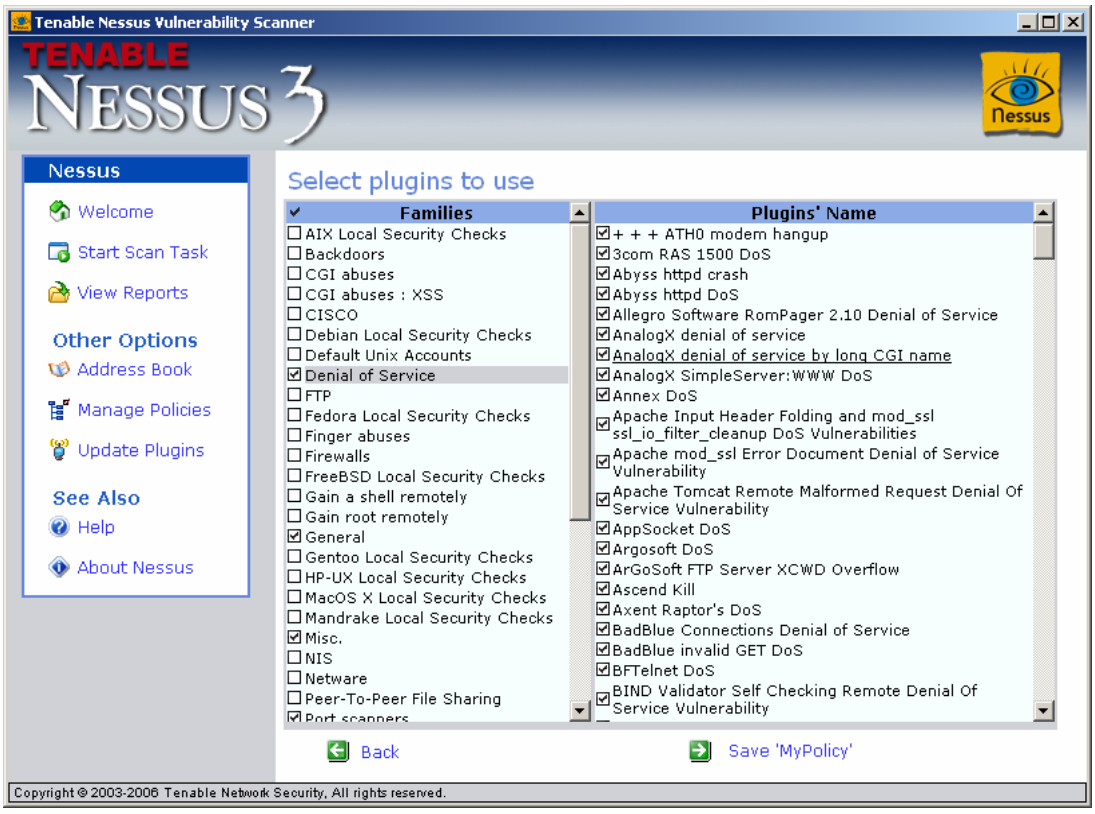


Рис. 8.13. Выбор подключаемых модулей

Диалоговое окно Nessus в процессе сканирования показано на рис. 8.14. Сценарий теста для каждой уязвимости в общем случае уникален. Иногда это лишь подключение к соответствующему порту и получение первичной информации о программном обеспечении сервера, в других случаях дополнительно выполняется ряд запросов, чтобы установить, доступны ли функции, в реализациях которых существуют уязвимости. Существует отдельная категория «опасных» тестов, которые представляют собой реализации атак на отказ в обслуживании (классическим примером является WinNuke). В версии Nessus для Windows сканирование портов и обнаружение узлов в сети — это тесты группы «Port scanners», которые можно включить или выключить.

Результаты работы сканера представляются пользователю в виде отчета, который можно экспортировать в документы формата PDF, HTML или в текстовые файлы. Пример окна Internet Explorer с фрагментом отчета о сканировании показан на рис. 8.15. В качестве сканируемого узла использовалась рабочая станция с установленной операционной системой Windows 2000 Professional Service Pack 4.

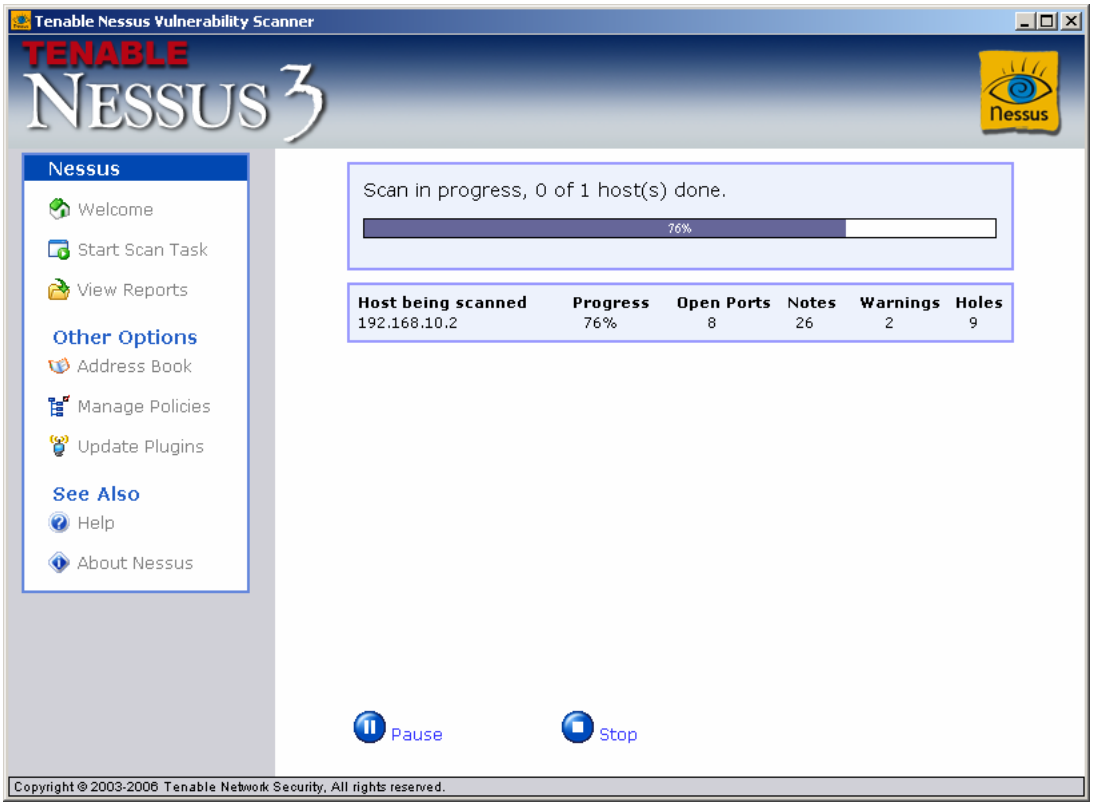


Рис. 8.14. Диалоговое окно Nessus в процессе сканирования

Рассмотрим, как следует интерпретировать результаты сканирования. Для каждого узла в отчете присутствует запись с обобщенными результатами сканирования (рис. 8.16), в которой приведено количество открытых портов (8 Open Ports), примечаний с дополнительной информацией (27 Notes), предупреждений (2 Warnings) и серьезных уязвимостей (10 Holes).

Для каждой обнаруженной уязвимости и для каждого успешного теста в отчете присутствует запись со следующими элементами (рис. 8.17): «Synopsis» — краткий обзор уязвимости, «Description» — ее описание, «Solution» — ссылка на web-страницу с детальным описанием уязвимости и мер, которые необходимо предпринять для ее устранения, «Risk Factor» — информация о степени опасности данной уязвимости и ссылки на эту уязвимость в различных базах данных уязвимостей.

Фактор риска вычисляется в соответствии со стандартом CVSS (Common Vulnerability Scoping System) и представляет собой числовую величину от 0 до 10, где 10 – максимальный уровень опасности, соответствующий критической уязвимости. CVSS – открытый стандарт для подсчета «базового уровня», который количественно представляет величину угрозы от уязвимости. «Базовый уровень» не учитывает количество инцидентов и потерь, связанных с уязвимостью. Этот стандарт также предусматривает возможность подсчета таких уровней, как «временный уровень» (связан со степенью известности и количеством использования уязвимости) и «уровень, зависящий от окружения» (вычисляется на базе информации о системе, на которой находится уязвимость).

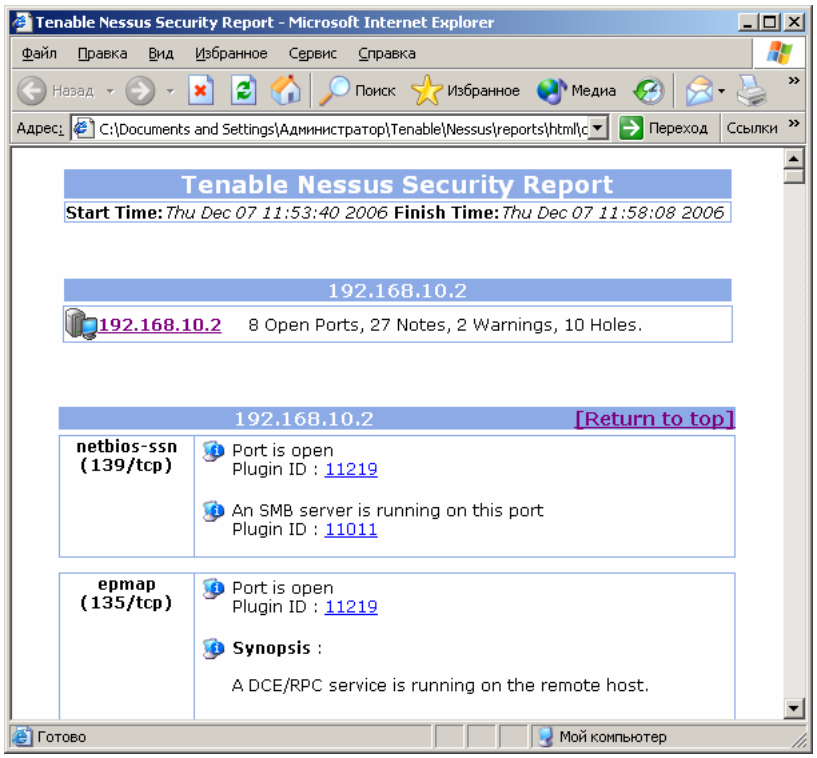


Рис. 8.15. Фрагмент отчета о сканировании

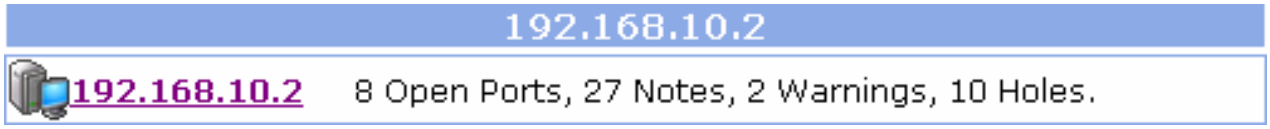
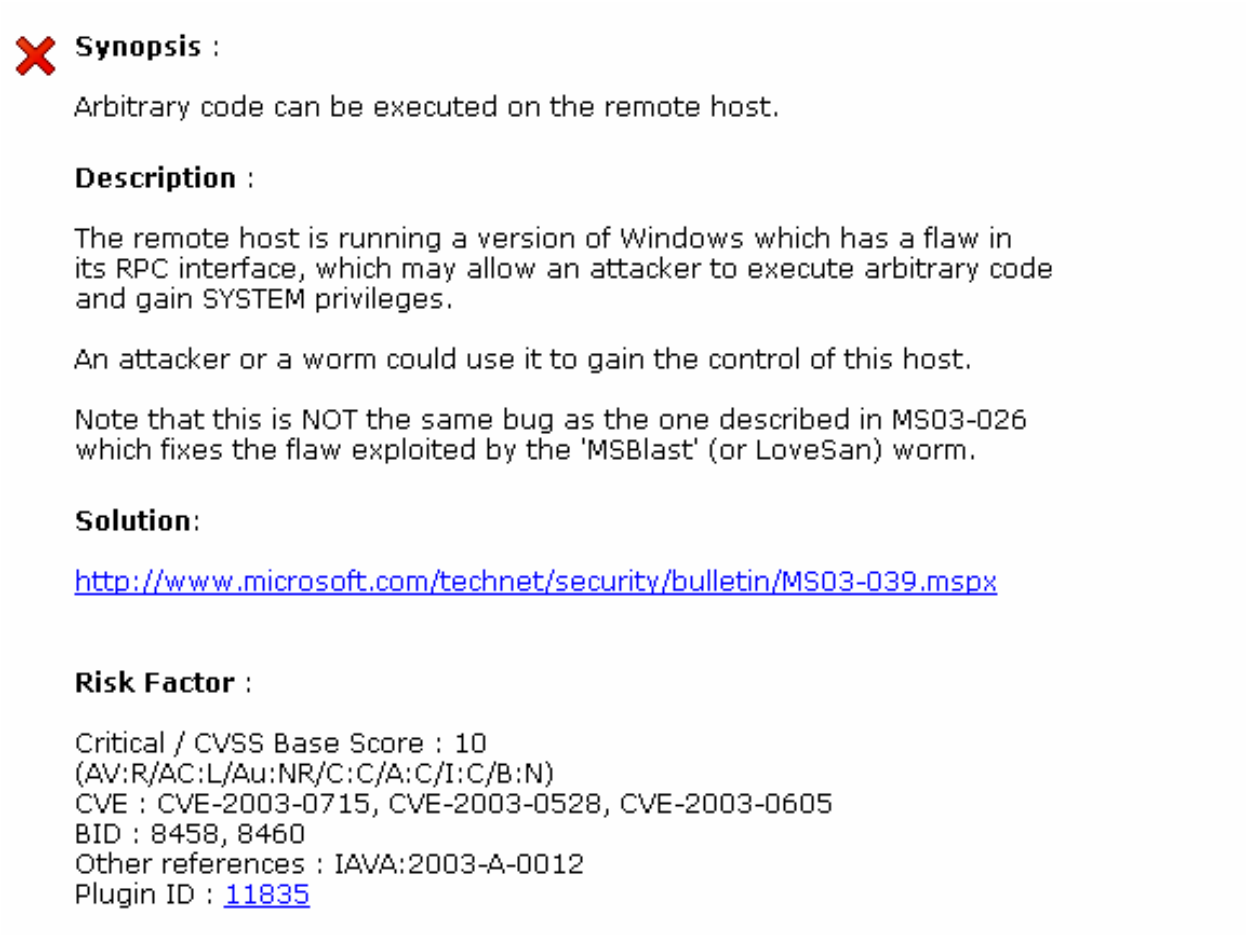


Рис. 8.16. Пример обобщенных результатов сканирования



Synopsis :

Arbitrary code can be executed on the remote host.

Description :

The remote host is running a version of Windows which has a flaw in its RPC interface, which may allow an attacker to execute arbitrary code and gain SYSTEM privileges.

An attacker or a worm could use it to gain the control of this host.

Note that this is NOT the same bug as the one described in MS03-026 which fixes the flaw exploited by the 'MSBlast' (or LoveSan) worm.

Solution:

http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx

Risk Factor :

Critical / CVSS Base Score : 10
(AV:R/AC:L/Au:NR/C:C/A:C/I:C/B:N)
CVE : CVE-2003-0715, CVE-2003-0528, CVE-2003-0605
BID : 8458, 8460
Other references : IAVA:2003-A-0012
Plugin ID : 11835

Рис. 8.17. Пример описания уязвимости

Расчет «базового уровня» выполняется по следующей формуле:
*BS = 10 * AV* AC* Au * ((C1 * C2) + (I1 * I2) + (A1 * A2))*,
где результат округляется до ближайшего целого числа;
*BS (*Base Score*)* – величина «базового уровня»;
AV (Access Vector – вектор доступа): 0,7 — в случае необходимости локального доступа для использования уязвимости; 1 — в случае возможности удаленного использования уязвимости;
AC (Access Complexity — сложность доступа и реализации атаки): 0,8 — высокая, 1 — низкая;
Au (Authentication — аутентификация): 0,6 — требуется, 1 — не требуется;
C1 (Confidentiality Impact – влияние на конфиденциальность): 0 — отсутствует, 0,7 — частично присутствует, 1 — полностью может нарушить конфиденциальность;
I1 (Integrity Impact — влияние на целостность): 0 — отсутствует, 0,7 — частично присутствует, 1 — полностью может нарушить целостность;
A1 (Availability Impact — влияние на доступность): 0 — отсутствует, 0,7 — частично присутствует, 1 — полностью может нарушить доступность;
C2, I2, A2 — коэффициенты воздействия угрозы (Impact Bias) на конфиденциальность, целостность и доступность. Коэффициенты могут принимать значе-

ния: (1/3;1/3;1/3) — уязвимость в равной степени распространяется на все свойства; (0,5; 0,25; 0,25) — уязвимость в большей степени затрагивает конфиденциальность; (0,25; 0,5; 0,25) – уязвимость в большей степени затрагивает целостность; (0,25; 0,25; 0,5) — уязвимость в большей степени затрагивает доступность.

Коэффициенты воздействия угрозы дают возможность назначения приоритетов того или иного свойства информационной системы с точки зрения выполняемых системой функций. Например, если уязвимость в шифрующей файловой системе в равной степени затрагивает (полностью нарушает) и конфиденциальность, и доступность данных, то конфиденциальности должен быть отдан приоритет.

Проведем расчет «базового уровня» на примере найденной уязвимости: *BS = 10*1*1*1*(1*1/3+1*1/3+1*1/3) = 10*. Данный расчет соответствует строке в описании уязвимости: *AV:R/AC:L/Au:NR/C:C/A:C/I:C/B:N*.

Таблица 8.3

Образец оформления перечня найденных уязвимостей

IP-адрес: 192.168.10.1	**Порт:** 445/tcp
Степень опасности: критическая	
Идентификация уязвимости: CVE: CVE-2003-0715, CVE-2003-0528, CVE-2003-0605 BID: 8458, 8460 IAVA: 2003-A-0012	
Краткий обзор: Существует возможность удаленного выполнения произвольного программного кода на данном сетевом узле	
Описание: На узле установлена операционная система Windows, имеющая уязвимость в реализации одного из программных модулей. При наличии доступа злоумышленника к данному узлу по сети существует возможность запуска на нем произвольного программного кода с максимальными полномочиями (полный контроль над узлом)	
Меры по устранению: Требуется перенастройка операционной системы и/или установка обновлений безопасности. Подробная информация может быть получена с официального web-сайта Microsoft: http://www.microsoft.com/technet/security/bulletin/MS03-039.msps.	

Строка с заголовком CVE содержит номер уязвимости в базе данных CVE (Common Vulnerabilities and Exposures — общеизвестные уязвимости и воздействия). CVE представляет собой тезаурус известных уязвимостей, дос-

туп к которому может быть получен по адресу «http://cve.mitre.org». CVE представляет собой не базу уязвимостей, а способ их именования. Например, для уязвимости с именем «CVE-2005-1206» элемент «CVE» указывает на то, что уязвимость уже получила имя «CVE», иначе использовался бы элемент «CAN» — кандидат на имя, 2005 — год, в котором произошло утверждение. Информацию об уязвимости можно найти, используя имя CVE. Уязвимости из примера будет соответствовать ссылка «http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1206».

BID (Bugtraq ID) — номер уязвимости в базе данных BugTraq (адрес в сети Интернет: «http://www.securityfocus.com»). Раздел «Other references» содержит ссылки на другие базы данных, в которых зарегистрирована данная уязвимость, например IAVA (Information Assurance Vulnerability Alert) — метод идентификации уязвимостей, используемый Министерством обороны США. В последней строке (Plugin ID) содержится уникальный номер подключаемого модуля Nessus, который использовался при тестировании данной уязвимости.

В качестве практического задания предлагается протестировать наличие уязвимостей на узлах исследуемой подсети, обнаруженных на предыдущем этапе проведения аудита. Для получения представления о том, какая информация циркулирует в сети в процессе сканирования, используется анализатор сетевого трафика Ethereal. Факт сканирования предлагается обнаружить при помощи системы обнаружения атак Snort. По результатам сканирования необходимо заполнить отчет по образцу, приведенному в табл. 8.3.

ВЫПОЛНИТЬ!

9. Установить на локальном компьютере IP-адрес таким образом, чтобы он оказался внутри исследуемой подсети (192.168.10.0/24). Можно выбрать любой свободный IP-адрес.
10. Запустить серверную часть (службу) Nessus при помощи оснастки «Службы» (***Пуск ⇒ Настройка ⇒ Администрирование ⇒ Службы***). Запустить клиентскую часть Nessus.
11. Очистить log-файлы СОА Snort (удалить содержимое каталога «\snort\log»). Запустить СОА Snort с полным набором правил (из командной строки каталога snort\bin):

    `snort -i <интерфейс> -c ../etc/snort.conf -l ../log`,

    где <интерфейс> — номер интерфейса сетевой платы, полученный при помощи команды:

    `snort –W`.
12. Записать в адресную книгу Nessus IP-адреса узлов сканируемой подсети, которые были обнаружены на предыдущем этапе.
13. Запустить анализатор трафика Ethereal. Включить захват трафика на сетевом интерфейсе, находящемся в одном сегменте со сканируемым узлом.

14. Запустить сканирование узла обнаруженной ранее рабочей станции, используя все тесты за исключением «опасных». Для этого указать пункт «Enable all but dangerous plugins with default settings» в диалоговом окне выбора тестов.
15. Дождаться завершения тестов, выключить захват трафика, прервать работу COA Snort (<Ctrl+C>).
16. Проанализировать результаты отчета, ответить на следующие вопросы:
    a. Какие порты открыты на рабочей станции?
    b. Какие критические уязвимости обнаружены? Что может стать результатом их использования?
    c. Какие конкретные действия следует предпринять для устранения обнаруженных уязвимостей?
    d. Каким образом факт сканирования отражается в файле журнала COA Snort (\snort\log\alert.ids)?
    e. Сколько пакетов было передано по сети в ходе процедуры сканирования? Каков суммарный объем переданной информации?
17. Заполнить отчет о результатах сканирования (образец см. в табл. 8.3).
18. Запустить сканирование узла обнаруженного ранее сервера, используя все тесты за исключением «опасных».
19. Проанализировать результаты теста и заполнить табл. 8.3 по результатам сканирования сервера со стороны внутренней сети.
20. Установить на локальном компьютере IP-адрес таким образом, чтобы он оказался во внешней по отношению к исследуемому компьютеру сети (192.168.200.0/24). Можно выбрать любой свободный IP-адрес.
21. Запустить сканирование сервера, используя все тесты за исключением «опасных».
22. Заполнить отчет о результатах сканирования сервера со стороны внешней сети (образец см. в табл. 8.3). Сравнить результаты с полученными ранее в пункте 19.
23. Провести расчет величины «базового уровня» угрозы для любой критической уязвимости.

8.9. Анализ защищенности web-серверов

Как уже было замечено, аудитор может проводить инструментальные проверки, охватывающие различные элементы ПИБ. Для примера того, как могут проходить такие проверки, рассмотрим возможный набор средств и методов проведения инструментальной проверки подсистемы защиты web-сервера.

Как известно, web-сервер представляет собой клиент-серверное приложение, использующее для передачи данных протокол HTTP и стандартный порт 80/tcp. web-сервер ожидает HTTP-запрос от клиента. При получении HTTP-запроса web-сервер отвечает HTTP-ответом, который может содержать

HTML-, XML-документ либо иной тип данных (изображение, текстовый документ, мультимедийный файл и др.). Если HTTP-запрос от клиента не может быть обработан (ошибка в запросе или неосуществимый запрос), то web-сервер должен послать ответ, содержащий код и описание ошибки.

Поскольку протокол HTTP является протоколом уровня приложений, использующим текстовые команды, посмотреть передаваемые web-сервером данные можно с использованием утилиты NetCat (nc). В примере ниже осуществляется стандартное TCP-подключение с использованием этой утилиты с перенаправлением потока вводимых с клавиатуры данных на сервер:

```
nc <имя или адрес узла> <номер порта>.
```

`nc 192.168.10.3 80`	
`GET / HTTP/1.0`	Серверу передается команда GET, запрашивающая корневой документ сервера, с указанием версии HTTP 1.0. После ввода этой команды необходимо дважды нажать Enter.
`HTTP/1.0 200 OK`	Сервер отвечает кодом 200, означающим, что запрос клиента обработан успешно и ответ сервера содержит затребованные данные.
`Server: Apache/1.3.37 (Unix) PHP/4.4.4`	Строка идентифицирует имя и версию web-сервера.
`Date: Wed, 29 Nov 2006 18:19:11 GMT`	Текущее время и дата системных часов сервера.
`Last-Modified: Sun, 15 Jun 2003 17:34:53 GMT`	Время последней модификации передаваемого документа.
`Content-Type: text/html`	Тип передаваемых данных (HTML).
`Content-Length: 620`	Длина передаваемых данных.
`<html>` `<head>` … `</html>`	Собственно передаваемые данные.

Таким образом, используя простое подключение к web-серверу, можно получить достаточно большой объем информации о сервере: версию сервера, набор подключенных модулей, их версии и др.

Для обследования web-сервера в первую очередь необходимо провести обследование ОС сетевого узла, на котором он запущен. Затем можно использовать тесты, специфичные для web-сервера.

Для автоматизации поиска доступных файлов и каталогов, располагающихся на web-сервере, можно использовать утилиту Cgichk. Для тестирования web-узла 192.168.10.1 с использованием этой утилиты можно воспользоваться командой:

```
> CGICHK.EXE 192.168.10.1
HEADER:
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sun, 15 Oct 2006 18:47:54 GMT
Connection: Keep-Alive
Content-Length: 1295
Content-Type: text/html
Set-Cookie: ASPSESSIONIDQQGGGRUC=IFFBLHDBEBLBFMDEHGNMOOBL; path=/
Cache-control: private
------------------------------------------------------------
DIRECTORIES:
 Found /images (403)
    Found /test (200)
    Found /_private (403)
    Found /scripts (403)
INTEREST:
    Found /cgi-bin/ (403)
SPECIFIC:
    Found /robots.txt (200)
```

Указанная утилита возвращает содержимое заголовков HTTP-ответа, а также проводит поиск web-директорий, располагающихся на сервере. Из вывода этой утилиты видно, что на сервере имеются доступные директории «images», «test», «_privat», «scripts». Исходя из названия этих директорий, можно сделать предположения об информации, которая находится в них. Также указывается список «интересных» директорий, которые могут содержать уязвимые компоненты. Секция вывода «SPECIFIC» содержит дополнительно найденные элементы сайта. Например, найденный файл «robots.txt» может быть использован для поиска скрытых web-директорий сервера.

Для автоматизации поиска известных уязвимостей web-приложений можно использовать утилиту Nikto. Утилита Nikto представляет собой сканер, написанный на интерпретируемом языке Perl, который реализует всестороннее тестирование web-сервера и web-приложений. Данный сканер включает базу о более чем 3200 потенциально опасных файлах и 624 web-серверах. Модули сканирования и база уязвимостей часто обновляются.

Для запуска утилиты nikto можно воспользоваться командой `perl nikto.pl –h 192.168.10.3`, ключ -h предназначен для указания IP-адреса или DNS-имени обследуемого узла. Символом «#» отмечены комментарии к выводу утилиты.

```
> perl nikto.pl -h 192.168.10.3
# Версия утилиты
- Nikto 1.35/1.34
# IP-адрес обследуемого сервера
+ Target IP:       192.168.10.3
# Имя обследуемого сервера (возможно имя виртуального сервера)
+ Target Hostname: 192.168.10.3
```

Номер порта, на котором функционирует сервер
```
+ Target Port:     80
```
Метка времени начала сканирования
```
+ Start Time:      Sun Oct 15 23:42:54 2006
---------------------------------------------------------------------
```
Указание на то, что обследуемый сервер не обязательно будет возвращать
правильный параметр «Server» (тип и версия обследуемого сервера), с
использованием опции –g можно явно задать версию сервера. От этого
параметра будет зависеть состав тестов, которые будет проводить nikto.
```
- Scan is dependent on "Server" string which can be faked, use -g
to override
```
Обнаруженные тип и версия сервера
```
+ Server: Apache/1.3.33 (Debian GNULinux)
```
Перечень доступных серверных HTTP-команд (запросов)
```
+ Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE
```
Указание на то, что команда TRACE должна быть разрешена только для
отладочных задач
```
+ HTTP method 'TRACE' is typically only used for debugging. It
should be disabled. OSVDB-877.
```
Указывает на то, что версия web-сервера Apache на обследуемом узле
является устаревшей, однако до сих пор поддерживается разработчиком и
считается безопасной.
```
+ Apache/1.3.33 appears to be outdated (current is at least
Apache/2.0.54). Apache 1.3.33 is still maintained and considered
secure.
```
Включено индексирование каталога /icons/. Оно должно быть включено
для специальных каталогов. В скобках указывается тип запроса,
с помощью которого была получена информация.
```
+ /icons/ - Directory indexing is enabled, it should only be en-
abled for specific directories (if required). If indexing is not
used all, the /icons directory should be removed. (GET)
```
Используя каталог /server-status, можно получить много информации о
сервере Apache. Рекомендуется ограничить доступ к этому ресурсу.
```
+ /server-status - This gives a lot of Apache information. Com-
ment out appropriate line in httpd.conf or restrict access to al-
lowed hosts. (GET)
```
Неотключенная команда TRACE может быть использована для реализации
атаки типа XSS (межсайтовый скриптинг) или для кражи
идентификационных данных. Указывается ссылка на адрес, по которому
можно получить подробное описание найденной уязвимости.
```
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf
for details (TRACE)
```
Каталог /doc открыт для просмотра.
```
+ /doc/ - The /doc directory is browsable. This may be /usr/doc.
(GET)
```

Было проведено 2563 теста, найдено 5 элементов.

```
+ 2563 items checked - 5 item(s) found on remote host(s)
+ End Time:         Sun Oct 15 23:43:07 2006 (13 seconds)
---------------------------------------------------------------
+ 1 host(s) tested
```

Таким образом, с помощью утилиты nikto аудитор может получить информацию об уровне защищенности web-сервера, сформировать перечень присутствующих уязвимостей различного типа. После обнаружения уязвимых сервисов, аудитор может произвести демонстрацию возможных действий злоумышленника и оценить уровень связанных с каждым классом уязвимостей рисков.

Фрагмент аналитического отчета по результатам тестирования подсистемы защиты web-сервера, установленного на сетевом узле с IP-адресом 192.168.10.3, может иметь следующий вид.

Общее описание ПИБ WEB-сервера

Внутренний web-сервер используется для публикации внутрикорпоративных новостей, документов, регламентов и приказов. Требования по доступности информации, хранящейся на web-сервере, низкие, простой в течение одного дня допустим. Требования по конфиденциальности средние, получение информации, хранящейся на сервере, конкурентами может привести к финансовым потерям. Требования по целостности высокие, искажение информации, хранящейся на сервере, может привести к дезорганизации деятельности многих подразделений и к нарушению нормальной работы предприятия в целом.

В качестве web-сервера используется сервер Apache 1.3.33, установленный на сервере под управлением ОС Debian GNU Linux. Сервер используется для предъявления только статических web-страниц. Доступ к web-серверу разрешен всем пользователям локальной сети. Доступ из внешних сетей запрещен периметровым межсетевым экраном. Дополнительные средства защиты web-сервера (аутентификация, SSL, TLS, защита от DoS-атак) не предусмотрены. Резервирование данных не выполняется. Системные журналы и журналы регистрации хранятся непосредственно на сервере.

Примечание

Данная информация может быть получена от системного администратора, а также путем подключения к web-серверу.

Выявленные риски и рекомендации по их обработке

Сервер поддерживает HTTP-методы GET, HEAD, OPTIONS, TRACE.

Примечание

Информация получена сканером nikto.

Метод OPTION может использоваться для получения списка поддерживаемых HTTP-методов, рекомендуется его отключить.

Метод TRACE используется только для целей отладки, его необходимо отключить, так как существуют техники атак на web-сервер, использующие этот метод.

Существует возможность просмотра каталогов /icons и /doc. Рекомендуется закрыть этот доступ.

Существует возможность просмотра системной информации сервера (например, с помощью следующего запроса web-обозревателя http://192.168.10.3/server-status). Данная функция должна быть отключена, если она не используется. Если она используется, то доступ к этой информации должен предоставляться только определенному набору узлов.

Примечание
Информация обнаружена сканером nikto и дополнительно должна быть проверена вручную.

Протоколы SSL и TLS не используются. В силу высоких требований по доступности рекомендуется внедрение протоколов SSL или TLS для обеспечения целостности данных, получаемых с web-сервера. Для этого можно использовать модуль mod_SSL web-сервера Apache.

К серверу разрешен неавторизованный доступ. Рекомендуется запретить неавторизованный доступ к серверу, так как его функциональное назначение предполагает, что доступ должны получать только сотрудники предприятия. Для прозрачности доступа рекомендуется использовать технологии единой регистрации в рамках всей информационной инфраструктуры.

Системные журналы и журналы регистрации хранятся непосредственно на сервере. Рекомендуется использовать централизованное (в рамках всей информационной инфраструктуры) хранилище системных журналов и журналов регистрации. Это позволит упростить к ним доступ и защитит от модификации. Данные журналов должны анализироваться на регулярной основе.

Web-сервер использует настройки безопасности по умолчанию. Для повышения уровня безопасности web-сервера рекомендуется установить модуль mod-security, осуществляющий обнаружение и предотвращение вторжений на web-сервер. В частности, из арсенала средств защиты модуля mod-security необходимо использовать функцию chroot (выполнение сервера в изолированном файловом пространстве) и маскировку баннера сервера.

Рекомендуется отключить все неиспользуемые функции и модули web-сервера, такие как поддержка CGI и др.

Примечание
Поскольку аудитор находился в рамках модели «серого ящика» (в частности, не было возможности провести анализ конфигурации сервера), он не мог точно определить, какие функции отключены, а какие — нет. Поэтому приводимая рекомендация носит общий характер.

ВЫПОЛНИТЬ!

24. С помощью утилиты netcat (или telnet) подключитесь к web-серверу, функционирующему на узле «Сервер». Определите версию web-сервера и запросите корневую страницу.
25. Последовательно выполните сканирование web-сервера с помощью утилит Cgichk и Nikto.
26. Дополните табл. 8.3. Сделайте вывод о возможности анализа защищенности web-сервера различными методами. Напишите тезисы «Аналитического отчета», описывающие ПИБ, выявленные риски и рекомендации для этого сервера, в предположении, что данный сервер является web-сервером, содержащим информацию о компании, её структуре, новостях и о профиле деятельности.
27. Сформируйте итоговый отчет о проведении тестирования. Укажите перечень первоочередных мероприятий для устранения найденных уязвимостей.

СПИСОК ЛИТЕРАТУРЫ

1. Осипенко, А. Л. Борьба с преступностью в глобальных компьютерных сетях: Международный опыт [Текст]: Монография / А.Л. Осипенко. — М.: Норма, 2004. – 432 с.; 21 см. 3000 экз. – ISBN 5-89123-817-9
2. Запечников, С.В. Основы построения виртуальных частных сетей [Текст]: Учеб. пособие для вузов / С.В. Запечников, Н.Г. Милославская, А.И. Толстой. — М.: Горячая линия–Телеком, 2003. — 249 с. ; 20 см. — 3000 экз. — ISBN 5-93517-139-2
3. Медведовский, И.Д. Атака на Internet [Текст] / И.Д. Медведовский, П.В.Семьянов, Д.Г.Леонов. – 2-е изд., перераб. и доп. – М.: ДМК, 1999. – 336 с.
4. Скрембрей, Дж. Секреты хакеров. Безопасность Windows 2000 – готовые решения [Текст] : [пер. с англ.] / Джоел Скрембрей, Стюарт Мак-Клар. – М.: Вильямс, 2002. – 464 с. : ил. ; 24 см. – Перевод. изд.: Hacking Exposed. Windows 2000: Network security secrets & solutions / Joel Scrambray, Stuart McClure. – 3500 экз. – ISBN 5-8459-0300-9
5. Милославская, Н. Г. Интрасети: доступ в Internet, защита [Текст] : учеб. пособие для вузов / Н. Г. Милославская, А. И. Толстой. – М.: ЮНИТИ-ДАНА, 2000. – 527 с. : ил. ; 21 см. – 6000 экз. – ISBN 5-238-00134-7
6. Компьютерные сети. Учебный курс: Официальное пособие Microsoft для самостоятельной подготовки [Текст] : [пер. с англ.] – 2-е изд., испр. и доп. / Корпорация Майкорософт. – М. : Русская редакция, 1997. – 576 с. : ил. ; 24 см. + 1 электрон. опт. диск. – 3000 экз. – ISBN 5-7502-0101-5 (в пер.)
7. Мандиа, К. Защита от вторжений. Расследование компьютерных преступлений [Текст] : [пер. с англ.] / К. Мандиа, К. Просис. – М.: ЛОРИ, 2005. – 476 с. : ил. ; 24 см. – Перевод. изд.: Incident response: investigating computer crime / Chris Prosise, Kevin Mandia. – 1500 экз. – ISBN 0-07-213182-9 (в пер.)
8. Лукацкий, А. В. Обнаружение атак [Текст] – 2-е изд., перераб. и доп. / А. В. Лукацкий. – СПб: БХВ-Петербург, 2003. – 608 с. : ил. ; 24 см. – 3000 экз. – ISBN 5-94157-246-8
9. Уилсон, Э. Мониторинг и анализ сетей. Методы выявления неисправностей [Текст] : [пер. с англ.] / Эд Уилсон. – М.: ЛОРИ, 2002. – 350 с. : ил. ; 24 см. – Перевод. изд.: Network monitoring and analysys. A protocol approach to troubleshooting / Ed Wilson. – 3200 экз. – ISBN 5-85582-163-3 (в пер.)
10. Рассел, Ч. Microsoft Windows 2000 Server. Справочник администратора [Текст] : [пер. с англ.] – 2-е изд., испр. / Ч. Рассел, Ш. Кроуфорд. – М.: ЭКОМ, 2002. – 1296 с. : ил. ; 25 см. + 1 электрон. опт. диск. – 3000 экз. – ISBN 5-7163-0084-7 (в пер.)

11. Корт, С. С. Теоретические основы защиты информации [Текст] : учеб. пособие для вузов / С. С. Корт. – М.: Гелиос АРВ, 2004. – 240 с. : ил. ; 24 см. – 2000 экз. – ISBN 5-85438-010-2
12. Стивенс, У. Р. Протоколы TCP/IP. Практическое руководство [Текст] : [пер. с англ.] / У. Р. Стивенс. – СПб: БХВ-Петербург, 2003. – 672 с. : ил. ; 24 см. – 5000 экз. – ISBN 5-94157-300-6
13. Кульгин, М. Практика построения компьютерных сетей. Для профессионалов [Текст] / М. Кульгин. – СПб.: Питер, 2001. – 320 с. : ил. ; 24 см. – 5000 экз. – ISBN 5-272-00351-9
14. Jones, A. Computer System Intrusion Detection: A Survey [Текст] / A. Jones, R. Sielken. – Department of Computer Science. University of Virginia, 2000. – 25 с. ; 30 см.
15. Treaster, M. A Survey of Distributed Intrusion Detection Approaches / M. Treaster. – National Center for Supercomputing Applications (NCSA). University of Illinois, 2005. – 13 с. ; 30 см.
16. Kazienko, P. Intrusion Detection Systems (IDS). Part I, II [Электронный ресурс] / P. Kazienko, P. Dorosz. – http:/www.windowsecurity.com, 2003.
17. Snort Users Manual. Версия 2.4.0. [Электронный ресурс]. – http://www.snort.org – 94 с. ; 30 см.
18. Guide to Securing Microsoft Windows 2000 Terminal Services. Vincent J. DiMaria, James F. Barnes, CDR Jerry L. Birdsong, Kathryn A. Merenyi. National Security Agency. 2001.
19. Петренко С. А. Аудит безопасности Intranet [Текст]. / Петренко С. А., Петренко А. А. – М.: ДМК Пресс, 2002. – 416 с.: ил.
20. ГОСТ Р 15408–02. Критерии оценки безопасности информационных технологий [Текст]. – Введ. 2004–01–01 – М.: Изд-во стандартов, 2002.
21. ISO/IEC 17799:2000. Информационные технологии. Свод правил по управлению защитой информации. Международный стандарт [Текст] / ISO/IEC, 2000.
22. K. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, S.M. Thesis, MIT Department of Electrical Engineering and Computer Science, June 1999.

ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. **Ethereal**, разработчик – Gerald Combs (C) 1998-2005, источник – http://www.ethereal.com, версия 0.10.11.
2. **InterNetView**, разработчик – Evgene Ilchenko, источник – http://www.tsu.ru /~evgene/info/inv, версия 2.0.
3. **Netcat**, разработчик – Weld Pond <weld@l0pht.com>, источник – http://www. l0pht.com, версия 1.10.
4. **Nmap**, разработчик – Copyright 2005 Insecure.Com, источник – http://www.insecure.com, версия 3.95.
5. **Snort**, разработчик – Martin Roesch & The Snort Team. Copyright 1998–2005 Sourcefire Inc., et al., источник – http://www.snort.org, версия 2.4.3.
6. **VipNet Office**, разработчик – ОАО Инфотекс, Москва, Россия, источник – http://www.infotecs.ru, версия 2.89 (Windows).
7. **VMware Workstation**, разработчик – VMware Inc, источник – http://www. vmware.com, версия 4.0.0.
8. **WinPCap**, источник – http://winpcap.polito.it.
9. **AdRem Netcrunch**, источник – http://www.adremsoft.com/netcrunch/
10. **Nessus**, источник – http://www.nessus.org

ПРИЛОЖЕНИЕ 1
ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ВИРТУАЛЬНЫХ МАШИН ДЛЯ ИМИТАЦИИ СЕТЕВЫХ СОЕДИНЕНИЙ

Для проведения практических занятий в компьютерных классах используется технология виртуальных машин (система VMware Workstation), позволяющая осуществлять одновременный запуск на одном компьютере нескольких операционных систем и установить между ними сетевые соединения. В зависимости от объема установленной на рабочем месте оперативной памяти может имитироваться наличие от двух сетевых узлов (основного и одного виртуального, требуется минимум 128Мб ОЗУ) до трех (основного и двух виртуальных, требуется минимум 256Мб ОЗУ) и более узлов с установленной ОС Windows 2000.

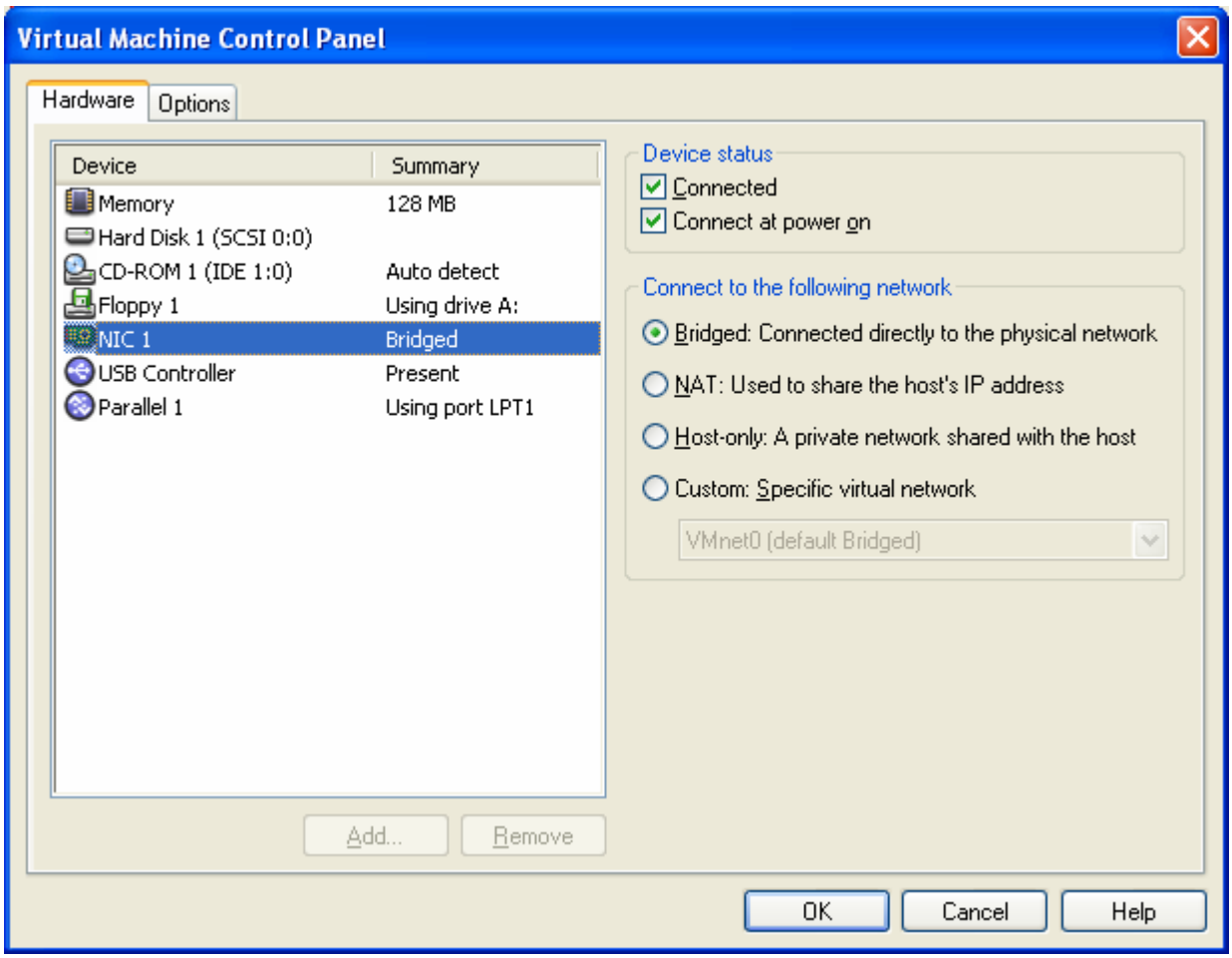


Рис. *а*. Настройка виртуального сетевого адаптера

Для анализа сетевых соединений и сетевых атак достаточно двух узлов. Один в этом случае играет роль атакующего, на нем также может вестись захват сетевого трафика, а другой — роль атакуемого. В качестве атакующего целесообразно использовать основную операционную систему, в качестве атакуемого — виртуальный компьютер. Для работы с межсетевыми экранами

может понадобиться дополнительно третий виртуальный компьютер, выполняющий роль, например, фильтрующего маршрутизатора. Для организации VPN-сети необходимо большее число сетевых узлов, в этом случае целесообразно использовать два основных компьютера с работающими на них виртуальными системами.

При наличии образов операционных систем настройка сетевых соединений в системе VMware Workstation производится следующим образом. Прежде всего определяется IP-адрес основного компьютера и его маска подсети, например, при помощи команды ipconfig.

Далее необходимо вызвать настройки каждой из используемых виртуальных машин (команда в главном меню VMware Workstation ***Edit ⇒ Virtual Mashine Settings…***) и в разделе Hardware выбрать настройки виртуального сетевого адаптера (рис. *а*). Установить пункт «Bridget. Connected directly to the physical network» (Прямое соединение к физической линии).

Далее IP-адрес виртуального компьютера настраивается обычным образом с учетом IP-адреса и маски подсети основного компьютера. Теперь в сети присутствуют два независимых сетевых узла.

Сетевое взаимодействие узлов, в случае соответствия их маски подсети, легко проверить, например, командой Ping (рис. *б*).

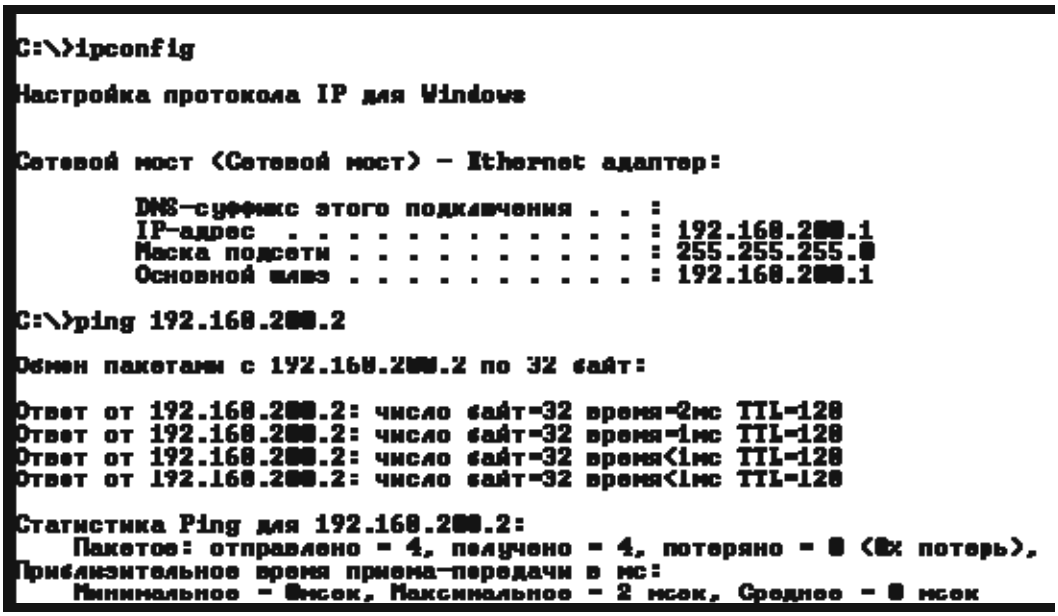


Рис. *б*. Проверка сетевого взаимодействия

ПРИЛОЖЕНИЕ 2
ПЕРЕЧЕНЬ СЕТЕВЫХ СЛУЖБ ОС WINDOWS SERVER 2003

Порт	Тип	Протокол	Наименование системной службы
n/a	GRE	GRE (IP protocol 47)	Routing and Remote Access
n/a	ESP	IPSec ESP (IP protocol 50)	Routing and Remote Access
n/a	AH	IPSec AH (IP protocol 51)	Routing and Remote Access
7	TCP	Echo	Simple TCP/IP Services
7	UDP	Echo	Simple TCP/IP Services
9	TCP	Discard	Simple TCP/IP Services
9	UDP	Discard	Simple TCP/IP Services
13	TCP	Daytime	Simple TCP/IP Services
13	UDP	Daytime	Simple TCP/IP Services
17	TCP	Quotd	Simple TCP/IP Services
17	UDP	Quotd	Simple TCP/IP Services
19	TCP	Chargen	Simple TCP/IP Services
19	UDP	Chargen	Simple TCP/IP Services
20	TCP	FTP default data	FTP Publishing Service
21	TCP	FTP control	FTP Publishing Service
21	TCP	FTP control	Application Layer Gateway Service
23	TCP	Telnet	Telnet
25	TCP	SMTP	Simple Mail Transfer Protocol
25	UDP	SMTP	Simple Mail Transfer Protocol
25	TCP	SMTP	Exchange Server
25	UDP	SMTP	Exchange Server
42	TCP	WINS Replication	Windows Internet Name Service
42	UDP	WINS Replication	Windows Internet Name Service
53	TCP	DNS	DNS Server
53	UDP	DNS	DNS Server
53	TCP	DNS	Internet Connection Firewall/Internet Connection Sharing
53	UDP	DNS	Internet Connection Firewall/Internet Connection Sharing
67	UDP	DHCP Server	DHCP Server
67	UDP	DHCP Server	Internet Connection Firewall/Internet Connection Sharing
69	UDP	TFTP	Trivial FTP Daemon Service
80	TCP	HTTP	Windows Media Services
80	TCP	HTTP	World Wide Web Publishing Service
80	TCP	HTTP	SharePoint Portal Server
88	TCP	Kerberos	Kerberos Key Distribution Center
88	UDP	Kerberos	Kerberos Key Distribution Center
102	TCP	X.400	Microsoft Exchange MTA Stacks
110	TCP	POP3	Microsoft POP3 Service
110	TCP	POP3	Exchange Server

Порт	Тип	Протокол	Наименование системной службы
119	TCP	NNTP	Network News Transfer Protocol
123	UDP	NTP	Windows Time
123	UDP	SNTP	Windows Time
135	TCP	RPC	Message Queuing
135	TCP	RPC	Remote Procedure Call
135	TCP	RPC	Exchange Server
137	TCP	NetBIOS Name Resolution	Computer Browser
137	UDP	NetBIOS Name Resolution	Computer Browser
137	TCP	NetBIOS Name Resolution	Server
137	UDP	NetBIOS Name Resolution	Server
137	TCP	NetBIOS Name Resolution	Windows Internet Name Service
137	UDP	NetBIOS Name Resolution	Windows Internet Name Service
137	TCP	NetBIOS Name Resolution	Net Logon
137	UDP	NetBIOS Name Resolution	Net Logon
137	TCP	NetBIOS Name Resolution	Systems Management Server 2.0
137	UDP	NetBIOS Name Resolution	Systems Management Server 2.0
138	UDP	NetBIOS Datagram Service	Computer Browser
138	UDP	NetBIOS Datagram Service	Messenger
138	UDP	NetBIOS Datagram Service	Server
138	UDP	NetBIOS Datagram Service	Net Logon
138	UDP	NetBIOS Datagram Service	Distributed File System
138	UDP	NetBIOS Datagram Service	Systems Management Server 2.0
138	UDP	NetBIOS Datagram Service	License Logging Service
139	TCP	NetBIOS Session Service	Computer Browser
139	TCP	NetBIOS Session Service	Fax Service
139	TCP	NetBIOS Session Service	Performance Logs and Alerts
139	TCP	NetBIOS Session Service	Print Spooler
139	TCP	NetBIOS Session Service	Server
139	TCP	NetBIOS Session Service	Net Logon
139	TCP	NetBIOS Session Service	Remote Procedure Call Locator
139	TCP	NetBIOS Session Service	Distributed File System
139	TCP	NetBIOS Session Service	Systems Management Server 2.0
139	TCP	NetBIOS Session Service	License Logging Service
143	TCP	IMAP	Exchange Server
161	UDP	SNMP	SNMP Service
162	UDP	SNMP Traps Outbound	SNMP Trap Service
389	TCP	LDAP Server	Local Security Authority
389	UDP	LDAP Server	Local Security Authority
389	TCP	LDAP Server	Distributed File System
389	UDP	LDAP Server	Distributed File System
443	TCP	HTTPS	HTTP SSL
443	TCP	HTTPS	World Wide Web Publishing Service
443	TCP	HTTPS	SharePoint Portal Server
445	TCP	SMB	Fax Service

Порт	Тип	Протокол	Наименование системной службы
445	UDP	SMB	Fax Service
445	TCP	SMB	Print Spooler
445	UDP	SMB	Print Spooler
445	TCP	SMB	Server
445	UDP	SMB	Server
445	TCP	SMB	Remote Procedure Call Locator
445	UDP	SMB	Remote Procedure Call Locator
445	TCP	SMB	Distributed File System
445	UDP	SMB	Distributed File System
445	TCP	SMB	License Logging Service
445	UDP	SMB	License Logging Service
500	UDP	IPSec ISAKMP	IPSec Services
515	TCP	LPD	TCP/IP Print Server
548	TCP	File Server for Macintosh	File Server for Macintosh
554	TCP	RTSP	Windows Media Services
563	TCP	NNTP over SSL	Network News Transfer Protocol
593	TCP	RPC over HTTP	Remote Procedure Call
593	TCP	RPC over HTTP	Exchange Server
636	TCP	LDAP SSL	Local Security Authority
636	UDP	LDAP SSL	Local Security Authority
993	TCP	IMAP over SSL	Exchange Server
995	TCP	POP3 over SSL	Exchange Server
1270	TCP	MOM-Encrypted	Microsoft Operations Manager 2000
1433	TCP	SQL over TCP	Microsoft SQL Server
1433	TCP	SQL over TCP	MSSQL$UDDI
1434	UDP	SQL Probe	Microsoft SQL Server
1434	UDP	SQL Probe	MSSQL$UDDI
1645	UDP	Legacy RADIUS	Internet Authentication Service
1646	UDP	Legacy RADIUS	Internet Authentication Service
1701	UDP	L2TP	Routing and Remote Access
1723	TCP	PPTP	Routing and Remote Access
1755	TCP	MMS	Windows Media Services
1755	UDP	MMS	Windows Media Services
1801	TCP	MSMQ	Message Queuing
1801	UDP	MSMQ	Message Queuing
1812	UDP	RADIUS Authentication	Internet Authentication Service
1813	UDP	RADIUS Accounting	Internet Authentication Service
1900	UDP	SSDP	SSDP Discovery Service
2101	TCP	MSMQ-DCs	Message Queuing
2103	TCP	MSMQ-RPC	Message Queuing
2105	TCP	MSMQ-RPC	Message Queuing
2107	TCP	MSMQ-Mgmt	Message Queuing
2393	TCP	OLAP Services 7.0	SQL Server: Downlevel OLAP Client Support

Порт	Тип	Протокол	Наименование системной службы
2394	TCP	OLAP Services 7.0	SQL Server: Downlevel OLAP Client Support
2460	UDP	MS Theater	Windows Media Services
2535	UDP	MADCAP	DHCP Server
2701	TCP	SMS Remote Control (control)	SMS Remote Control Agent
2701	UDP	SMS Remote Control (control)	SMS Remote Control Agent
2702	TCP	SMS Remote Control (data)	SMS Remote Control Agent
2702	UDP	SMS Remote Control (data)	SMS Remote Control Agent
2703	TCP	SMS Remote Chat	SMS Remote Control Agent
2703	UPD	SMS Remote Chat	SMS Remote Control Agent
2704	TCP	SMS Remote File Transfer	SMS Remote Control Agent
2704	UDP	SMS Remote File Transfer	SMS Remote Control Agent
2725	TCP	SQL Analysis Services	SQL Analysis Server
2869	TCP	UPNP	Universal Plug and Play Device Host
2869	TCP	SSDP event notification	SSDP Discovery Service
3268	TCP	Global Catalog Server	Local Security Authority
3269	TCP	Global Catalog Server	Local Security Authority
3343	UDP	Cluster Services	Cluster Service
3389	TCP	Terminal Services	NetMeeting Remote Desktop Sharing
3389	TCP	Terminal Services	Terminal Services
3527	UDP	MSMQ-Ping	Message Queuing
4011	UDP	BINL	Remote Installation
4500	UDP	NAT-T	Routing and Remote Access
5000	TCP	SSDP legacy event notification	SSDP Discovery Service
5004	UDP	RTP	Windows Media Services
5005	UDP	RTCP	Windows Media Services
42424	TCP	ASP.Net Session State	ASP.NET State Service
51515	TCP	MOM-Clear	Microsoft Operations Manager 2000

Учебное издание

Андрончик Александр Николаевич
Богданов Валентин Викторович
Домуховский Николай Анатольевич
Коллеров Андрей Сергеевич
Синадский Николай Игоревич
Хорьков Дмитрий Алексеевич
Щербаков Михаил Юрьевич

ЗАЩИТА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ
ПРАКТИЧЕСКИЙ КУРС

Редактор *Н. В. Рощина*
Компьютерный набор *авторов*

ИД № 06263 от 12.11.2001 г.

Подписано в печать 27.11.2007		Формат 60x84 1/16
Бумага писчая	Плоская печать	Усл. печ. л 14,3
Уч. изд. л. 15,9	Тираж 300 экз.	Заказ

Редакционно-издательский отдел ГОУ ВПО УГТУ-УПИ
Ризография НИЧ ГОУ ВПО УГТУ-УПИ
620002, Екатеринбург, ул. Мира, 19