Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

В. М. Логин, И. Н. Цырельчук, В. В. Хорошко

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ БЕЗОПАСНОСТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия для специальности 1-39 81 01 «Компьютерные технологии проектирования электронных систем»

Минск БГУИР 2015

УДК 004.056(076.5)
ББК 32.973.26-018.2я73
Л69

Р е ц е н з е н т ы:
кафедра информационно-измерительной техники и технологий
Белорусского национального технического университета
(протокол №16 от 06.05.2014);

директор ОАО «КБТЭМ-ОМО»,
доктор технических наук С. М. Аваков

Логин, В. М.

Л69 Интегрированные системы безопасности. Лабораторный практикум : пособие / В. М. Логин, И. Н. Цырельчук, В. В. Хорошко. – Минск : БГУИР, 2015. – 64 с. : ил.

ISBN 978-985-543-106-1.

Рассматриваются различные специальные программные средства для реализации интегрированных систем безопасности: межсетевые экраны для различных операционных систем, сканеры портов и сканеры уязвимостей, а также сетевые анализаторы.

Предназначено для учащихся второй ступени высшего образования и может быть использовано аспирантами, инженерами, а также другими специалистами, занимающимися вопросами обеспечения безопасности и защиты информации в интегрированных системах безопасности.

УДК 004.056(076.5)
ББК 32.973.26-018.2я73

ISBN 978-985-543-106-1

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Тенденции современного развития компьютерных технологий неразрывно связаны с процессами широкой автоматизации и интеграции, которые касаются не только систем безопасности, но и всех остальных систем, предназначенных для автоматизации управления жизнеобеспечением и функционированием жилого здания, офиса, предприятия или любого другого объекта. Логическим развитием такой интеграции явилось создание интегрированных систем безопасности (ИСБ) с широкими функциональными возможностями, позволяющими автоматизировать также управление инженерными системами здания или объекта. Основой таких ИСБ служит единая аппаратно-программная платформа, представляющая собой автоматизированную систему управления (АСУ) с многоуровневой сетевой структурой, имеющую общий центр управления на базе локальной компьютерной сети и содержащую линии коммуникаций, контроллеры приема информации, управляющие контроллеры и другие периферийные устройства, предназначенные для сбора и обработки информации от различных датчиков (в том числе от извещателей пожарной и охранной сигнализации), а также для управления различными средствами автоматизации (оповещение, противопожарная автоматика и пожаротушение, инженерные системы и т. д.)

Современные ИСБ строятся на основе иерархической сетевой структуры, в которую входят компьютерные сети, а также локальные сети различного уровня сложности специальных вычислительных устройств – контроллеров.

Если компьютер подключен к компьютерной сети или Интернету, то он уязвим для вирусов, атак злоумышленников и других вторжений. Для защиты компьютера от этих опасностей необходимо, чтобы на нем постоянно работали межсетевой экран (брандмауэр) и антивирусное программное обеспечение (ПО) с актуальными обновлениями. Кроме того, необходимо, чтобы все последние обновления операционной системы (ОС) были также установлены на вашем компьютере.

Для реализации комплексной сетевой безопасности ИСБ можно воспользоваться межсетевыми экранами для различных ОС, сканерами портов и сканерами уязвимостей, а также сетевыми анализаторами.

В данном пособии рассмотрены основные теоретические сведения, а также приведена практическая часть с вышеперечисленным специализированным ПО.

ЛАБОРАТОРНАЯ РАБОТА №1
МЕЖСЕТЕВОЙ ЭКРАН ДЛЯ ОС WINDOWS

Цель работы: ознакомиться с различными типами межсетевых экранов и теорией их построения; изучить работу Outpost Firewall Pro: самого распространенного межсетевого экрана для систем Windows 2000 SP4, Windows XP, Windows Server 2003, Windows Vista или Windows 7.

1.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Межсетевой экран (firewall) – это устройство контроля доступа в сеть, предназначенное для блокировки всего трафика, за исключением разрешенных данных. Этим оно отличается от маршрутизатора, функцией которого является доставка трафика в пункт назначения в максимально короткие сроки.

Существует мнение, что маршрутизатор также может играть роль межсетевого экрана. Однако между этими устройствами существует одно принципиальное различие: маршрутизатор предназначен для быстрой маршрутизации трафика, а не для его блокировки. Межсетевой экран представляет собой средство защиты, которое пропускает определенный трафик из потока данных, а маршрутизатор является сетевым устройством, которое можно настроить на блокировку определенного трафика. Кроме того, межсетевые экраны, как правило, обладают большим набором настроек. Прохождение трафика на межсетевом экране можно настраивать по службам, IP-адресам отправителя и получателя, по идентификаторам пользователей, запрашивающих службу. Межсетевые экраны позволяют осуществлять централизованное управление безопасностью.

1.1.1. Определение типов межсетевых экранов

Существуют два основных типа межсетевых экранов: межсетевые экраны прикладного уровня и межсетевые экраны с пакетной фильтрацией. В их основе лежат различные принципы работы, но при правильной настройке оба типа устройств обеспечивают правильное выполнение функций безопасности, заключающихся в блокировке запрещенного трафика.

1.1.2. Межсетевые экраны прикладного уровня

Межсетевые экраны прикладного уровня, или прокси-экраны, представляют собой программные пакеты, базирующиеся на операционных системах общего назначения (таких, как Windows и Unix) или на аппаратной платформе межсетевых экранов. Межсетевой экран обладает несколькими интерфейсами, по одному на каждую из сетей, к которым он подключен. Набор правил политики определяет, каким образом трафик передается из одной сети в другую. Ес-

ли в правиле отсутствует явное разрешение на пропуск трафика, межсетевой экран отклоняет или аннулирует пакеты.

Правила политики безопасности усиливаются посредством использования модулей доступа. В межсетевом экране прикладного уровня каждому разрешаемому протоколу должен соответствовать свой собственный модуль доступа. Лучшими модулями доступа считаются те, которые построены специально для разрешаемого протокола. Например, модуль доступа FTP предназначен для протокола FTP и может определять, соответствует ли проходящий трафик этому протоколу и разрешен ли этот трафик правилами политики безопасности.

При использовании межсетевого экрана прикладного уровня все соединения проходят через него (рис. 1.1). Как показано на рисунке, соединение начинается на системе-клиенте и поступает на внутренний интерфейс межсетевого экрана. Межсетевой экран принимает соединение, анализирует содержимое пакета и используемый протокол и определяет, соответствует ли данный трафик правилам политики безопасности. Если это так, то межсетевой экран инициирует новое соединение между своим внешним интерфейсом и системой-сервером.

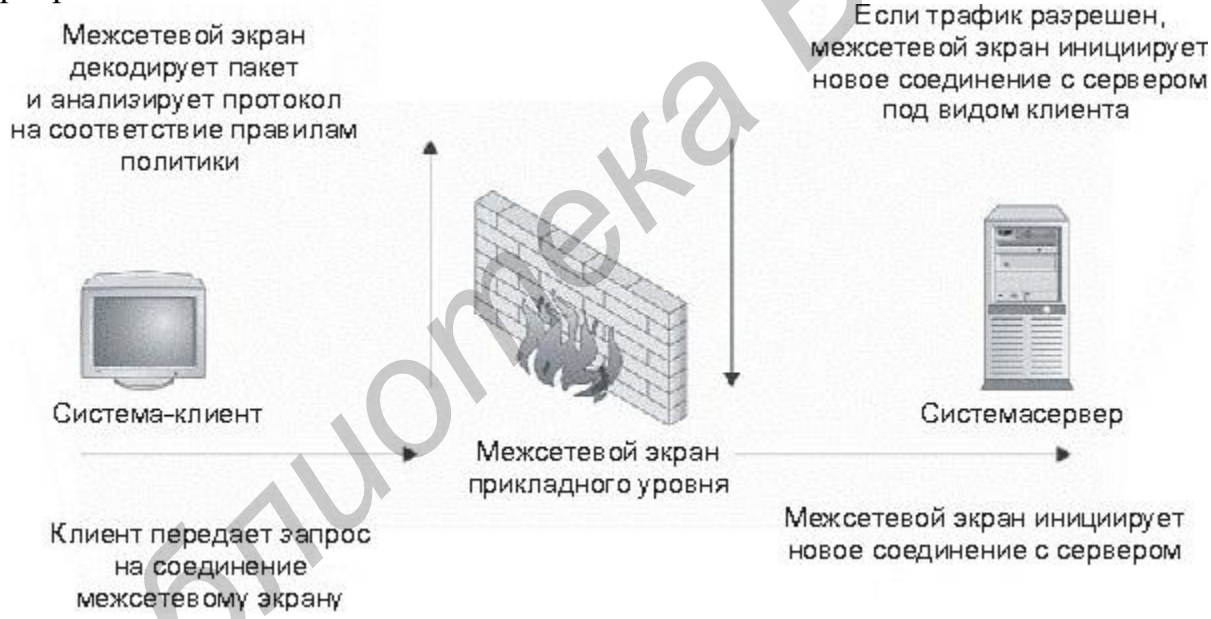

Рис. 1.1. Соединения модуля доступа межсетевого экрана прикладного уровня

Межсетевые экраны прикладного уровня используют модули доступа для входящих подключений. Модуль доступа в межсетевом экране принимает входящее подключение и обрабатывает команды перед отправкой трафика получателю. Таким образом, межсетевой экран защищает системы от атак, выполняемых посредством приложений.

Дополнительным преимуществом архитектуры данного типа является то, что при ее использовании очень сложно, иногда невозможно, «скрыть» трафик внутри других служб. Например, некоторые программы контроля над системой, такие, как NetBus и Back Orifice, могут быть настроены на использование лю-

бого предпочитаемого пользователем порта. Следовательно, их можно настроить на использование порта 80 (HTTP). При использовании правильно настроенного межсетевого экрана прикладного уровня модуль доступа не сможет распознавать команды, поступающие через соединение, и соединение, скорее всего, не будет установлено.

Межсетевой экран также скрывает адреса систем, расположенных по другую сторону от него. Так как все соединения инициируются и завершаются на интерфейсах межсетевого экрана, внутренние системы сети не видны напрямую извне, что позволяет скрыть схему внутренней адресации сети.

1.1.3. Межсетевые экраны с пакетной фильтрацией

Межсетевые экраны с пакетной фильтрацией могут также быть программными пакетами, базирующимися на операционных системах общего назначения (таких, как Windows и Unix) либо на аппаратных платформах межсетевых экранов. Межсетевой экран имеет несколько интерфейсов, по одному на каждую из сетей, к которым подключен экран. Аналогично межсетевым экранам прикладного уровня доставка трафика из одной сети в другую определяется набором правил политики. Если правило не разрешает явным образом определенный трафик, то соответствующие пакеты будут отклонены или аннулированы межсетевым экраном.

Правила политики усиливаются посредством использования фильтров пакетов. Фильтры изучают пакеты и определяют, является ли трафик разрешенным согласно правилам политики и состоянию протокола (проверка с учетом состояния). Если протокол приложения функционирует через TCP, определить состояние относительно просто, так как TCP сам по себе поддерживает состояния. Это означает, что когда протокол находится в определенном состоянии, разрешена передача только определенных пакетов. Рассмотрим в качестве примера последовательность установки соединения. Первый ожидаемый пакет – пакет SYN. Межсетевой экран обнаруживает этот пакет и переводит соединение в состояние SYN. В данном состоянии ожидается один из двух пакетов – либо SYN ACK (опознавание пакета и разрешение соединения), либо пакет RST (сброс соединения по причине отказа в соединении получателем). Если в данном соединении появятся другие пакеты, межсетевой экран аннулирует или отклонит их, так как они не подходят для данного состояния соединения, даже если соединение разрешено набором правил.

Если протоколом соединения является UDP, межсетевой экран с пакетной фильтрацией не может использовать присущее протоколу состояние, вместо чего отслеживает состояние трафика UDP. Как правило, межсетевой экран принимает внешний пакет UDP и ожидает входящий пакет от получателя, соответствующий исходному пакету по адресу и порту, в течение определенного времени. Если пакет принимается в течение этого отрезка времени, его передача разрешается. В противном случае межсетевой экран определяет, что трафик UDP не является ответом на запрос, и аннулирует его.

При использовании межсетевого экрана с пакетной фильтрацией соединения не прерываются на межсетевом экране (рис. 1.2), а направляются непосредственно к конечной системе. При поступлении пакетов межсетевой экран выясняет, разрешены ли данный пакет и состояние соединения правилами политики. Если это так, пакет передается по своему маршруту. В противном случае пакет отклоняется или аннулируется.

Рис. 1.2. Передача трафика через межсетевой экран с фильтрацией пакетов

Межсетевые экраны с фильтрацией пакетов не используют модули доступа для каждого протокола и поэтому могут использоваться с любым протоколом, работающим через IP.

Как правило, межсетевые экраны с фильтрацией пакетов имеют возможность поддержки большего объема трафика, так как в них отсутствует нагрузка, создаваемая дополнительными процедурами настройки и вычисления, имеющими место в программных модулях доступа.

Межсетевые экраны, работающие только посредством фильтрации пакетов, не используют модули доступа, и поэтому трафик передается от клиента непосредственно на сервер. Если сервер будет атакован через открытую службу, разрешенную правилами политики межсетевого экрана, межсетевой экран никак не отреагирует на атаку. Межсетевые экраны с пакетной фильтрацией также позволяют видеть извне внутреннюю структуру адресации. Внутренние адреса скрывать не требуется, так как соединения не прерываются на межсетевом экране.

1.1.4. Гибридные межсетевые экраны

Как и многие другие устройства, межсетевые экраны изменяются и совершенствуются с течением времени, то есть эволюционируют. Производители межсетевых экранов прикладного уровня в определенный момент пришли к

выводу, что необходимо разработать метод поддержки протоколов, для которых не существует определенных модулей доступа. Вследствие этого увидела свет технология модуля доступа Generic Services Proxy (GSP). GSP разработана для поддержки модулями доступа прикладного уровня других протоколов, необходимых системе безопасности и при работе сетевых администраторов. В действительности GSP обеспечивает работу межсетевых экранов прикладного уровня в качестве экранов с пакетной фильтрацией.

В то время как базовая функциональность межсетевых экранов обоих типов осталась прежней, что является причиной большинства «слабых мест» этих устройств, сегодня на рынке присутствуют гибридные межсетевые экраны. Практически невозможно найти межсетевой экран, функционирование которого построено исключительно на прикладном уровне или фильтрации пакетов. Это обстоятельство отнюдь не является недостатком, так как оно позволяет администраторам, отвечающим за безопасность, настраивать устройство для работы в конкретных условиях.

1.1.5. Разработка конфигурации межсетевого экрана

Теперь давайте рассмотрим некоторые стандартные сетевые архитектуры и выясним, каким образом следует настраивать сетевой экран в той или иной конкретной ситуации. В данном случае подразумевается, что в организации присутствуют указанные ниже системы, которые принимают входящие соединения из Интернета:

– веб-сервер, работающий только через порт 80;

– почтовый сервер, работающий только через порт 25, принимающий всю входящую и отправляющий всю исходящую почту.

Существует внутренняя система DNS, которая запрашивает системы Интернета для преобразования имен в адреса, однако в организации отсутствует своя собственная главная внешняя DNS.

Интернет-политика организации позволяет внутренним пользователям применять следующие протоколы:

– HTTP;

– HTTPS;

– FTP;

– Telnet;

– SSH.

На базе этой политики можно построить правила политики для различных архитектур.

Архитектура 1: системы за пределами межсетевого экрана, доступные в сети Интернет. На рис. 1.3 показано размещение этих систем между сетевым экраном и внешним маршрутизатором. В табл. 1.1 приведены правила межсетевого экрана.

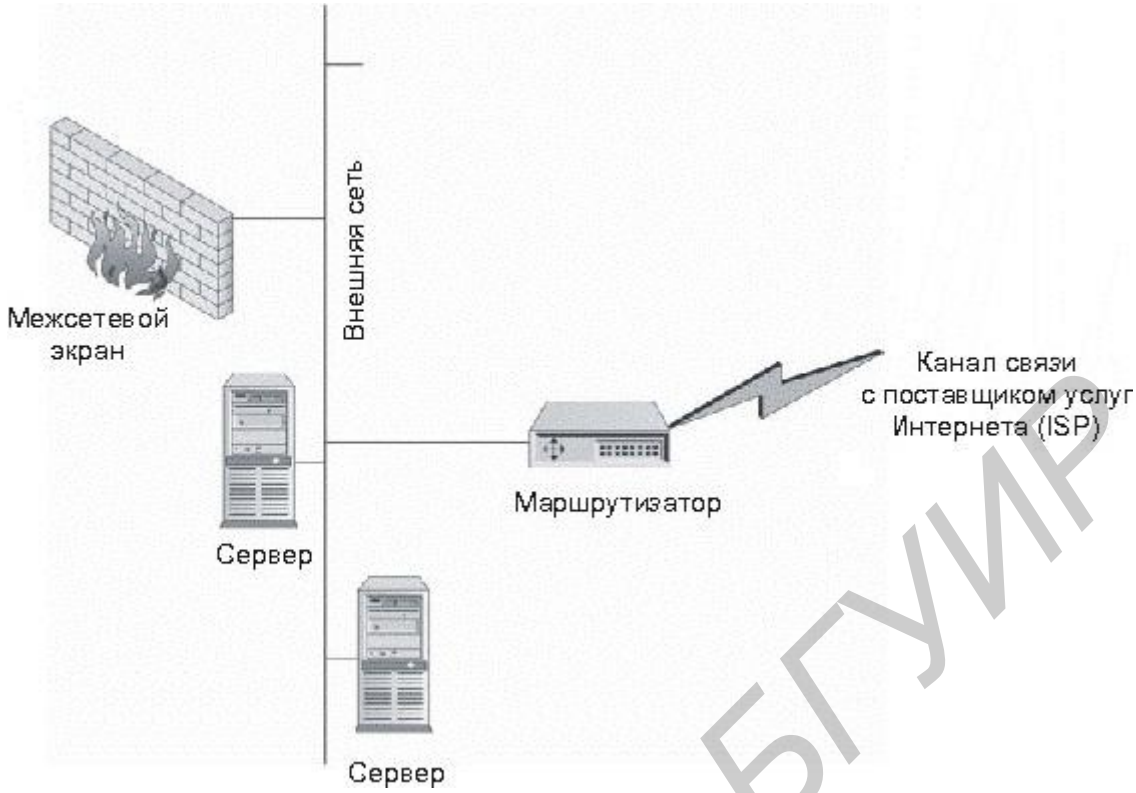

Рис. 1.3. Системы за пределами межсетевого экрана, доступные в сети Интернет

Таблица 1.1

Правила межсетевого экрана для расположенных за пределами межсетевого экрана систем, доступных в сети Интернет

Исходный IP	**Конечный IP**	**Служба**	**Действие**
Внутренний почтовый сервер	Почтовый сервер	SMTP	Принятие
Внутренняя сеть	Почтовый сервер	http, https, ftp, telnet, ssh	Принятие
Внутренняя DNS	Любой	DNS	Принятие
Любой	Любой	Любая	Сброс

На маршрутизаторе может быть установлена фильтрация, позволяющая поступать на веб-сервер только внешним данным HTTP и передавать на почтовый сервер только поступающие извне данные SMTP. Как видно из приведенных правил, независимо от того, какой тип межсетевого экрана используется, веб-сервер и почтовый сервер не защищены межсетевым экраном. В данном случае межсетевой экран лишь защищает внутреннюю сеть организации.

Архитектура 2: один межсетевой экран (рис. 1.4). В данной архитектуре используется один межсетевой экран для защиты как внутренней сети, так и любых других систем, доступных в Интернете. Эти системы располагаются в отдельной сети. В табл. 1.2 приведены правила межсетевого экрана.

Рис. 1.4. Один межсетевой экран

Таблица 1.2

Правила межсетевого экрана для архитектуры с одним межсетевым экраном

Исходный IP	**Конечный IP**	**Служба**	**Действие**
Любой	Веб-сервер	HTTP	Принятие
Любой	Почтовый сервер	SMTP	Принятие
Почтовый сервер	Любой	SMTP	Принятие
Внутренняя сеть	Любой	HTTP, HTTPS, FTP, telnet, SSH	Принятие
Внутренняя DNS	Любой	DNS	Принятие
Любой	Любой	Любая	Сброс

Как видно из табл. 1.2, правила практически аналогичны правилам архитектуры 1. Межсетевой экран дополняет правила, которые использовались в маршрутизаторе в предыдущей архитектуре.

Архитектура 3: двойные межсетевые экраны. Третья архитектура использует двойные межсетевые экраны (рис. 1.5). Доступные в Интернете системы располагаются между межсетевыми экранами, а внутренняя сеть расположена за вторым межсетевым экраном. В табл. 1.3 приведены правила для межсетевого экрана 1.

Не следует ограничивать область действия межсетевых экранов одними лишь Интернет-соединениями. Межсетевой экран представляет собой устройство, которое может использоваться в любой ситуации, требующей контроля доступа. В частности, данные устройства можно использовать во внутренних сетях, которые необходимо защищать от других внутренних систем. Секретные внутренние сети могут содержать компьютеры с особо важной информацией или функциями либо сети, в которых проводятся эксперименты над сетевым оборудованием.

Хорошим примером секретных сетей являются банковские сети. Каждый вечер банки связываются с системой федерального резерва для передачи де-

нежных средств. Ошибки в этих сетях могут стоить банкам больших денег. Системы, управляющие такими соединениями, являются крайне секретными и жизненно важными для банковских структур. Для ограничения доступа из других подразделений банка к этим системам можно установить межсетевой экран.

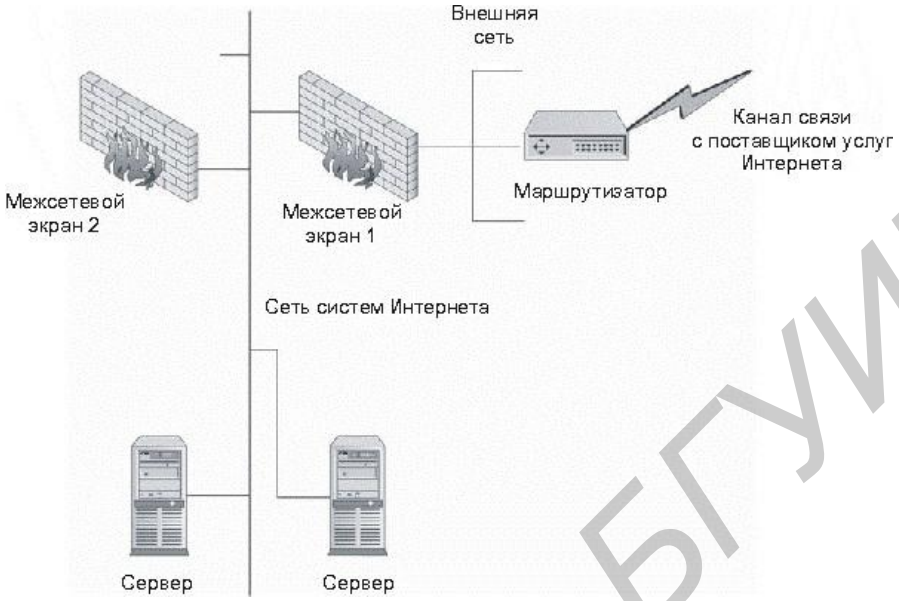

Рис. 1.5. Двойные межсетевые экраны

Таблица 1.3

Правила межсетевого экрана 1 в архитектуре с двумя межсетевыми экранами

Исходный IP	Конечный IP	Служба	Действие
Любой	Веб-сервер	HTTP	Принятие
Любой	Почтовый сервер	SMTP	Принятие
Почтовый сервер	Любой	SMTP	Принятие
Внутренняя сеть	Любой	HTTP, HTTPS, FTP, telnet, SSH	Принятие
Внутренняя DNS	Любой	DNS	Принятие
Любой	Любой	Любая	Сброс

Как видно из табл. 1.3, правила в данном случае аналогичны правилам межсетевого экрана в архитектуре 2. Но еще имеется и второй межсетевой экран. Правила для межсетевого экрана 2 приведены в табл. 1.4.

Таблица 1.4

Правила межсетевого экрана 2 в архитектуре с двойным межсетевым экраном

Исходный IP	Конечный IP	Служба	Действие
Внутренний почтовый сервер	Почтовый сервер	SMTP	Принятие
Внутренняя сеть	Любой	HTTP, HTTPS, FTP, telnet, SSH	Принятие
Внутренняя DNS	Любой	DNS	Принятие
Любой	Любой	Любая	Сброс

1.1.6. Построение набора правил межсетевого экрана

Качественно созданный набор правил не менее важен, чем аппаратная платформа. Большая часть межсетевых экранов работает по принципу «первого

соответствия» при принятии решения о передаче или отклонении пакета. При построении набора правил согласно алгоритму «первого соответствия» наиболее специфичные правила располагаются в верхней части набора правил, а наименее специфичные (то есть более общие) – в нижней части набора. Такое размещение правил гарантирует, что общие правила не перекрывают собой более специфичные. Данный подход хорош в общем плане, однако он не решает проблему производительности межсетевого экрана. Чем больше правил необходимо проверять для каждого пакета, тем больше вычислений должен производить межсетевой экран. При разработке качественного набора правил следует принимать в расчет это обстоятельство, так как от него зависит уровень эффективности работы межсетевого экрана. Для повышения эффективности работы экрана следует оценить ожидаемую нагрузку трафика на нем и упорядочить трафик по типам. Как правило, наибольший объем занимает трафик HTTP.

Для повышения эффективности межсетевого экрана следует поместить правила, относящиеся к HTTP, вверху набора правил. Это означает, что правило, позволяющее внутренним системам использовать HTTP для подключения к любой системе в Интернете, и правило, разрешающее внешним пользователям осуществлять доступ к веб-сайту организации, должны быть расположены очень близко к верхней границе набора правил. Единственными правилами, которые должны находиться выше двух упомянутых правил, являются специфичные правила отказа в доступе, относящиеся к протоколу HTTP.

1.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Настроить Outpost Firewall Pro согласно нижеописанным требованиям. Для более качественной настройки программы пользоваться «Руководством пользователя программой Outpost Firewall Pro.pdf».

В какой версии ОС Windows выполнять все нижеописанные действия, уточнить у преподавателя!

1. Настройка сетевых соединений:
   – настроить политики безопасности;
   – настроить локальную сеть.
2. Предотвращение сетевых атак:
   – настроить уровень обнаружения атак;
   – настроить защиту от Ethernet-атак;
   – настроить сканер портов.
3. Наблюдение за сетевой активностью:
   – проанализировать сетевую активность;
   – проанализировать список используемых портов.
4. Защита от вредоносного ПО:
   – настройка графика проверки системы;
   – настройка постоянной защиты от вредоносных программ;
   – настройки сканирования почтовых вложений.

5. Контроль веб-активности:

– настроить уровни веб-контроля;
– настройка блокировщика рекламы;
– настроить блокировщик шпионских сайтов.
– блокировать передачу персональных данных.

1.3. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Результаты выполнения практического задания.
5. Выводы.

1.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как могут быть реализованы межсетевые экраны?
2. Как называется межсетевой экран, использующий модули доступа для контроля над соединениями?
3. Перечислите достоинства межсетевого экрана прикладного уровня.
4. Перечислите недостатки межсетевого экрана с пакетной фильтрацией.
5. Что еще отслеживает межсетевой экран с фильтрацией пакетов, помимо набора правил, для принятия решения о блокировке или передаче пакета?
6. Выделите два основных типа межсетевых экранов.
7. Какие действия по умолчанию осуществляются межсетевым экраном в отношении трафика?
8. Является ли один из типов межсетевых экранов более безопасным, чем другой?
9. Что межсетевой экран прикладного уровня по умолчанию делает с внутренними адресами?
10. В чем основное различие межсетевого экрана и маршрутизатора?
11. Когда рекомендуется выбирать межсетевой экран с пакетной фильтрацией?
12. Можно ли использовать межсетевые экраны для защиты внутренних сетей от других внутренних систем?
13. При каком условии межсетевой экран прикладного уровня может называться гибридным?
14. Почему порядок правил в наборе правил межсетевого экрана играет важную роль?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Harris, A. Cisco ASA Firewall Fundamentals 2nd Edition / A. Harris. – CCNA, 2010. – 960 с.

ЛАБОРАТОРНАЯ РАБОТА №2
МЕЖСЕТЕВОЙ ЭКРАН ДЛЯ ОС LINUX

Цель работы: ознакомиться с различными типами межсетевых экранов и теорией их построения; изучить работу Iptables: межсетевой экран с открытыми исходными текстами на платформе Linux.

2.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1.1. Iptables: межсетевой экран на платформе Linux

Iptables – утилита для экранирования/фильтрации пакетов, встроенная в большинство систем Linux с ядром версии 2.4 и выше. Данная утилита произошла от более ранних проектов межсетевых экранов в Linux и позволяет создать межсетевой экран с использованием команд операционной системы. Первая система Ipfwadm позволяла создать простой набор правил пропускания или отбрасывания пакетов на основе определенных критериев. В ядро 2.2 ввели Ipchains, чтобы преодолеть ограничения Ipfwadm. Iptables представляет собой обновленный вариант этих программ и допускает многочисленные применения, ставшие привычными для современных межсетевых экранов.

2.1.1.1. Установка Iptables

В большинство систем Linux с ядром 2.4 и выше межсетевой экран Iptables встроен, поэтому никаких дополнительных программ устанавливать не требуется. (Если версия ядра вашей системы меньше 2.4, то в нее встроены Ipchains или Ipfwadm. Это сходные средства, но они не рассматриваются в этой лабораторной работе). Инструкции Iptables можно выполнять из командной строки или из командного файла. Чтобы проверить, установлен ли межсетевой экран Iptables, наберите в командной строке `Iptables -L` и посмотрите, какова реакция. Должен быть выведен текущий набор правил, который, вероятно, пуст, если вы еще не сконфигурировали межсетевой экран.

2.1.1.2. Использование Iptables

Идея, лежащая в основе Iptables и Ipchains, состоит в создании каналов входных данных и их обработке в соответствии с набором правил (конфигурацией вашего межсетевого экрана) с последующей передачей в выходные каналы. В Iptables правила располагаются в таблицах, а внутри таблиц – в цепочках. Основными цепочками, используемыми в Iptables, служат:

– Input;
– Forward;
– Prerouting;
– Postrouting;
– Output.

Общий формат инструкций Iptables таков:

```
Iptables команда спецификация_правил расширения
```

где команда `спецификация_правил` и `расширения` – это одна или несколько допустимых опций.

В табл. 2.1 перечислены основные команды Iptables, а табл. 2.2 содержит краткую спецификацию правил Iptables.

Таблица 2.1

Команды Iptables

Команда	**Описание**
-A цепочка	Добавляет в конец указанной цепочки одно или несколько правил, заданных в инструкции вслед за командой
-I цепочка номер_правила	Вставляет правила в позицию с заданным номером в указанной цепочке. Это полезно, если вы хотите перекрыть правила, заданные ранее
-R цепочка номер_правила	Заменяет правило в позиции с заданным номером в указанной цепочке
-L цепочка	Выдает все правила в цепочке. Если цепочка не задана, выдаются все цепочки
-X цепочка	Удаляет указанную цепочку. По умолчанию удаляются все цепочки
-P цепочка политика	Задает политику для указанной цепочки

Таблица 2.2

Спецификации правил Iptables

Спецификация	**Описание**	
-p протокол	Задает протокол, которому соответствует правило. Допустимыми типами протоколов являются icmp, tcp, udp и all	
-s адрес/маска	Задает определенный адрес или сеть для соответствия. Используется стандартная нотация с косой чертой для указания диапазона IP-адресов	
-j цель	Указывает, что делать с пакетом, если он соответствует спецификациям. Допустимыми опциями для цели являются:	
	DROP	Отбрасывает пакет без всяких дальнейших действий
	REJECT	Отбрасывает пакет и посылает в ответ пакет с уведомлением об ошибке
	LOG	Протоколирует пакет в файле
	MARK	Помечает пакет для дальнейших действий
	TOS	Изменяет поле TOS (тип обслуживания)
	MIRROR	Меняет местами исходный и целевой адреса и посылает пакеты обратно, по сути «отражая» их назад отправителю
-j цель	SNAT	Трансляция исходных сетевых адресов. Эта опция применяется при выполнении трансляции сетевых адресов. Исходный адрес преобразуется в другое статическое значение, определенное с помощью ключа - to-source
	DNAT	Трансляция целевых сетевых адресов. Данная опция аналогична предыдущей, но применяется к целевым адресам
	MASQ	Маскарад с помощью общедоступного IP-адреса
	REDIRECT	Перенаправляет пакет

Существуют и другие команды и опции, но мы перечислили самые распространенные. Весь список команд можно найти в оперативной справке Iptables, набрав man iptables в командной строке.

2.1.2. Создание межсетевого экрана Iptables

Рассмотрим несколько команд, чтобы увидеть, как они используются в практическом приложении. Далее на примере Iptables показано, как создать межсетевой экран. Можно вводить команды интерактивно, по одной, чтобы сразу видеть результаты. Можно также поместить их в командный файл и выполнять его при загрузке системы, поднимая тем самым межсетевой экран. Набирайте их точно так же, как показано.

В следующем примере предполагается, что диапазон IP-адресов 172.16.0.1 – 172.16.0.254 принадлежит вашей подсети ЛВС, к которой подключен интерфейс eth1, и что интерфейс eth0 является соединением с Интернетом.

1. Начните с удаления всех существующих правил с помощью команды Flush:

```
iptables -F FORWARD
```

Это стирает все правила цепочки FORWARD, являющейся основной «воронкой» для всех пакетов, пытающихся пройти через межсетевой экран.

2. Очистите другие цепочки:

```
iptables -F INPUT
iptables -F OUTPUT
```

Эти команды стирают все правила на пути пакетов, направленных в локальную машину, и в выходной цепочке.

3. Поместите стандартную инструкцию «запретить все» в самое начало:

```
iptables -P FORWARD DROP
iptables -A INPUT -i eth0 -j DROP
```

4. Решение о допуске фрагментированных пакетов в Iptables необходимо оформить явным образом:

```
iptables -A FORWARD -f -j ACCEPT
```

5. Существует два вида распространенных атак, которые необходимо сразу заблокировать. Один из них называется подделкой, так как злоумышленник подделывает заголовки IP-пакетов, чтобы казалось, будто внешний пакет имеет внутренний адрес, и таким образом может попасть в вашу сеть, даже если вы используете собственные IP-адреса. Другой тип атаки реализуется отправкой потока пакетов на широковещательный адрес сети, чтобы перегрузить ее. Это называется DoS-атакой. Атаки перечисленных типов можно блокировать с помощью двух простых инструкций:

```
iptables -A FORWARD -s 172.16.0.0/24 -i eth0 -j DROP
iptables -A FORWARD -p icmp -i eth0 -d 172.16.0.255 -j DENY
```

Первая инструкция предписывает отбрасывать все пакеты, приходящие из Интернет-интерфейса eth0 с внутренним адресом 172.16.0.0/24. По определению ни один пакет не должен приходить из недоверенного интерфейса с внутренним, собственным исходным адресом. Вторая инструкция отвергает все приходящие извне на адрес внутренней сети широковещательные пакеты протокола ICMP.

6. Предположим, необходимо принять входящие потоки данных, поступающие по соединениям, инициированным изнутри (например, кто-то просматривает веб-страницу). Пока соединение, инициированное изнутри, поддерживается – все хорошо. Можно, однако, ограничить тип пропускаемого внутрь трафика. Предположим, вы хотите разрешить сотрудникам только веб-доступ и электронную почту. Можно определить типы трафика для прохода внутрь и только для уже инициированного соединения. Следующая инструкция разрешает потоки данных по веб-протоколу HTTP и почтовому протоколу SMTP на основе этого критерия.

```
iptables -A FORWARD -p tcp -i eth0 -d 192.168.0.0/24 --dports
www,smtp --tcp-flags SYN, ACK -j ACCEPT
```

Флаг `-m multiport` извещает Iptables, что вы будете выдавать инструкции сопоставления с портами. Конструкция `-sports` разрешает только трафик электронной почты и веб-навигации. Опция `-syn` разрешает пакеты SYN с неустановленным флагом ACK (и RST), то есть инициирование соединений TCP, а предшествующий восклицательный знак инвертирует смысл этого условия. В результате допускаются только пакеты, не инициирующие соединений.

7. Наконец, вы хотите установить протоколирование, чтобы, просматривая журнал, можно было увидеть, какие пакеты были отброшены. Журнал желательно периодически просматривать, даже если проблем нет, просто чтобы иметь представление о видах отброшенного трафика. Если вы видите повторно отброшенные пакеты из одной и той же сети или одного адреса, то вас, возможно, атаковали. Протоколирование всех видов трафика задается одной инструкцией:

```
iptables -A FORWARD -m tcp -p tcp -j LOG
iptables -A FORWARD -m udp -p udp -j LOG
iptables -A FORWARD -m udp -p icmp -j LOG
```

2.1.3. IP-маскарад с помощью Iptables

Когда создавался Интернет, несколько больших блоков адресов были выделены для использования в собственных сетях. Эти адреса не маршрутизируются в Интернете, их можно использовать, не опасаясь конфликтов с другими сетями. Диапазонами собственных адресов являются

```
10.0.0.0 - 10.255.255.255
192.168.0.0 - 192.168.255.255
172.16.0.0 - 173.31.255.255
```

Используя эти адреса в своей внутренней сети и имея один внешний маршрутизируемый IP-адрес для межсетевого экрана, вы эффективно закроете внутренние машины от внешнего доступа. С помощью Iptables несложно выстроить дополнительный защитный рубеж, применяя IP-маскарад. Межсетевой экран отрезает внутренний IP-заголовок и заменяет его заголовком, задающим экран в качестве отправителя. Затем пакет данных посылают в место назначения с исходящим IP-адресом общедоступного интерфейса межсетевого экрана. Когда пакет возвращается, экран вспоминает, по какому внутреннему IP-адресу тот направлен, и переадресует его для внутренней доставки. Этот процесс называется также трансляцией сетевых адресов (NAT). В Iptables трансляцию адресов можно организовать с помощью следующих инструкций:

```
iptables -t nat -P POSTROUTING DROP
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

2.1.4. Turtle Firewall

Если вы хотите быстро построить межсетевой экран, не набирая все эти инструкции Iptables и не запоминая их синтаксис, то существует инструмент, который создает экранирующие инструкции с помощью графического интерфейса, избавляя вас от технической работы.

По сути Turtle является набором командных файлов Perl, которые делают за вас всю черновую работу по подготовке межсетевого экрана Iptables к работе. Эта программа существенно облегчает просмотр правил и проверку того, что инструкции поступают в правильном порядке. Она выполняется как служба, поэтому вам не нужно заботиться об инициализации межсетевого экрана с помощью командного файла.

2.1.5. Установка Turtle Firewall

Установка и начальная настройка Turtle Firewall очень проста, так как используется модуль администрирования Webmin, доступный на большинстве платформ Linux.

1. Используя веб-навигатор, войдите на сервер Webmin, указав его IP-адрес или имя хоста. Отобразится интерфейс Webmin.

2. Щелкните кнопкой мыши на вкладке Module Index, и в окне отобразится основной экран Turtle Firewall (рис. 2.1).

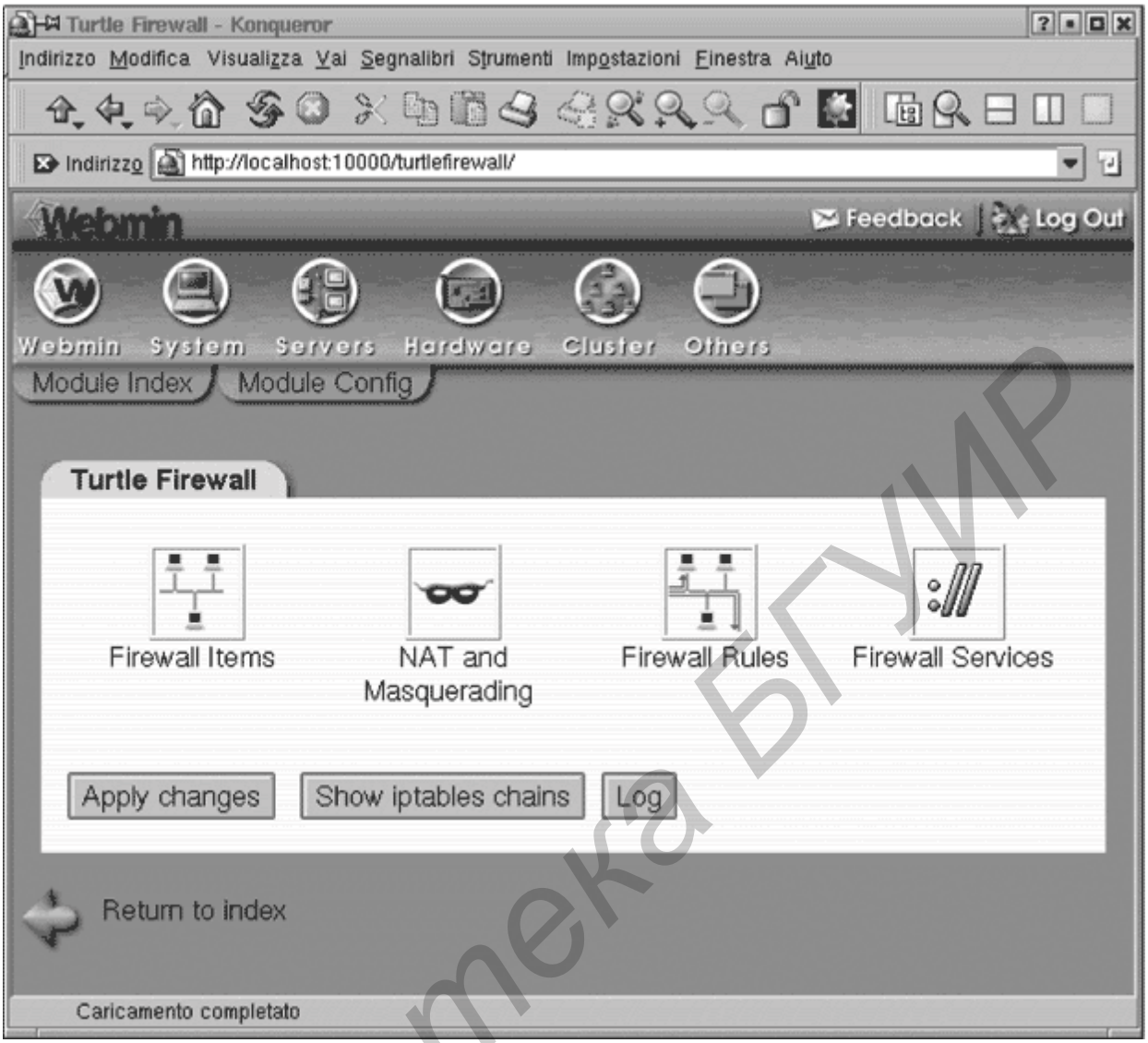

Рис. 2.1. Основной экран Turtle Firewall

3. Щелкните кнопкой мыши на иконке Firewall Items, чтобы начать конфигурирование межсетевого экрана. Сначала вам придется задать основные сведения о нем (рис. 2.2). В Turtle Firewall применяется концепция зон для определения доверенных и недоверенных сетей. Доверенная зона связывается с сетью, где работают люди, которым принято доверять (пример – ваша внутренняя сеть). Недоверенная зона – это сеть, в которой может работать кто угодно: от сотрудников до заказчиков, продавцов или даже злоумышленников. В Turtle они называются «good» и «bad», но это по сути то же самое, что доверенная и недоверенная.

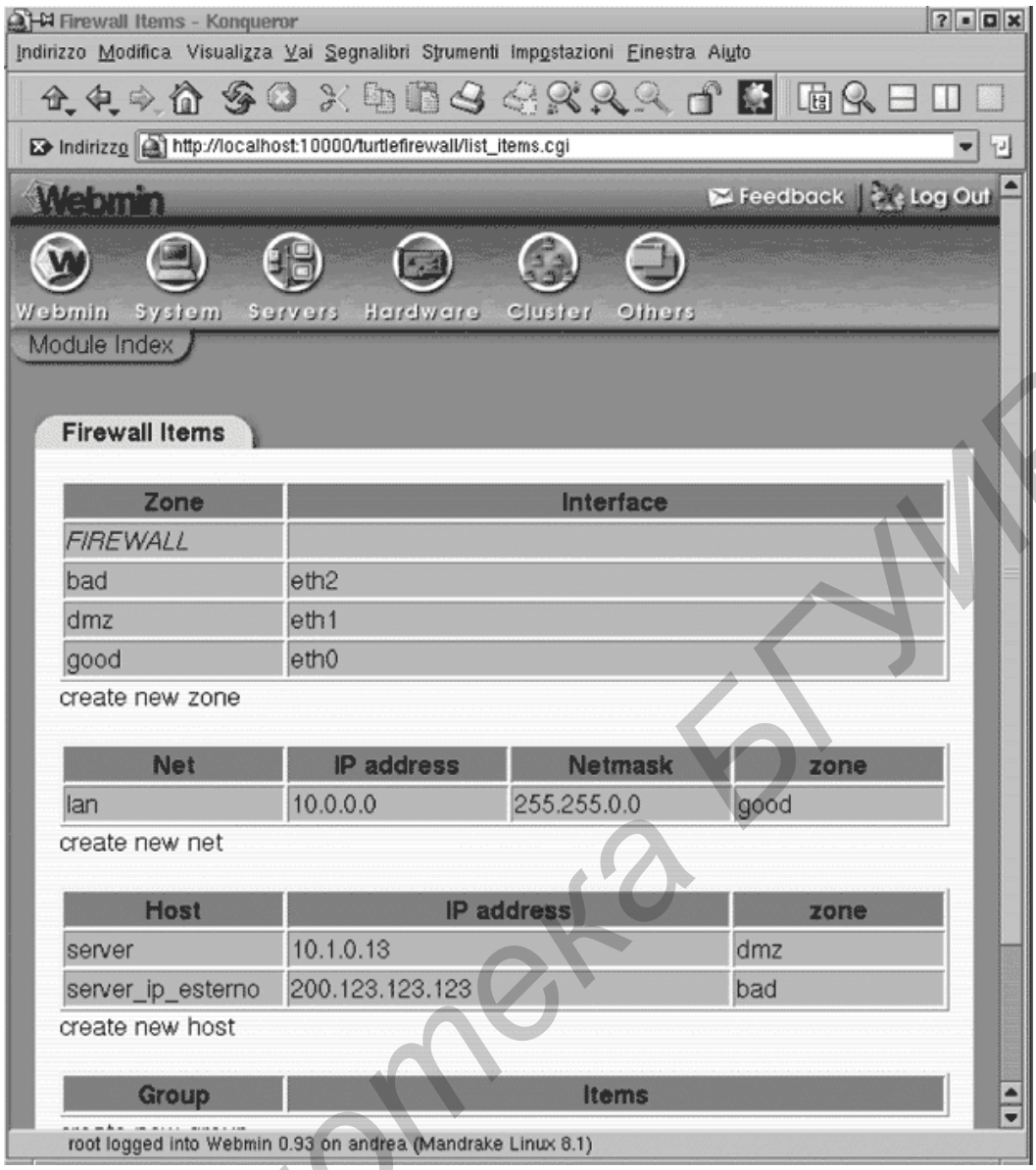

Рис. 2.2. Конфигурирование Turtle Firewall

В Turtle предусмотрен также элемент для демилитаризованной зоны (dmz), в которой располагают серверы со свободным доступом к недоверенной зоне. Задайте интерфейсы для доверенной, недоверенной и демилитаризованной (если таковая имеется) зон.

4. Затем в блоке Net следует задать адреса внутренней сети. Укажите диапазон IP-адресов с маской подсети для внутренней сети, которая будет защищаться межсетевым экраном, в предоставленном поле (см. рис. 2.2).

5. После этого задайте все хосты во внутренней сети и демилитаризованной зоне, требующие специального рассмотрения (пример – почтовый или веб-сервер). Сделайте это в блоке Hosts (см. рис. 2.2).

6. Наконец, в области Group можно определить все специальные хосты, к которым желательно подходить особым образом (пример – машины администраторов). Теперь ваш межсетевой экран готов к работе в базовом режиме.

Вероятно, вы захотите добавить некоторые дополнительные ограничения или разрешения, например, возможность кому-то извне использовать для входа SSH. Это можно сделать, написав правило под вкладкой Firewall Rules. Щелкните на ней кнопкой мыши, и с вами начнут графический диалог для написания нового правила. Вы обнаружите сходство структуры диалога с форматами инструкций Iptables (рис. 2.3).

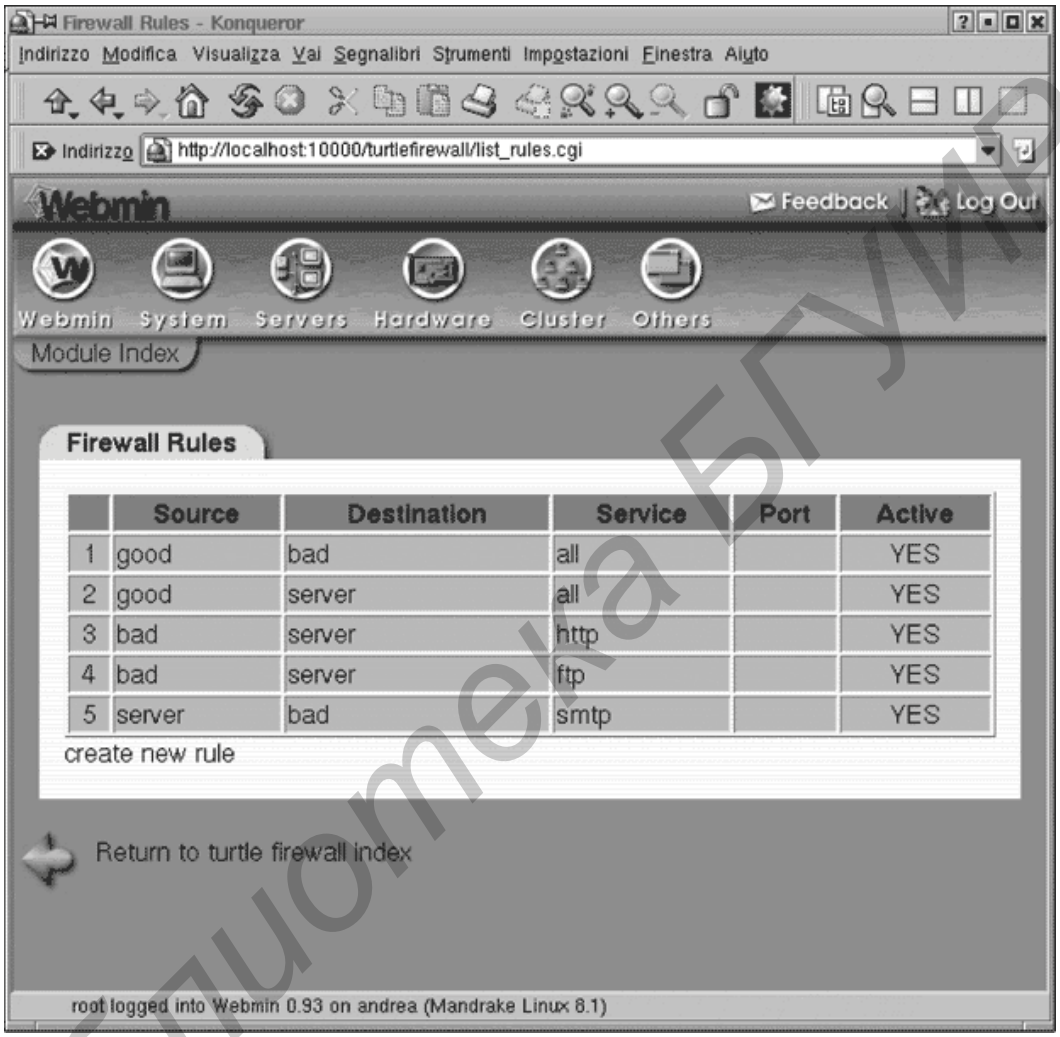

Рис. 2.3. Правила Turtle Firewall

Если вы хотите реализовать функцию маскарада Iptables, используя собственные IP-адреса для вашей внутренней сети, щелкните кнопкой мыши на иконке «NAT and Masquerading» на основном экране. Вы сможете специфицировать, какая зона будет подвергаться маскараду (рис. 2.4). Обычно это доверенный интерфейс. Здесь же можно задать хосты, сетевые адреса которых будут транслироваться. Хост, заданный в качестве виртуального, будет служить фасадом реального хоста, и межсетевой экран будет переправлять все пакеты реальному хосту через виртуальный. Это создает дополнительный защитный рубеж для внутренних серверов.

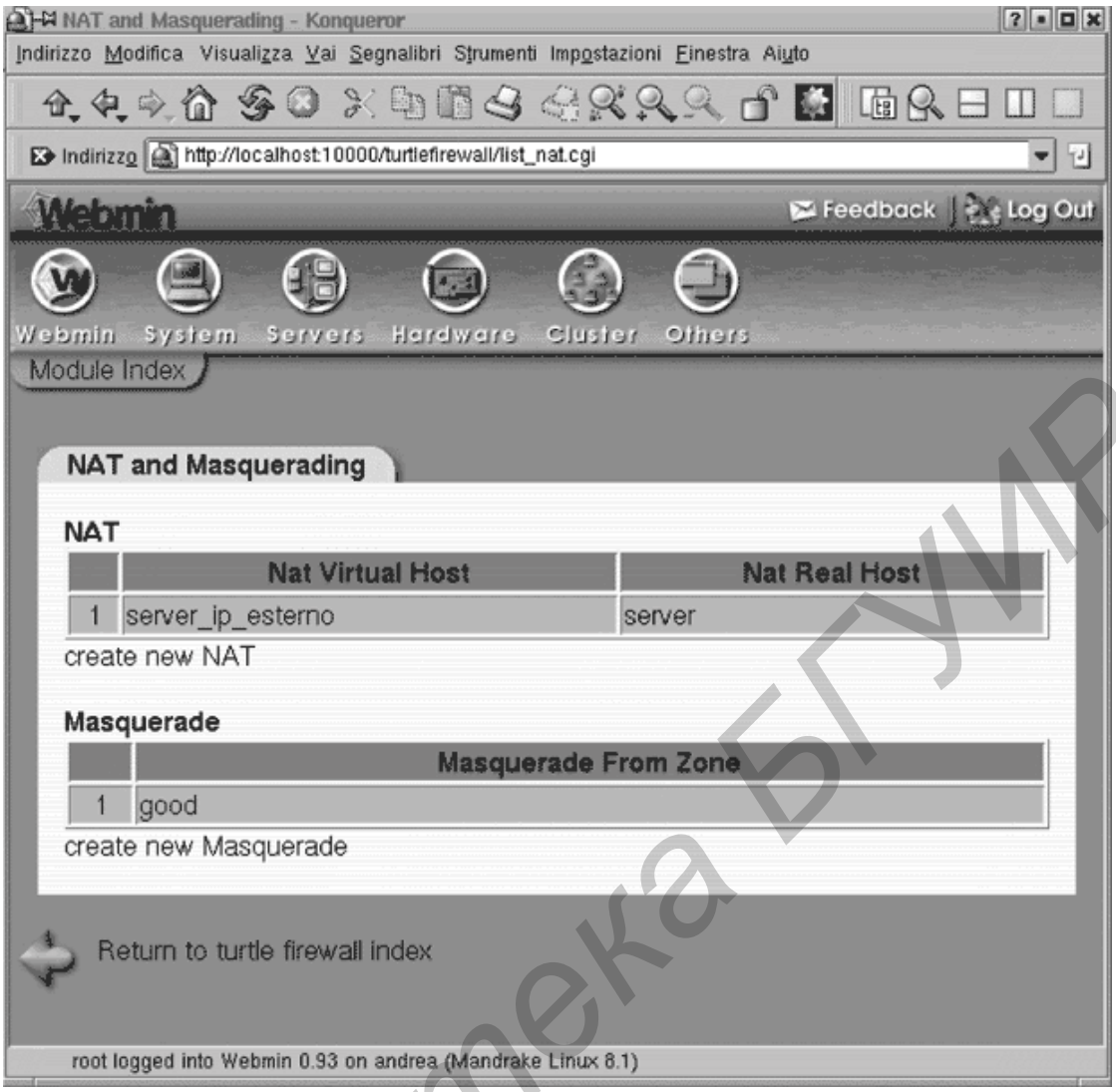

Рис. 2.4. Трансляция сетевых адресов и маскарад в Turtle Firewall

2.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Создать межсетевой экран, защищающий от наиболее распространенных атак из Интернета.

2. Создать IP-маскарад с помощью Iptables.

3. Создать и отконфигурировать межсетевой экран, защищающий от наиболее распространенных атак из Интернета с помощью Turtle Firewall.

2.3. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Реализация решения задачи.
5. Выводы.

2.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое межсетевой экран?
2. Нужно ли разрабатывать тактику использования сети? Если да, то для чего?
3. Нужно ли составлять карту входящих и исходящих сервисов? Если да, то для чего?
4. Нужно ли составлять таблицу сопоставлений «IP – имя компьютера» для сети, где реализован DHCP? Если да, то зачем?
5. Какие методы настройки межсетевых экранов вы знаете?
6. В какой ситуации лучше всего применять метод «разрешить все», а затем задавать поведения, которые требуют блокировки?
7. Как реализован метод «запретить все»?
8. Что такое Iptables?
9. Какую информацию можно получить благодаря команде `iptables -L`?
10. Как расположены, правила внутри Iptables?
11. Что такое Turtle Firewall?
12. Обязательно ли нужен браузер для корректной работы Iptables?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Andreasson, O. Iptables Tutorial 1.1.19 / O. Andreasson. – CSVA, 2009. – 240 с.

ЛАБОРАТОРНАЯ РАБОТА №3
СКАНЕРЫ ПОРТОВ. Nmap

Цель работы: ознакомиться с различными типами сканеров портов и разобраться в принципах их работы; изучить основные методы сканирования для достижения максимальной эффективности; изучить работу Nmap – свободной утилиты, предназначенной для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов и определения состояния объектов сканируемой сети (портов и соответствующих им служб).

3.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Малоизвестная, но важная организация **Internet Assigned Numbers Authority (IANA)** присваивает номера портов TCP/UDP. Она отслеживает множество различных стандартов и систем, обеспечивающих функционирование сети Интернет. Среди ее обязанностей – распределение IP-адресов и назначение ответственных за имена доменов верхнего уровня. IANA обладает значительной властью, хотя по большей части остается в тени. Немногие люди за пределами инженерных подразделений коммуникационных компаний знают о существовании IANA, но она управляет значительной частью «недвижимости» Интернета. IANA отвечает также за поддержание списка сетевых портов, по которым можно подключаться к определенным сервисам, предполагая, что приложение или операционная система соответствуют этим стандартам. Разумеется, всем компаниям, производящим программное обеспечение, надлежит скрупулезно следовать этим стандартам, иначе их продукты могут оказаться несовместимыми с другими подключенными к Интернету системами. В табл. 3.1 перечислены некоторые из наиболее употребительных TCP-портов для серверных приложений.

Полный список номеров портов представлен на веб-сайте IANA (http://www.iana.org). Номер порта присвоен почти каждому значительному приложению. Как для TCP-, так и для UDP-сервисов эти номера лежат в диапазоне от 1 до 65 535. Номера портов от 0 до 1023 считаются зарезервированными для общеупотребительных приложений, обычно выполняющихся от имени пользователя root (administrator) или другого привилегированного пользователя. Соответствующие им номера портов называются общеизвестными. Номера портов с 1024 по 65 535 можно регистрировать в IANA для конкретных приложений. Они обычно соответствуют определенным сервисам, но подобная регистрация не имеет для производителей столь же обязательной силы, как в случае зарезервированных номеров.

Наконец, существуют недолговечные номера портов, которые операционная система выбирает случайным образом из номеров, превышающих 1024 (обычно – в верхней части диапазона). Они используются для машин, которые

произвольным образом устанавливают соединения с другими машинами. Например, для загрузки веб-страницы ваша машина обратится к порту 80 веб-сервера. Сервер увидит входящее соединение с некоторым случайным номером порта, превышающим 1024. В таком случае сервер будет знать, что это, вероятно, пользователь, а не другое приложение, устанавливающее с ним соединение. Он также использует недолговечный номер порта для отслеживания определенного пользователя и сеанса. Например, если вы параллельно откроете два навигатора, то ваш компьютер для сеанса каждого из них создаст два разных номера порта для установления соединений, которые сервер будет считать различными.

Таблица 3.1

Общеупотребительные серверные порты

Номер порта	**Протокол**	**Сервис**
21	FTP	Протокол передачи файлов (управляющий порт)
22	SSH	Защищенный shell
23	Telnet	Telnet
25	SMTP	Почтовый сервис
53	DNS	Разрешение доменных имен
79	Finger	Finger
80	HTTP	Веб-сервис
135-139	NetBIOS	Сетевые коммуникации Windows
443	SSL	Защищенный веб-сервис

То, что пакет помечен для порта 80, не запрещает ему содержать данные, отличные от веб-трафика. Система номеров портов зависит от определенной «честности» машин, с которыми приходится взаимодействовать, и именно это представляет опасность. На самом деле многие приложения, такие как программы мгновенного обмена сообщениями и одно ранговое ПО, которые обычно блокируются межсетевым экраном организации, нарушают эту конвенцию и проникают через порт 80, который согласно конфигурации остается открытым, поскольку пользователям, находящимся позади межсетевого экрана, разрешен веб-доступ.

Когда порт на компьютере открыт, он получает весь направляемый в него трафик, законный или незаконный. Посылая некорректно сформированные пакеты, пакеты со слишком большим количеством данных или с некорректно отформатированными данными, иногда можно вызвать аварийное завершение основного приложения, перенаправить поток управления в этом приложении и незаконно получить доступ к машине. Это называется переполнением буфера и составляет большой процент современных уязвимостей.

Переполнение буфера происходит, если прикладные программисты неаккуратно пишут программы и не обеспечивают должную обработку данных, переполняющих области памяти, отведенные входным переменным. Когда в программу поступают входные данные, не умещающиеся в отведенный буфер, они могут изменить внутренний ход выполнения программы и в результате предоставить хакеру доступ к ресурсам системного уровня. Почти все программы

независимо от размера содержат ошибки такого рода. Современное программное обеспечение насчитывает миллионы строк исходных текстов, в результате вероятность наличия ошибок очень велика. Возможно, со временем, когда вырастут новые поколения программистов, обученных автоматически писать безопасный код, данная проблема исчезнет. Пока же необходимо внимательно следить за тем, какие приложения или порты видны в вашей сети. Эти порты являются **потенциальными «окнами»** в серверах и рабочих станциях, через которые хакеры могут запускать свой вредоносный код в ваш компьютер. Поскольку именно здесь бывает большинство нарушений безопасности, очень важно понимать, что происходит на этом уровне на ваших серверах и в других машинах. Этого можно легко добиться с помощью программного средства, называемого сканером портов.

3.1.1. Обзор сканеров портов

Сканеры портов опрашивают набор портов TCP или UDP и смотрят, не ответит ли приложение. Если ответ получен, это означает, что некоторое приложение слушает порт с данным номером. Имеется 65 535 возможных портов TCP и столько же – UDP. Сканеры можно сконфигурировать для опроса всех возможных портов или только общеупотребительных (с номерами, меньшими 1024). Веская причина для полного сканирования состоит в том, что сетевые троянские и другие вредоносные программы, чтобы избежать обнаружения, нередко используют нетрадиционные порты с номерами в верхней части диапазона. Кроме того, некоторые производители не следуют стандартам должным образом и подключают серверные приложения к портам с большими номерами. Полное сканирование охватывает все возможные места, где могут скрываться приложения, хотя и требует больше времени и использует несколько большую часть полосы пропускания.

Сканеры портов предстают во множестве видов – от очень сложных с множеством различных возможностей до имеющих минимальную функциональность. На самом деле вы сами можете вручную выполнить функции сканера портов, применяя Telnet и проверяя порты по очереди. Просто подключайтесь к IP-адресу, добавляя номер порта, например: `telnet 192.168.0.1:80`.

Данная команда использует Telnet для соединения с машиной. Номер после двоеточия (для некоторых реализаций Telnet необходимо просто оставить пробел между IP-адресом и номером порта) говорит Telnet, что для соединения надо использовать порт 80 вместо стандартного для Telnet порта 23. Вместо того чтобы получить от Telnet обычное приглашение, которое выдается при подключении к его подразумеваемому порту, вы соединитесь с веб-сервером, если таковой запущен на машине. После нажатия клавиши ввода вы получите первый ответ веб-сервера навигатору. Вы увидите информацию из заголовка HTTP, которая обычно обрабатывается навигатором и скрыта от пользователя. Она выглядит примерно так:

```
    GET / HTTP
    HTTP/1.1 400 Bad Request
    Date: Mon, 15 Mar 2004 17:13:16 GMT
    Server: Apache/1.3.20 Sun Cobalt (Unix) Chili!Soft-ASP/3.6.2
    mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.1.2
mod_auth_pam_external/0.1
    FrontPage/4.0.4.3 mod_perl/1.25
    Connection: close
    Content-Type: text/html; charset=iso-8859-1
```

Так же можно поступить с любым открытым портом, но вы не всегда получите четкий ответ. По сути именно это и делают сканеры портов: они пытаются установить соединение и ожидают ответ.

Некоторые сканеры портов пытаются также идентифицировать операционную систему на другом конце, выявляя так называемые идентификационные метки TCP. Хотя TCP/IP является стандартом сетевых коммуникаций, каждый производитель реализует его немного иначе, чем другие. Эти различия, обычно не мешающие взаимодействию, проявляются в ответах на любое воздействие, такое как эхо-тест или попытка установления TCP-соединения. Например, цифровая подпись ответа на эхо-тест от системы Windows выглядит иначе, чем в ответе системы Linux. Имеются даже различия между версиями операционной системы. Ниже приведен пример идентификационных меток TCP для Windows ME, 2000 и XP.

```
# Windows Millennium Edition v4.90.300
# Windows 2000 Professional (x86)
# Windows Me or Windows 2000 RC1 through final release
# Microsoft Windows 2000 Advanced Server
# Fingeprint Windows XP Pro with all current updates to May
2002
Fingeprint Windows Millenium Edition (Me), Win 2000, or WinXP
Tseq(Class=RI%gcd=<6%SI=<23726&>49C%IPID=I%TS=0)
T1(DF=Y%W=5B4|14F0|16D0|2EE0|402E|B5C9|B580|C000|D304|FC00|FD
20|FD68|FFFF%ACK=S++%Flags=AS%Ops=NNT|MNWNNT)
T2(Resp=Y|N%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=5B4|14F0|16D0|2EE0|B5C9|B580|C000|402E|D304|
FC00|FD20|FD68|FFFF%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(DF=N%W=0%ACK=0%Flags=R%Ops=)
```

Наборы букв и знаков в нижней части листинга являются уникальными установками, используемыми Windows при установлении TCP-соединений. Сравнивая полученный от машины ответ с базой известных идентификационных меток TCP, можно сделать разумное предположение об операционной системе на другом конце.

Если вашу сеть сканируют злоумышленники, это предоставляет им ценную информацию. Знание операционной системы и ее версии может послужить хорошей отправной точкой для определения того, какие зацепки и средства

проникновения стоит испробовать. Это очень веская причина для регулярного сканирования своей сети, чтобы определить, какие порты в системе оставлены открытыми. Затем следует их просмотреть, закрыть неиспользуемые порты и защитить те, которые должны оставаться открытыми.

3.1.2. Планирование сканирования портов

При планировании сканирования портов в любой сети помните, что эта деятельность создает большую нагрузку на сеть. Сканирование десятков тысяч портов за короткое время порождает в сети интенсивный трафик. Если вы используете для сканирования устаревшей сети на 10 Мбит/с мощный компьютер, это может существенно повлиять на сетевую производительность. При сканировании через Интернет данная проблема будет менее острой, так как ограничивающим фактором послужит пропускная способность промежуточных соединений, однако все равно можно снизить производительность загруженного веб-сервера или почтового сервера. В крайних случаях большой объем информации может даже привести к прекращению работы машин.

Независимо от способа использования описанных выше средств обязательно получите разрешение владельца сканируемых хостов. Сканирование портов – деятельность на грани законности. В действительности вы не взламываете системы, а просто опрашиваете сеть, однако это может привести к нарушению работы корпоративной сети. И прежде чем вы решите просканировать несколько любимых веб-серверов, проверьте, нет ли в контракте на предоставление интернет-услуг пунктов, запрещающих подобную деятельность. Даже при наличии разрешения необходимо принять во внимание предполагаемый эффект сканирования целевой сети. Если это интенсивно используемая сеть, вы должны выполнять сканирование ночью или в периоды наименьшей активности. Некоторые сканеры имеют возможность замедлять посылку пакетов, чтобы не очень сильно воздействовать на сеть. Это означает, что сканирование будет выполняться дольше, но в более дружественном для сети режиме.

3.1.3. Применение сканеров портов

Когда вы получите разрешение на сканирование, следует определить, с какой целью вы собираетесь сканировать сеть.

3.1.3.1. Инвентаризация сети

Если вы не знаете точно, сколько машин у вас работает или хотите узнать IP-адреса всех ваших серверов, сканеры портов предлагают быстрый способ просмотра диапазона адресов и выявления всех активных машин в этом сегменте. Можно даже воспользоваться средством Nlog для корректного занесения их в базу данных.

3.1.3.2. Оптимизация сети/сервера

Сканер портов покажет все сервисы, запущенные в данный момент на машине. Если это серверная машина, то, вероятно, таковых окажется много, и, возможно, не все из них на самом деле нужны для выполнения основной функ-

ции машины. Помните: чем больше сервисов, тем меньше безопасности. И все эти программы могут замедлять работу перегруженного сервера.

3.1.3.3. Выявление шпионского ПО, троянских программ и «сетевых червей»

Активные веб-серверы нередко подцепляют на веб-сайтах небольшие программы, которые пытаются отслеживать их поведение или выдавать на их компьютеры специальную всплывающую рекламу. Эти программы называются шпионским ПО, потому что нередко они пытаются следить за активностью пользователя и могут передавать собранные данные обратно на центральный сервер. Эти программы обычно не опасны, но их чрезмерное количество может существенно снизить производительность труда пользователя. Кроме того, написаны они зачастую неаккуратно и могут мешать работе других программ или даже вызывать их аварийное завершение. Они могут также помогать хакерам в поиске уязвимостей.

Другим классом сетевого программного обеспечения, который следует исключить в своей сети, являются троянские программы, созданные специально для взлома сетей, подобно троянскому коню из греческой мифологии. Обычно их присутствие можно обнаружить только по открытому сетевому порту, а с помощью антивирусных средств выявить их крайне сложно. Оказавшись внутри компьютера, большинство троянских программ пытаются вступить во внешние коммуникации, чтобы дать своему создателю или отправителю знать, что они заразили машину на этих портах. В табл. 3.2 перечислены наиболее распространенные троянские программы и их номера портов. Многие номера портов легко распознаваемы по определенному набору цифр (например, для NetBus это 54321, а для Back Orifice – 31337, что в хакерской кодировке читается как «элита»). В целом же троянские программы стремятся использовать порты с большими, необычными, нераспознаваемыми номерами.

«Сетевые черви» – особо опасный тип вирусов. Зачастую они снабжены сетевыми средствами и открывают порты на компьютере-«хозяине». «Сетевые черви» используют сеть для распространения и поэтому иногда выявляются при сканировании портов, которое может стать помощником в защите от этого вида вирусов.

Таблица 3.2

Порты, используемые наиболее распространенными троянскими программами

Номер порта	IP-протокол	Известные троянские программы, использующие эти порты
12456 и 54321	TCP	NetBus
31335	TCP	Trin00
31337	TCP	Back Orifice
31785-31791	TCP	Hack 'a'Tack
54321	UDP	Back Orifice 2000
60000	TCP	Deep Throat
65000	TCP	Stacheldraht

3.1.4. Сканер портов Nmap

Nmap – вне всяких сомнений, лучший сканер портов. На Nmap опирается сканер уязвимостей Nessus, который будет изучен в лабораторной работе №6. Доступно также несколько дополнений, включая программу Nlog. Достаточно сказать, что Nmap должен входить в инструментарий каждого администратора безопасности. Перечислим некоторые из основных достоинств Nmap:

– у него есть множество опций. Можно понизить частоту отправки зондирующих пакетов или, наоборот, повысить ее. Далее Nmap выходит за рамки простого сканирования портов и осуществляет идентификацию ОС, что полезно при установлении соответствия между IP-адресами и машинами;

– он легкий, но мощный: код Nmap невелик и будет выполняться даже на самых старых машинах;

– он прост в использовании, хотя существует множество различных способов его запуска. Реализуемое по умолчанию базовое сканирование SYN делает все, что требуется большинству приложений. Имеется как режим командной строки, так и графический интерфейс для UNIX и Windows.

3.1.5. Сканирование сетей с помощью Nmap

Графический клиент Nmap предоставляет весьма простой интерфейс (рис. 3.1). Вверху имеется поле для ввода IP-адреса или диапазона IP-адресов, а чтобы начать сканирование, достаточно нажать кнопку Scan.

Важно знать сетевые маски и нотации с косой чертой.

Вам будут часто встречаться обозначения IP-сетей или с сетевой маской, или с косой чертой и числом после нее. Это два способа задать размер сети. Для их понимания необходимо представлять себе структуру IP-адреса. Стандартный адрес IPv4 состоит из 32 бит. Его обычно представляют в виде четырех частей – восьмибитных октетов. Октеты для удобочитаемости обычно преобразуют в десятичные числа. Если вы видите 192.168.1.1, то компьютер видит

11000000 10101000 00000001 00000001

Маска сети обычно представляет собой набор из четырех чисел. Она показывает, где кончается локальная сеть и начинается глобальная. Обычно маска выглядит примерно так:

255.255.255.0

Чтобы определить размер сети, представленной сетевой маской, достаточно вычесть каждый октет из 256 и перемножить полученные разности. Например, сетевая маска 255.255.255.248 описывает восьмиэлементную IP-сеть, поскольку:

(256–255)*(256–255)*(256–255)*(256-248) = 8.

Сетевая маска 255.255.255.0 представляет IP-сеть из 256 узлов, так как

(256–255)*(256–255)*(256–255)*(256–0) = 256.

Наконец, сетевая маска 255.255.0.0 описывает сеть из 65536 IP-адресов, так как

(256–255)*(256–255)*(256–0)*(256-0) = 65536.

Нотация с косой чертой чуть сложнее для понимания, но идея остается прежней. Число после косой черты показывает, сколько бит описывают глобальную сеть. Вычитая это число из 32, получаем число бит, описывающих локальную сеть. Например, запись 192.168.0.0/24 представляет сеть, начинающуюся с 192.168.0.0 и насчитывающую 256 IP-адресов. Это такой же размер, как и у рассмотренной выше сети с маской 255.255.255.0.

32 бита IP-адреса минус 24 бита для префикса сети дает 8 бит для локального использования, то есть 256 возможных адресов. Для запоминания двоичных чисел воспользуйтесь табл. 3.3.

В табл. 3.4 приведены различные форматы для ввода IP-адресов. Адреса могут также извлекаться из файла, если выбрать пункт Input элемента File основного меню и задать текстовый файл с данными в подходящем для Nmap формате.

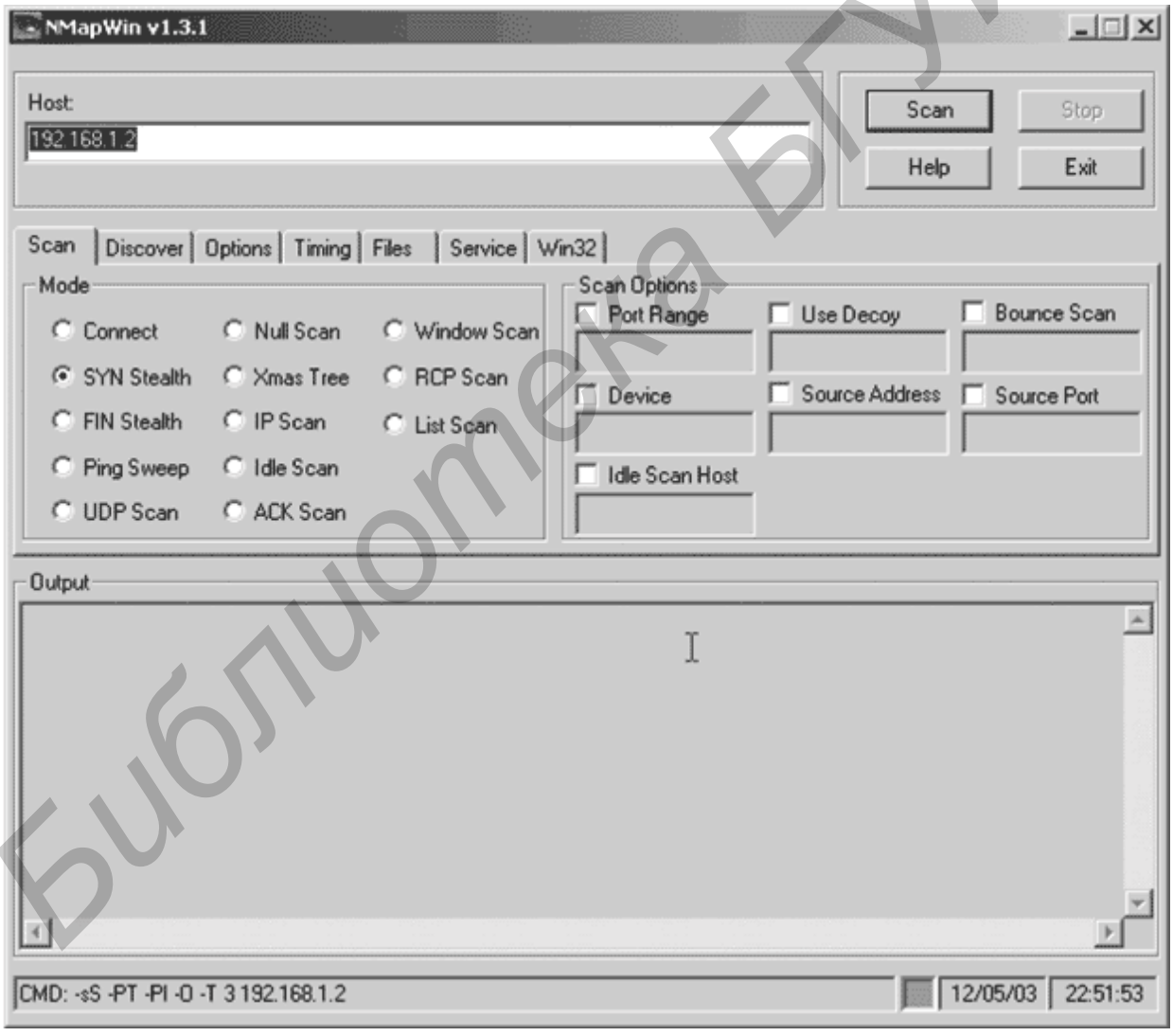

Рис. 3.1. Образ экрана NmapWin

Таблица 3.3

Соответствие нотации и максимально возможного количества сетевых устройств

Нотация с косой чертой	Размер сети
/24	256 IP-адресов
/25	128 IP-адресов
/26	64 IP-адреса
/27	32 IP-адреса
/28	16 IP-адресов
/29	8 IP-адресов
/30	4 IP-адреса
/31	2 IP-адреса
/32	1 IP-адрес

Таблица 3.4

Форматы IP-адресов

Формат	Пример
Одиночный IP-адрес	192.168.0.1
IP- адреса, разделенные запятыми	192.168.0.1,192.168.0.2
IP-диапазон, разделенный дефисом	192.168.0.1-255
Использование стандартной нотации с косой чертой	192.168.0.1/24 (сеть класса С из 256 адресов)

3.1.6. Запуск Nmap из командной строки

Nmap можно запустить из командной строки как в UNIX, так и в Windows. Общий формат таков: `nmap параметры IP-диапазон`.

3.1.7. Типы сканирования в Nmap

Nmap поддерживает множество различных типов сканирования. В табл. 3.5 перечислены наиболее употребительные. Указаны также параметры командной строки, если вы захотите использовать этот интерфейс.

Таблица 3.5

Типы сканирования в Nmap и параметры командной строки

Тип сканирования (параметры командной строки)	Описание
SYN (-sS)	Подразумеваемый тип сканирования, пригодный для большинства целей. Он менее заметен, чем TCP Connect, то есть не будет фиксироваться большинством простых средств протоколирования. В этом режиме в каждый возможный порт посылаются одиночные TCP-пакеты с установленным флагом SYN. Если в ответ возвращается пакет SYN ACK, то Nmap делает вывод, что здесь запущен сервис. Если ответа нет, то предполагается, что порт закрыт. SYN-сканирование не завершает трехходовое квитирование установления связи в TCP, так как не возвращает целевой машине пакет с установленным флагом ACK; с точки зрения

Окончание табл. 3.5

Тип сканирования (параметры командной строки)	Описание
	сканируемой системы действующие соединения не устанавливаются. Однако удаленная система будет удерживать эту «половинку сокета» открытой, пока не пройдет максимально допустимое время ответа. Некоторые современные серверы и программы выявления вторжений достаточно интеллектуальны, чтобы уловить подобные действия, но для большинства машин SYN-сканирование будет невидимым
TCP-соединение: Connect (-sT)	Этот тип сканирования напоминает SYN, за исключением того, что трехходовое квитирование установления связи в TCP выполняется до конца и устанавливается полноценное соединение. Подобное сканирование не только шумно, но и создает дополнительную нагрузку на сканируемые машины и сеть. Однако, если скрытность или экономия полосы пропускания не являются приоритетными, то сканированием Connect по сравнению с SYN можно порой получить более точные результаты. Кроме того, если у вас нет привилегий администратора или суперпользователя на машине Nmap, вы не сможете воспользоваться никаким другим типом сканирования
Эхо-тестирование: Ping Sweep (-sP)	Выполняется простое эхо-тестирование всех адресов, чтобы увидеть, какие из них ответят на ICMP-запрос. Если вас на самом деле не интересует, какие сервисы запущены, и вы просто хотите знать, какие IP-адреса активны, то данный тип позволит достичь цели много быстрее, чем полное сканирование портов. Однако некоторые машины могут быть сконфигурированы так, чтобы не отвечать на ping (например, новый межсетевой экран XP), но тем не менее выполнять некоторые сервисы, поэтому Ping Sweep – менее надежный метод, чем полное сканирование портов

3.1.8. Опции раскрытия для Nmap

Можно настроить способ, которым Nmap выполняет раскрытие сетей и определяет, какие хосты работают. В табл. 3.6 перечислено несколько различных вариантов.

3.1.8.1. Опции времени для Nmap

Nmap предоставляет средства для повышения или понижения частоты, с которой посылаются пакеты сканирования. Чтобы уменьшить расход сетевого трафика, можно понизить частоту. Помните только, что чем реже посылаются пакеты, тем дольше продлится сканирование. Для больших сетей время может вырасти экспоненциально. С другой стороны, если расход дополнительного сетевого трафика неважен, а важна скорость, можно поднять частоту. Различные уровни и частоты пакетов приведены в табл. 3.7. В версии для Windows или с помощью опций командной строки можно устанавливать специальные частоты.

Таблица 3.6

Опции раскрытия для Nmap

Опция	Описание
TCP + ICMP (-PB)	Подразумеваемая настройка. Nmap обычно использует для определения статуса хоста и ICMP-, и TCP-пакеты. Это наиболее надежный и точный способ, так как если хост активен, то хотя бы по одному методу ответ, как правило, будет получен. К сожалению, это также самый шумный способ, который скорее всего приведет к регистрации каким-нибудь устройством сканируемой сети
Эхо-тестирование TCP (-PT)	Для обнаружения хостов используется только метод TCP. Многие межсетевые экраны и некоторые маршрутизаторы отбрасывают пакеты ICMP, возможно, с протоколированием. Если вы пытаетесь остаться невидимым, то метод TCP – это наилучший вариант. Однако для некоторых экзотических типов сканирования (FIN, XMAS, NULL) какие-то хосты могут остаться незамеченными
Без эхо-тестирования (-P0)	Если задается эта опция, то Nmap не будет пытаться сначала выяснить, какие хосты активны, а будет вместо этого посылать пакеты по каждому IP-адресу заданного диапазона, даже если по этому адресу машины нет. Это расточительно как с точки зрения полосы пропускания, так и времени, особенно когда сканируются большие диапазоны. Однако это может быть единственным способом просканировать хорошо защищенную сеть, которая не отвечает на ICMP-пакеты

Таблица 3.7

Параметры Nmap для управления частотой посылки пакетов

Уровень частоты	Параметр командной строки	Частота пакетов	Пояснения
Параноидальный	-F 0	1 раз в 5 минут	Не используйте эту опцию при сканировании большого числа хостов, иначе сканирование никогда не закончится
Исподтишка	-F 1	1 раз в 15 секунд	-
Вежливый	-F 2	1 раз в 4 секунды	-
Нормальный	-F 3	Со скоростью работы ОС	Используется по умолчанию

3.1.8.2. Другие опции Nmap

В табл. 3.8 перечислены некоторые другие опции Nmap, которые управляют, например, разрешением доменных имен, идентификацией ОС и т. д., и не попадают в другие категории.

Таблица 3.8

Прочие опции Nmap

Опция	Описание
Не выполнять разрешение имен (-n)	Обычно Nmap пытается разрешать доменные имена для всех сканируемых IP-адресов. Это может существенно затягивать сканирование, поэтому если вас не интересуют имена хостов, разрешение имен можно отключить. Помните, однако, что знать имена хостов полезно, особенно при сканировании сетей с DHCP, где IP-адреса могут меняться

Окончание табл. 3.8

Опция	Описание
Быстрое сканирование (-F)	Эта опция вызывает сканирование только портов, перечисленных в файлах употребительных портов Nmap. По умолчанию это общеупотребительные серверные порты с номерами, меньшими 1024. Данные файлы можно отредактировать и добавить в список другие порты. Подобное сканирование может оказаться значительно более быстрым, но оно не выявит троянские программы и сервисы, использующие порты с большими номерами
Диапазон портов (-p диапазон_портов)	По умолчанию Nmap сканирует все 65 535 возможных портов TCP. Однако, если вы хотите просканировать только определенный диапазон, можно задать его в качестве аргумента опции -p. Это полезно, если вы хотите просканировать только один тип серверов, например, порт 80 для веб-серверов, или только верхние диапазоны, чтобы найти необычные сервисы и потенциальные троянские программы
Идентификация ОС (-O)	Подразумеваемая опция. Как упоминалось ранее, каждая реализация стека TCP имеет свои особенности. При сравнении точной идентификационной метки ответов с базой данных известных идентификационных меток TCP, Nmap, как правило, может с высокой достоверностью (иногда вплоть до диапазона версий) идентифицировать ОС, с которой общается. Изредка попадается что-то незнакомое, и тогда ответ TCP печатается внизу отчета. Если вы обнаружите неопределенную сигнатуру, то сможете помочь в построении базы данных идентификационных меток ОС. Если вы точно знаете, чему она соответствует, скопируйте ее и отправьте по электронной почте на адрес группы разработчиков Nmap. Они добавят ее в базу данных, чтобы в будущем при сканировании машины такого типа ее можно было правильно идентифицировать. Все известные Nmap идентификационные метки TCP содержатся в файле nmap-os-fingerprints в каталоге Data установки Nmap
Отправить через интерфейс (-e имя_интерфейса)	Эта опция заставляет пакеты сканирования отправляться через определенный интерфейс. На практике это необходимо только на машине с несколькими сетевыми платами или если Nmap не опознает ваш сетевой интерфейс автоматически

Существуют дополнительные опции тонкой настройки сканирования, доступные из командной строки. Подробности можно найти в оперативной справке Nmap.

3.1.9. Советы при сканировании с помощью Nmap

Как упоминалось ранее, Nmap может вызывать проблемы в сетях при некорректном или неаккуратном применении. Вот несколько советов, которые помогут сделать сканирование безопасным:

– тщательно выбирайте исходную точку сканирования. Сканирование изнутри сети даст значительно больше информации, чем сканирование извне, через межсетевой экран. Поучительно выполнить сканирование обоих видов и

сравнить результаты. Не страшно, если открытый серверный порт виден изнутри сети; гораздо опаснее, если он виден извне;

– сканирование целесообразно выполнять рано утром или поздно вечером. Таким образом, вы минимизируете вероятность замедления работы жизненно важных серверов и пользовательских машин;

– чтобы исключить перегрузку своей сети, установите в сканирующую машину старую сетевую карту на 10 Мбит/с или подключите ее через концентратор на 10 Мбит/с. Таким образом, максимальный трафик, который сканирование может создать в сети, не превысит 10 Мбит/с, что вряд ли перегрузит сеть на 100 Мбит/с.

3.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустить две различные ОС (какие именно, уточнить у преподавателя).
2. Установить (если еще не установлен) Nmap.
3. Просканировать поочередно сразу один тестовый компьютер, затем другой.
4. Определить ОС, которая установлена на просканированном тестовом компьютере.
5. На одном из тестовых компьютеров установить Firewall (компьютер и тип уточнить у преподавателя).
6. Повторить процесс сканирования компьютера, на котором установлен Firewall.
7. Полученные результаты проанализировать.

3.3. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Реализация решения задачи и листинг конфигурации Nmap.
5. Выводы.

3.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Кто занимается распределением IP-адресов в сети Интернет и назначает ответственных за имена доменов?
2. Какие номера портов называются «общеизвестными»?
3. Чем «недолговечные» номера портов отличаются от «общеизвестных»?
4. Какой основной принцип работы сканера портов?
5. Можно ли считать сканером портов следующую команду: `telnet 192.168.0.1:80`?

6. Можно ли с помощью сканера портов идентифицировать операционную систему, установленную на сканируемом компьютере? Если да, то является ли данный метод совершенным?

7. Определите максимальный размер сети, представленный следующей сетевой маской: `255.255.255.0`.

8. Легальным ли считается процесс сканирования портов другого компьютера?

9. Как влияет на работу сети процесс сканирования портов?

10. Способны ли межсетевые экраны обнаруживать процесс сканирования портов? Если да, то как они могут реагировать на это?

11. Для каких целей можно использовать сканер портов?

12. Что такое троянская программа?

13. Объясните принцип работы «сетевых червей».

14. Что такое Nmap?

15. На каких операционных системах может работать Nmap?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Orebaugh, A. Nmap in the Enterprise: Ваше руководство при сканировании сети / A. Orebaugh, B. Pinkard. – Syngress, 2008. – 384 с.

ЛАБОРАТОРНАЯ РАБОТА №4
СКАНЕРЫ УЯЗВИМОСТЕЙ. NESSUS

Цель работы: ознакомиться с типичными уязвимостями прикладного уровня; изучить универсальный сканер уязвимостей Nessus – бесплатный современный сканер безопасности локальных и удаленных систем.

4.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Что чаще всего делает систему уязвимой? Приложения. Взглянув на эталонную модель OSI, вы увидите, что уровень приложений находится на вершине стека сетевых коммуникаций, что делает его самым сложным и изменчивым. Сканеры уязвимостей позволяют проверить различные приложения в системе на предмет наличия дыр, которыми могут воспользоваться.

Соответствие основных протоколов и уровней модели OSI представлено в табл. 4.1.

Таблица 4.1

Соответствие основных протоколов и уровней модели OSI

Уровень модели OSI	Название уровня	Примеры протоколов
Уровень 7	Прикладной уровень	DNS, FTP, HTTP, SMTP, SNMP, Telnet
Уровень 6	Уровень представления	XDR
Уровень 5	Уровень сеанса	RPC
Уровень 4	Транспортный уровень	NetBIOS, TCP, UDP
Уровень 3	Сетевой уровень	ARP, IP, IPX, OSPF
Уровень 2	Канальный уровень	Arcnet, Ethernet, Token ring
Уровень 1	Физический уровень	Коаксиальный кабель, оптоволокно, витая пара

4.1.1. Выявление дыр в безопасности ваших систем

Необходимо помнить, что компьютерная безопасность сродни другим видам безопасности. Средний компьютерный нарушитель предпочитает цели подоступнее и попроще. Надо опасаться рядовых компьютерных преступников, автоматических «червей» и вирусов. Задача состоит в том, чтобы в вашей сети было меньше дыр, чем у соседа, чтобы хакеры обошли вас стороной при выборе цели для взлома.

В действительности только очень небольшой процент компьютерных преступников исследуют и разрабатывают собственные методы атак. Большинство хакеров действуют с помощью опубликованных и известных дыр в безопасности и средств, показывающих, как проникнуть в ваши компьютеры. Подобную информацию можно найти на бесчисленных веб-сайтах, а хакерские инструменты, использующие эти дыры, доступны для загрузки.

Все основные сбои в работе Интернета, вызванные компьютерными преступлениями, возникали в результате использования дыр в безопасности, из-

вестных за некоторое время до инцидента. Обычно эпидемия распространяется через месяцы или даже годы после того, как становится известна лежащая в ее основе уязвимость. При нашествии Code Red в 2001 г. использовалась уязвимость, корректирующая заплата для которой была доступна более года; то же с «червем» Nimda. «Червь» SQL Slammer, атаковавший базы данных SQL в феврале 2003 г., действовал спустя полгода после выпуска программной коррекции. Факт состоит в том, что в большинстве вторжений в компьютеры используют хорошо известные методы и уязвимости, для которых доступны заплаты или защитные решения. Так называемое мгновенное использование уязвимостей и неопубликованных дыр в безопасности – относительная редкость.

Почему люди пренебрегают простыми вещами и не заделывают дыры в безопасности своих систем? Если бы они это делали, то было бы значительно меньше компьютерных преступлений. Однако множество систем с множеством уязвимостей продолжает существовать по тысяче причин, например:

1. **Нехватка времени или персонала.** Организации сокращают расходы и в трудные времена увольняют технический персонал (информационные технологии (ИТ) не приносят прибыли).

2. **Опасения в отношении стабильности системы.** Хорошо известно, что производители систем при выпуске заплат порой исправляют одну вещь и портят две другие. Для критически важных систем затраты времени и ресурсов для надлежащего тестирования программных коррекций зачастую превышают выгоды от обновления.

3. **Слишком много заплат, чтобы с ними управиться.** Подписчики Windows Update сервиса коррекций Microsoft как минимум раз в неделю получают уведомление о необходимости обновить или «залатать» систему. Для занятых системных администраторов это может быть слишком большой нагрузкой в дополнение к их обычным обязанностям.

4. **Невежество.** Системные администраторы многих организаций просто не знают о существовании проблемы и наличии заплаты. Теперь при автоматическом обновлении от Microsoft эта проблема для систем Windows стала менее острой, но она остается актуальной для других производителей и менее известного программного обеспечения.

Еще один момент, облегчающий жизнь хакерам, состоит в том, что обычно имеется несколько различных путей проникновения в систему. На самом деле для множества выполняемых сервисов может существовать десяток или больше потенциальных окон для входа в подключенный к Интернету сервер. Если атака одного типа не работает, всегда можно попробовать другую. Далее будут описаны некоторые возможные способы, с помощью которых знающий человек может вызвать разрушение системы организации.

Переполнение буфера. Переполнение буфера является, несомненно, наиболее популярным способом взлома систем. Первым документированным использованием переполнения буфера был выпущенный Робертом Моррисом 2 ноября 1988 г. «интернет-червь», который, используя ошибку в программе

Finger, распространял себя с одной машины на другую. Для своего тиражирования он использовал плохую конфигурацию Sendmail и rsh. «Червь» взломал защиту крупнейших университетов и других организаций, пытавшихся справиться с быстро распространяющейся ошибкой. С тех пор возможность переполнения буфера была найдена почти во всех важных программах и часто использовалась теми, кто пытался получить несанкционированный доступ к системам.

Как защитить себя от переполнения буфера? Если вы не хотите отлаживать все применяемое вами программное обеспечение, остается ждать, когда кто-то обнаружит ошибку и сообщит о ней, а затем – когда программистская компания выпустит заплату. К сожалению, отслеживание выпускаемых заплат и определение того, какие из них имеют к вам отношение, не говоря уже об их тестировании и установке, способно занять все рабочее время. Многие организации предпочитают просто не беспокоиться, а организации, прилежно устанавливающие все заплаты, зачастую не успевают делать это вовремя.

Одним из хороших способов узнать, имеются ли условия для переполнения буфера в ваших приложениях, является их тестирование с помощью программного обеспечения сканирования уязвимостей. Это позволит обнаружить большинство известных переполнений буфера, существующих в системе, и своевременно применить корректирующие заплаты, необходимые для устранения этих условий.

Слабые места маршрутизаторов и межсетевых экранов. Эти устройства являются первой линией обороны против посторонних, пытающихся проникнуть в вашу корпоративную сеть. Однако в связи с возрастающей сложностью устройств и мастерством атакующих при некорректном конфигурировании этот рубеж может оказаться слабым. Одна неверная строка конфигурации может разрушить межсетевую защиту. Технический специалист, пытающийся побыстрее настроить доступ для сотрудников или внешних пользователей, чаще будет ошибаться в сторону расширения доступа, а не лучшей защиты.

Даже межсетевые экраны – наиболее защищенные устройства – не обладают абсолютной невосприимчивостью к атакам. Некоторые межсетевые экраны строятся поверх обычных операционных систем, таких как Windows или UNIX, и поэтому могут быть уязвимы для всех обычных атак уровня ОС. Даже если операционная система межсетевого экрана является собственной, в ней могут существовать уязвимости. Многие межсетевые экраны взаимодействуют с пользователями при помощи веб-сервера, а значит, могут быть использованы дыры в веб-интерфейсе.

Использование уязвимостей веб-серверов. В наше время практически каждая компания должна иметь веб-сервер, хотя известно наличие ошибок и дыр в безопасности этой системы. Сама идея веб-сервера – возможность брать с сервера файлы без какой-либо аутентификации – создает потенциал для брешей

в защите. Некоторые веб-серверы защищены лучше, чем другие, но у каждого есть свои недостатки. Взлом этой системы может вызвать искажение не только веб-страницы, но и баз данных и других внутренних систем, к которым он осуществляет доступ, что в наше время является общепринятым.

Серверы DNS. Серверы, которые управляют и поддерживают доменные имена вашей организации, являются привлекательной целью для хакеров. Основной DNS-сервер, BIND (Berkeley Internet Name Domain), постоянно находился в первой десятке наиболее эксплуатируемых хакерами сервисов. DNS – старая программа, и сама ее структура способствует наличию дыр (вместо модульной архитектуры – один монолитный бинарный файл). DNS часто запускается от имени суперпользователя, что делает его взлом еще более опасным. Кроме того, поскольку DNS трудно настраивать и его плохо понимают, он зачастую сконфигурирован неправильно и защищен плохо. Настройки межсетевых экранов для DNS нередко сконфигурированы неверно – большинство системных администраторов разрешают нефильтрованный доступ внутрь и наружу.

Управление пользователями и файлами. Эта область является одной из самых уязвимых для информационной безопасности. Вы должны предоставить пользователям доступ к системам и программам, которые нужны им для выполнения работы. Однако ключевым принципом хорошей защиты является принцип минимизации привилегий, то есть предоставление пользователям минимально достаточного для работы доступа – и не больше. Определение этого уровня – непростая задача: слишком мало прав влекут за собой звонки из службы помощи пользователям и жалобы, слишком много прав ослабят защиту своей системы.

Пустые и слабые пароли. Иметь аутентификацию с пустым паролем кажется невозможным, но во многих сетях делается именно это. Также неожиданно распространено использование комбинации пользователь/пароль вида administrator/administrator, которая служит подразумеваемой настройкой Windows. «Черви» и программы взлома автоматически проверяют это условие. Если они его находят, то имеют полный административный доступ к системе. Аналогично, когда пользователи задают пароли, они могут просто оставить их пустыми. Это дает шанс любому, имеющему список пользователей, попытаться найти сетевые сервисы с пустым паролем. При сканировании уязвимостей перечисленные условия будут проверяться.

Утечка информации. В поисках способа проникнуть в систему хакеры или взломщики начинают с некоторой базовой разведки. Они пытаются как можно больше узнать о вашей системе и сети, прежде чем попытаться взломать их: ищут электронные эквиваленты выключенного света, накопившихся газет, незакрытых окон и т. д. Делается это с помощью ряда инструментов (например сканеры портов) или других средств взлома, доступных в Интернете. К сожале-

нию, многие операционные системы охотно помогают этим незаконным сборщикам информации, особенно Windows. Поскольку эта ОС создавалась как самонастраивающаяся сетевая система, она предлагает всевозможную информацию любой системе, которая опрашивает ее с помощью подходящей команды. На основе этих данных внешний пользователь может сгенерировать списки пользователей, разделяемые диски и каталоги, имена систем и сотрудников и другую информацию, полезную для взлома методом грубой силы, когда с помощью автоматизированных программ пробуются различные комбинации паролей, а также для применения методов морально-психологического воздействия.

Атаки на доступность (DoS). Не сумев получить доступ к вашей системе, многие компьютерные преступники отключат ее, чтобы никто другой не мог ей воспользоваться. При большом объеме операций электронной коммерции час простоя может стоить миллионы долларов. Атаки на доступность могут принимать различные формы – от простого затопления основных маршрутизаторов потоками данных до реального использования слабого места в программе с целью нарушения работы сервиса и всего сервера. От первого трудно защититься, но последнее вполне предотвратимо при выявлении и последующем исправлении или исключении условия, которое создает возможности для атаки на доступность.

4.1.2. Сканеры уязвимостей

Сканеры уязвимостей – это программные или аппаратные средства, служащие для осуществления диагностики и мониторинга сетевых компьютеров, позволяющие сканировать сети, компьютеры и приложения на предмет обнаружения возможных проблем в системе безопасности, оценивать и устранять уязвимости. Они позволяют проверить различные приложения в системе на предмет наличия «дыр», которыми могут воспользоваться злоумышленники.

Nessus – действительно исключительная программа. Это великолепный пример того, как хорошо могут работать проекты с открытыми исходными текстами. Он надежен, хорошо документирован, отлично поддерживается, он лучший в своем классе. Nessus постоянно попадает в число лучших среди всех сканеров уязвимостей – коммерческих и некоммерческих. Это поразительно, если принять во внимание его конкурентов, стоящих тысячи долларов и созданных крупными компаниями.

Глубина тестирования. В настоящее время Nessus предлагает более 2000 отдельных тестов уязвимостей, которые охватывают практически все области потенциально слабых мест в системах. Очень немногие существующие сканеры могут конкурировать с достигнутым в Nessus уровнем тестирования, и новые тесты добавляются ежедневно всемирной сетью разработчиков. Скорость выпуска новых тестов для выявляемых уязвимостей обычно измеряется днями, если не часами.

Архитектура «клиент – сервер». Для выполнения проверок безопасности Nessus опирается на архитектуру «клиент – сервер». Сервер выполняет проверки, а клиент конфигурирует и управляет сеансами. Тот факт, что клиент и сервер могут быть разделены, предоставляет несколько уникальных преимуществ. Во-первых, сканирующий сервер можно расположить вне вашей сети, но обращаться к нему изнутри сети через клиента. Во-вторых, различные клиенты могут поддерживать разные операционные системы.

Независимость. Поскольку исходные тексты Nessus открыты, а встраиваемые модули написаны разнообразными группами специалистов по информационной безопасности, не приходится опасаться каких-либо конфликтов интересов, возможных в коммерческих компаниях.

Встроенный язык сценариев атак. В дополнение к архитектуре со встраиваемыми модулями в Nessus имеется собственный язык сценариев атак, называемый NASL (Nessus Attack Scripting Language). Этот простой в изучении служебный язык позволяет легко и быстро писать собственные встраиваемые модули безопасности, не зная Си или всей внутренней системы основной программы.

Интеграция с другими средствами. Сканер уязвимостей Nessus можно применять сам по себе или совместно с некоторыми другими защитными средствами с открытыми исходными текстами. Часть из них рассмотрена в этой лабораторной работе, и все они являются лучшими из имеющихся. Вместо встроенного можно применить лучший в мире сканер портов – Nmap. Сканер портов в Nessus быстрее и немного экономнее в расходовании памяти, но Nmap, как вы узнали из лабораторной работы №5, предоставляет значительно больше возможностей и настроек. Почти все параметры Nmap можно конфигурировать из клиента Nessus. Nessus также работает с Nikto и Whisker – средствами, которые выполняют более сложные проверки на веб-серверах.

Интеллектуальное тестирование. Nessus можно настроить так, чтобы он не выполнял автоматически все тесты уязвимостей на всех хостах. На основе результатов сканирования портов или других исходных данных, таких как результаты предыдущих тестов уязвимостей, Nessus будет запускать только тесты, подходящие для данной машины.

Множество форматов отчетов. Nessus входит в число лучших программ с открытыми исходными текстами и по возможностям генерации отчетов. Хотя они не совершенны, данные сканирования можно выдать почти в любом формате. Базовый HTML и HTML с круговыми диаграммами и графиками – два наиболее популярных формата. В отчеты входят итоговые данные, так что почти без редактирования их можно разместить на внутреннем веб-сайте. Под-

держиваются также форматы отчетов XML, LaTeX и обычный текст. Клиент Windows предлагает дополнительные форматы отчетов.

4.1.2.1. Входная страница Nessus

Первое, что вы увидите, будет входная страница Nessus (рис. 4.1). Вследствие архитектуры «клиент – сервер», прежде чем начать работать с Nessus, необходимо сначала войти в его сервер. Если клиент и сервер будут запускаться на одной машине, то правильными параметрами входа будут следующие:

- сервер: localhost;
- порт: 1241;
- входное имя: имя, заданное при настройке Nessus;
- пароль: пароль, заданный при настройке Nessus.

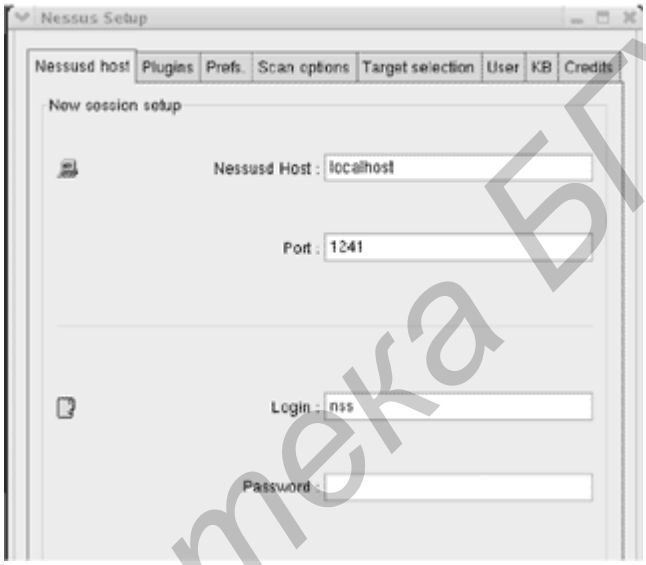

Рис. 4.1. Входной экран Nessus

Клиент и сервер могут также выполняться на разных машинах. В этом случае замените localhost на IP-адрес или имя хоста сервера Nessus. Это дает возможность входить из дома в серверы Nessus, функционирующие в организации, и запускать сканирование поздней ночью.

4.1.2.2. Вкладка встраиваемых модулей Nessus

После входа вы получаете доступ к различным вкладкам. С помощью вкладки Plugins можно выборочно включать или отключать определенные группы или отдельные встраиваемые модули (рис. 4.2). На вкладке перечислены все категории, а когда вы щелкаете мышью на некоторой категории, то ниже появляются все ее модули. Снимая флажок, который находится справа от элемента, можно отключить категорию или модуль.

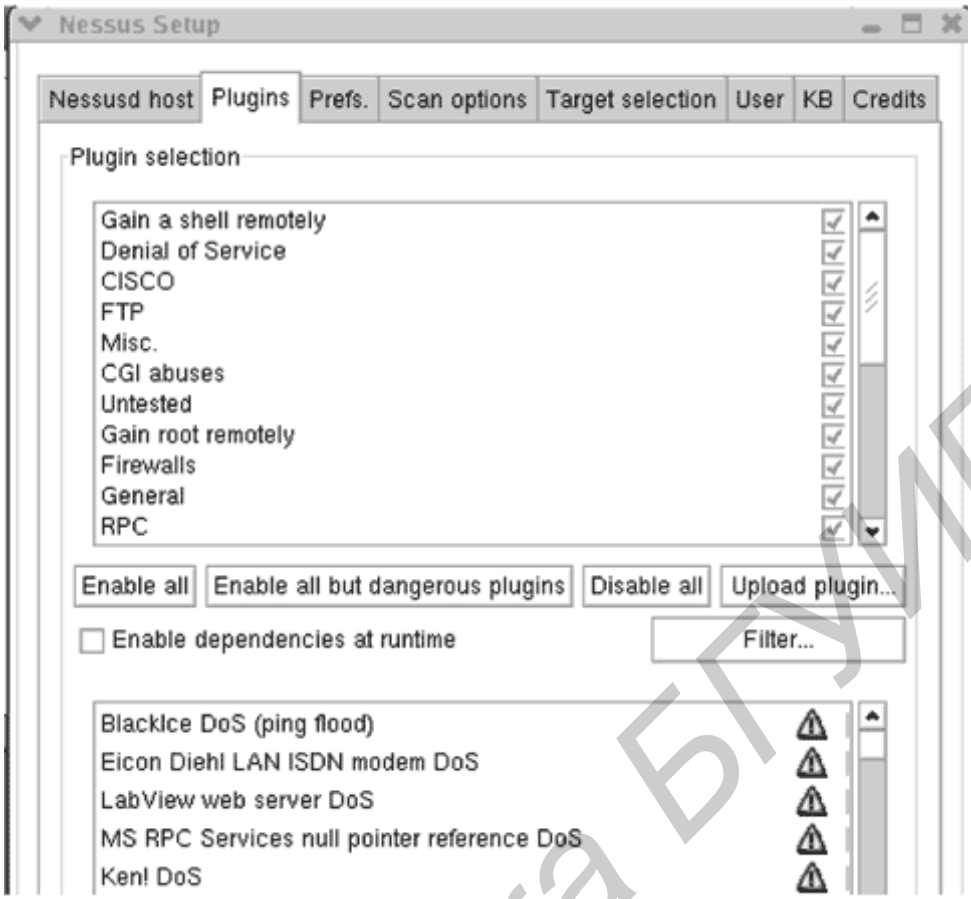

Рис. 4.2. Вкладка встраиваемых модулей Nessus (Plugins)

Модули, которые могут вызывать проблемы у сервиса или крах серверов, отмечены треугольником с восклицательным знаком (см. рис. 4.2). Кроме того, в Nessus имеются кнопки, которые позволяют быстро включить все встраиваемые модули (Enable all), включить все модули, кроме опасных (Enable all but dangerous plugins), отключить все модули (Disable all) или загрузить пользовательский встраиваемый модуль (Upload plugin...). Можно использовать кнопку Filter для сортировки модулей по имени (Name), описанию (Description), сводке (Summary), автору (Author), идентификационному номеру (ID) или категории (Category). Как правило, рекомендуется запускать Nessus с отключенными опасными модулями; включайте их, только если вы готовы к настоящей проверке доступности и сознательно идете на риск краха некоторых серверов.

4.1.2.3. Вкладка предпочтений Nessus

Большинство серверных опций Nessus конфигурируются с помощью вкладки Preferences (рис. 4.3). В следующих разделах и подразделах даны подробные сведения об этих опциях.

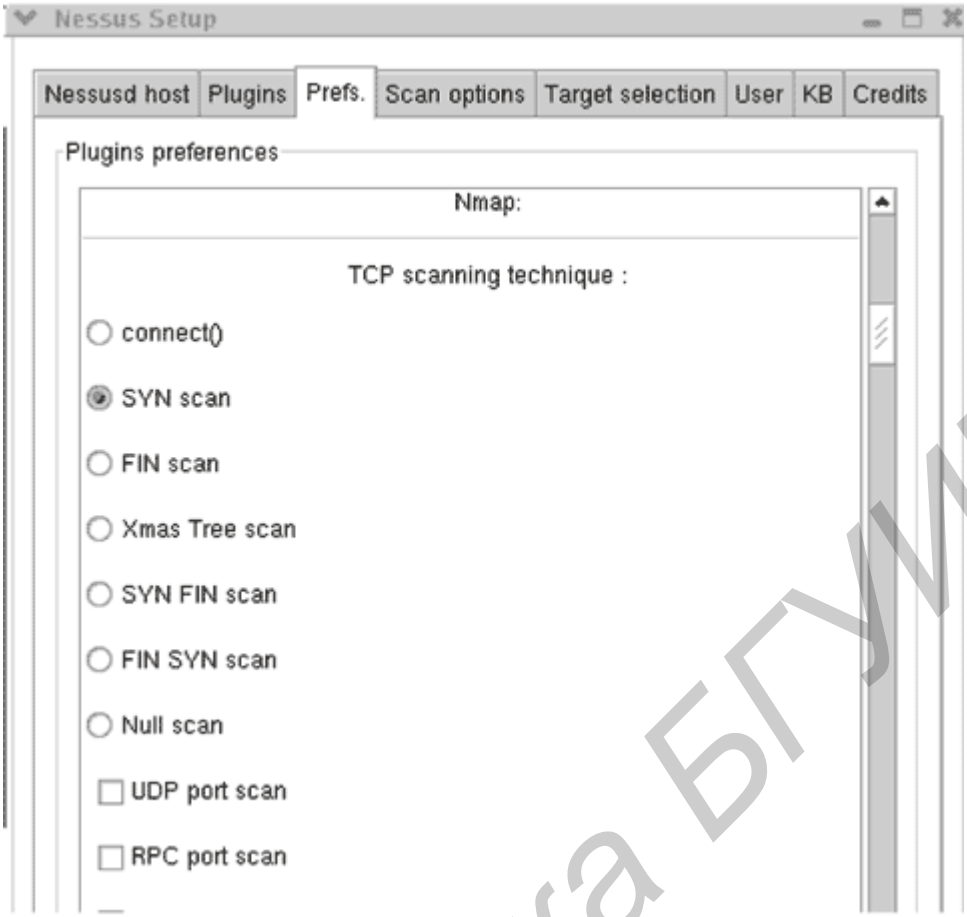

Рис. 4.3. Вкладка предпочтений Nessus (Preferences)

Nmap. Настройки Nmap используются для индивидуального конфигурирования части, отвечающей при выполнении тестов за сканирование портов. Многие из них непосредственно связаны с настройками Nmap, обсуждавшимися в лабораторной работе №5, куда и следует обратиться за подробными сведениями о каждой опции.

Login configurations (конфигурации входа). В этом разделе задаются счета для входа, если вы хотите, чтобы Nessus более глубоко тестировал некоторые сервисы. Стандартное сканирование Nessus проверяет сеть без привлечения каких-либо дополнительных данных о ней, кроме IP-адресов. Однако если задать входное имя и пароль для конкретного сервиса, то Nessus подвергнет его дополнительным проверкам.

Brute-force login (Hydra) (вход методом грубой силы). Этот раздел позволяет воспользоваться дополнительной программой Hydra, которая проверяет целостность паролей вашей системы. Вы предоставляете ей файл входных имен и паролей, а она попытается пройти по всему списку для всех указанных сервисов.

Services (сервисы). Этот раздел предназначен для тестирования сервисов SSL. Можно задать проверяемые сертификаты и получить отчет об уровне шифрования, который поддерживают ваши веб-серверы. Это могут быть ло-

кальные серверы, которые все еще допускают старое 40-битное шифрование, что в наше время считается небезопасным для критически важных данных.

4.1.2.4. Вкладка Scan Options (опции сканирования)

В отличие от отдельных тестов на вкладке предпочтений эта вкладка содержит настройки, влияющие на весь процесс сканирования (рис. 4.4).

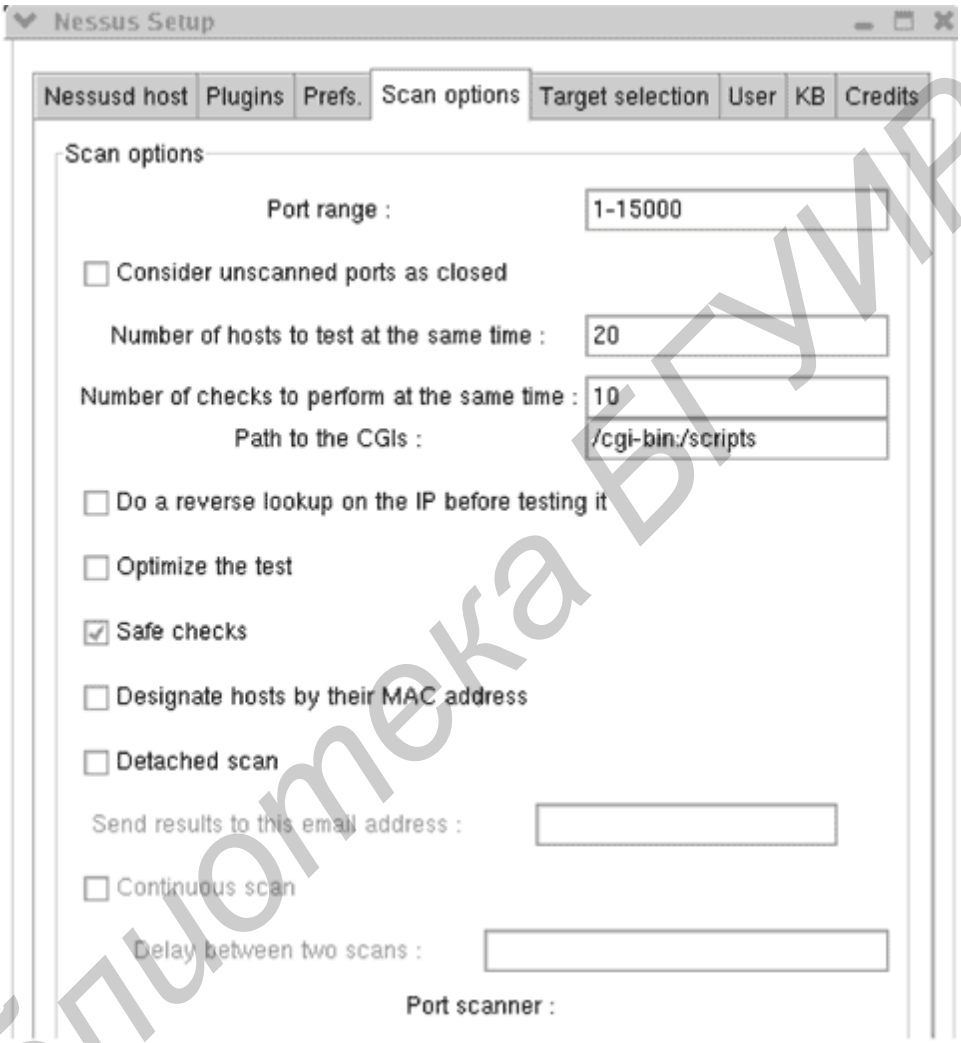

Рис. 4.4. Вкладка Scan Options в Nessus

Port range (диапазон портов). Данный параметр контролирует фазу сканирования портов, задавая целевой диапазон (по умолчанию – 1…15 000, что должно охватить большинство обычных сервисов). Если вы желаете поискать троянские программы и другие сервисы, действующие на необычно больших номерах портов, подразумеваемый диапазон необходимо расширить и сканировать все 65 535 портов TCP и UDP.

Consider unscanned ports as closed (считать несканированные порты закрытыми). Данная опция заставляет Nessus объявлять несканированные порты закрытыми. Вы можете что-то пропустить, если посредством предыду-

щей опции не задали достаточно широкий диапазон портов, но зато сканирование выполнится быстрее и с меньшим сетевым трафиком.

Number of hosts to test at the same time (число одновременно тестируемых хостов). Задается число хостов, которые Nessus тестирует параллельно. В большой сети возникает желание завысить данный параметр и тестировать все хосты одновременно. Однако с некоторого момента это становится контрпродуктивным. В действительности сканирование будет длиться дольше, а может и вообще не закончится, если остановится на каком-то одном хосте.

Number of checks to perform at the same time (число одновременно выполняемых проверок). Nessus поддерживает многозадачность не только в смысле одновременного тестирования нескольких хостов, но и в смысле одновременного выполнения нескольких проверок. Подразумеваемое значение 10 представляется разумным, однако в зависимости от производительности сервера Nessus можно увеличивать или уменьшать его.

Path to the CGIs (маршрут к CGI). Подразумеваемое место, где Nessus будет искать на удаленной системе CGI-процедуры для их тестирования. Если вы используете на машине необычную конфигурацию, то необходимо задать правильный маршрут, чтобы Nessus проверил CGI-процедуры.

Do a reverse lookup on the IP before testing it (выполнять обратный поиск IP-адреса перед тестированием). При использовании данной настройки перед проверкой делается попытка выполнить обратный поиск DNS и определить имя хоста для каждого IP-адреса. Это существенно замедляет сканирование и по умолчанию отключено.

Optimize the test (оптимизировать тестирование). По умолчанию при выполнении тестов Nessus пытается поступать разумно и не делать проверок, неприменимых к конкретному хосту.

Safe checks (безопасные проверки). По умолчанию всегда устанавливается данный режим. Это значит, что Nessus не будет выполнять никаких небезопасных проверок, которые могут вызвать аварию или как-то иначе повредить серверу.

4.1.2.5. Вкладка Target Selection (выбор цели)

На этой вкладке задаются цели сканирования (рис. 4.5). Перечислим способы задания целей сканирования:

- один IP-адрес: 192.168.0.1;
- IP-адреса, разделенные запятыми: 192.168.0.1,192.168.0.2;
- IP-диапазоны, разделенные дефисом: 192.168.0.1–192.168.0.254;
- стандартная нотация с косой чертой: 192.168.0.1/24 (сеть класса C из 256 адресов);
- имя хоста: myhost.example.com;
- любая комбинация вышеприведенных обозначений, разделенных запятыми: 192.168.0.1–192.168.0.254,195.168.0.1/24.

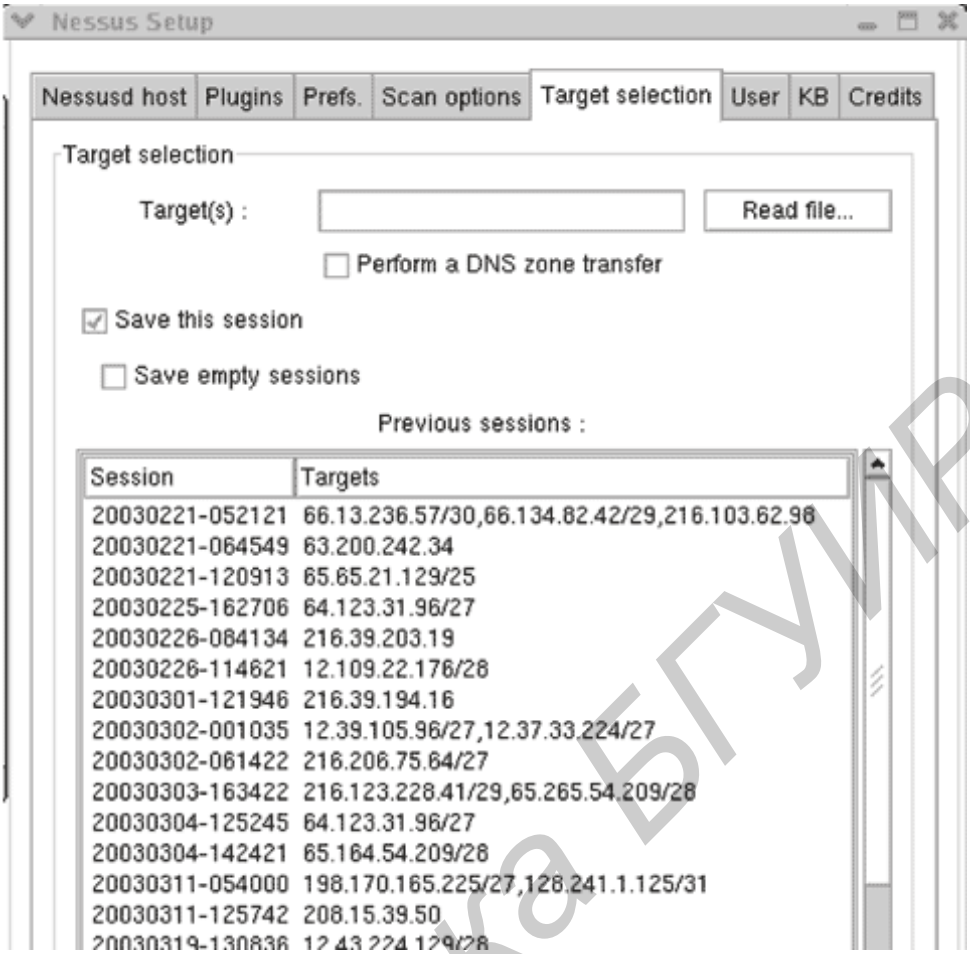

Рис. 4.5. Вкладка Target Selection в Nessus

4.1.2.6. Вкладка User (Пользователь)

На этой вкладке отображаются все пользователи сервера Nessus и все ассоциированные с ними правила (например, возможность входа только с определенного IP-адреса). Счета пользователей создаются с помощью процедуры nessus-adduser, но на этой вкладке всегда можно отредактировать или добавить правила для любого существующего пользователя.

4.1.3. Особенности сканирования уязвимостей

Сканирование портов является довольно безобидной деятельностью, хотя она и настораживает, когда вы видите ее зафиксированной в журналах. Однако тестирование уязвимостей может быть существенно более разрушительным, приводя к авариям серверов, разрывая соединения с Интернетом, или даже удаляя данные (например тест Integrist). Многие из тестов Nessus специально созданы для организации атак на доступность. Даже с включенной опцией безопасной проверки тесты могут вызывать проблемы на некоторых системах. В связи с этим дадим несколько рекомендаций.

1. **Не сканируйте без разрешения.** Вы никогда не должны сканировать сеть, которая не находится под вашим непосредственным контролем или если вы не имеете официального разрешения от владельца. Некоторые из видов активности, инициированной Nessus, могут юридически рассматриваться хакерским взломом (особенно при включенной опции атак на доступность).

2. **Убедитесь, что все резервные копии актуальны.** Вы всегда должны быть уверены, что ваши резервные копии актуальны, но это вдвойне важно при сканировании уязвимостей на тот случай, если сканирование приведет к проблемам с сервером.

3. **Планируйте время сканирования**. В продолжение предыдущего замечания не забывайте координировать время проведения сканирования для получения требуемых результатов с минимальным влиянием на других служащих. Сканирование почтового сервера в 8 часов утра, когда все стремятся получить свою почту, может нарушить рабочий процесс.

4. **Избегайте избыточного сканирования.** Планируйте сканирование так часто, как сочтете необходимым, но не полагайте автоматически, что ежесуточное сканирование сделает вашу сеть более безопасной.

5. **Правильно размещайте сервер сканирования.** Если вы хотите по-настоящему проверить внешнюю (из Интернета) уязвимость вашей информационной системы, следует разместить сервер Nessus вне вашего межсетевого экрана. При сканировании внутренней сети сервер необходимо разместить позади межсетевого экрана. Установка Nessus на ПК-блокноте может облегчить выполнение сканирования как изнутри, так и извне вашей сети, не требуя множества машин.

4.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустить две различные ОС (какие именно уточнить у преподавателя).
2. На одной из них установить и настроить Firewall.
3. Установить и сконфигурировать Nessus.
4. Продумать, создать и обосновать свой сценарий сканирования системы (не выбирайте «опасные тестовые сценарии»), например, тестирование настроек Firewall.
5. Запустить сканирование и получить результаты его выполнения.
6. Полученные результаты прикрепить к отчету и обосновать.

4.3. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Реализация решения задачи и скриншоты конфигурации Nessus.
5. Выводы.

4.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каковы основные причины несоблюдения требований безопасности своих систем?

2. Что гласит основной принцип хорошей защиты при управлении пользователями и файлами?

3. Для чего предназначен «сканер уязвимостей»?

4. Что такое Nessus?

5. Какие достоинства у архитектуры построения «клиент – сервер», на которой базируется Nessus?

6. Что такое «пустой» и «слабый» пароли?

7. Что такое NASL?

8. С какими сторонними утилитами может интегрироваться Nessus?

9. В каком виде Nessus может представлять отчеты о результатах сканирования?

10. Какие вы знаете достоинства и недостатки встроенного в Nessus сканера портов по сравнению с существующим Nmap?

11. Зачем в Nessus реализована поддержка MySQL?

12. Чем Nessus WX отличается от Nessus?

13. Почему нельзя сканировать сеть без разрешения?

14. Зачем планировать время сканирования?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Beale, J. Nessus Network Auditing / J. Beale, R. Alder. – Syngress, 2004. – 544 с.

2. Lambert, M. Nessus (software) / M. Lambert, M. Surhone. – Книга по требованию, 2010. – 184 с.

ЛАБОРАТОРНАЯ РАБОТА №5
СЕТЕВЫЕ АНАЛИЗАТОРЫ. TcpDump И WIRESHARK

Цель работы: ознакомиться с основными сетевыми анализаторами (TcpDump, WinDump и Wireshark) и их историей, разобрать принципы их функционирования; ознакомиться с правилами анализа сети.

5.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Мы приступаем к рассмотрению некоторых средств, позволяющих действовать и реагировать, когда вопреки всем вашим профилактическим мерам в сети проявляются компьютерные атаки или проблемы безопасности. К этой категории средств относятся сетевые анализаторы. Грубо говоря, сетевой анализатор (network sniffer) прослушивает или «обнюхивает» (sniffs) пакеты определенного физического сегмента сети. Это позволяет анализировать трафик на наличие некоторых шаблонов, исправлять определенные проблемы и выявлять подозрительную активность

В отличие от средств, описанных ранее, анализаторы действуют на более низком уровне. Если обратиться к эталонной модели OSI, то анализаторы проверяют два нижних уровня – физический и канальный.

Таблица 5.1

Соответствие основных протоколов и уровней модели OSI

Уровень модели OSI	**Название уровня**	**Примеры протоколов**
Уровень 7	Прикладной уровень	DNS, FTP, HTTP, SMTP, SNMP, Telnet
Уровень 6	Уровень представления	XDR
Уровень 5	Уровень сеанса	RPC
Уровень 4	Транспортный уровень	NetBIOS, TCP, UDP
Уровень 3	Сетевой уровень	ARP, IP, IPX, OSPF
Уровень 2	Канальный уровень	Arcnet, Ethernet, Token ring
Уровень 1	Физический уровень	Коаксиальный кабель, оптоволокно, витая пара

Физический уровень – это реальная физическая проводка или иная среда, примененная для создания сети. На канальном уровне происходит первоначальное кодирование данных для передачи через конкретную среду. Сетевые стандарты канального уровня включают беспроводной 802.11, Arcnet, коаксиальный кабель, Ethernet, Token Ring и многое другое. Анализаторы обычно зависят от типа сети, в которой они работают. Например, для анализа трафика в сети Ethernet вы должны иметь анализатор Ethernet.

В данной лабораторной работе рассматриваются несколько сетевых анализаторов Ethernet с открытыми исходными текстами. Мы остановимся на Ethernet, так как этот протокол наиболее употребителен в локальных сетях.

5.1.1. Особенности применения сетевых анализаторов

Чтобы применять сетевые анализаторы этично и продуктивно, необходимо выполнять следующие рекомендации:

Получение разрешений. Анализ сети, как и многие другие функции безопасности, имеет потенциал для ненадлежащего использования. Перехватывая все данные, передаваемые по сети, вы вполне можете подсмотреть пароли для различных систем, содержимое почтовых сообщений и другие критичные данные, как внутренние, так и внешние, так как большинство систем не шифрует свой трафик в локальной сети. Если подобные данные попадут в нехорошие руки, это, очевидно, может привести к серьезным нарушениям безопасности. Кроме того, это может стать нарушением приватности служащих, если таковая фигурирует в политике организации. Всегда получайте письменное разрешение руководства, желательно высшего, прежде чем начинать подобную деятельность.

Топологии сети. Прежде чем настраивать анализатор, убедитесь, что вы полностью понимаете физическую и логическую организацию вашей сети. Проводя анализ в неправильном месте сети, вы либо получите ошибочные результаты, либо не сможете увидеть то, что ищете. Проверьте, что между анализирующей рабочей станцией и тем, что вы собираетесь наблюдать, нет маршрутизаторов. Маршрутизаторы будут направлять трафик в сегмент сети, только если происходит обращение к расположенному там узлу.

Жесткие критерии поиска. В зависимости от того, что вы ищете, использование открытого фильтра (то есть показ всего) сделает вывод данных объемным и трудным для анализа. Используйте специальные критерии поиска, чтобы сократить вывод, который выдает ваш анализатор. Даже если вы не знаете точно, что ищете, можно тем не менее написать фильтр для ограничения результатов поиска. Поступая таким образом, вы сделаете результаты анализа значительно более полезными.

Эталонное состояние сети. Применив сетевой анализатор во время нормальной работы и записав итоговые результаты, вы получите эталонное состояние, которое можно сравнивать с результатами, полученными во время попыток выделения проблемы. При помощи этих данных можно определить, когда сеть насыщается и каковы основные причины этого – перегруженный сервер, рост числа пользователей, изменение типа трафика и т. п. Если есть точка отсчета, проще понять, кто и в чем виноват.

5.1.2. Установка TcpDump

Существует много доступных анализаторов как свободных, так и коммерческих, но TcpDump наиболее доступен и недорог. Он поставляется с большин-

ством дистрибутивов UNIX, включая Linux и BSD. TcpDump полностью оправдывает свое имя: он выдает содержимое пакетов TCP/IP, проходящих через сетевой интерфейс, на устройство вывода (обычно на экран или в файл).

Чтобы анализатор Tcpdump работал, он должен иметь возможность перевести сетевую плату в так называемый режим прослушивания (или неразборчивый режим – promiscuous mode). Это означает, что сетевая плата будет перехватывать весь трафик Ethernet, а не только тот, что адресован ей.

5.1.3. Запуск TcpDump

Существует ряд операций для фильтрации вывода, чтобы найти определенный тип трафика или снизить общий объем вывода. На самом деле, в активно используемой сети нефильтрованный вывод TcpDump будет пролетать на экране быстрее, чем вы сможете его прочитать. Однако прямо сейчас для демонстрации возможностей TcpDump запустим его из командной строки, набрав просто: `tcpdump`.

Вы увидите весь нефильтрованный трафик TCP, проходящий через плату Ethernet вашей машины. Он может выглядеть примерно так, как в приведенном ниже примере:

```
12:25:38.610428 192.168.1.3.4870 > 65.83.241.167.domain: 1416+ PTR?
167.241.83.65.in-addr.arpa. (44) (DF)
12:25:38.649808 65.83.241.167.domain > 192.168.1.3.4870: 11416 1/0/0 (69)
12:25:43.497909 arp who-has 192.168.1.1 tell 192.168.1.3
12:25:43.498153 arp reply 192.168.1.1 is-at 0:6:25:9f:34:ac
12:25:43.498943 192.168.1.3.4870 > 65.83.241.167.domain: 11417+ PTR?
1.1.168.192.in-addr.arpa. (42) (DF)
12:25:43.533126 65.83.241.167.domain > 192.168.1.3.4870: 11417 NXDomain
0/1/0 (119)
12:25:44.578546 192.168.1.1.8783 > 192.168.1.255.snmptrap: Trap(35)
E:3955.2.2.1 192.168.1.1 enterpriseSpecific[specific-trap(1)!=0]
43525500[|snmp]
```

На первый взгляд, выдача кажется запутанной, но если разбить ее на составляющие, то смысл начинает проясняться. Первое число является временной меткой с точностью до долей секунды, так как в активно используемой сети каждую секунду проходит множество пакетов. Следующее число – это IP-адрес отправителя пакета, за которым следует «>» (знак больше), а затем целевой адрес. Наконец, могут присутствовать некоторые комментарии и другие данные. В примере можно видеть несколько различных видов трафика, включая трафик DNS (domain), ARP и SNMP.

По умолчанию Tcpdump выполняется, пока не будет остановлен нажатием Ctrl+C или другим сигналом прерывания. Когда TcpDump останавливается, он выдает сводные данные о просмотренном трафике, включая:

– пакеты, полученные фильтром: количество пакетов, обработанных фильтром TcpDump, а не общее число пакетов TCP в сети (если только вы не выполняете TcpDump без критериев фильтрации);

– пакеты, отброшенные ядром: число пакетов, которые были отброшены в связи с отсутствием ресурсов в системе.

Обычно вы будете запускать TcpDump с некоторыми установленными опциями или фильтрами, чтобы уменьшить и сфокусировать вывод. Общий вид инструкции запуска TcpDump таков:

```
TcpDump опции выражения
```

Замените опции и выражения одной или несколькими допустимыми переменными. Опции TcpDump перечислены в табл. 5.2.

Таблица 5.2

Опции Tcpdump

Опция	Описание
-a	Пытается преобразовать адреса в имена. Это создает дополнительную нагрузку на систему и может привести к потере пакетов
-c число	Останавливает TcpDump после обработки заданного числа пакетов
-C размер_файла	Ограничивает размер выходных файлов заданным числом байт
-d	Выдает процедуру сопоставления пакетов с образцом в удобочитаемом виде и затем останавливается
-e	В каждой строке выдачи печатает заголовок канального уровня (в сетях Ethernet это MAC-адрес)
-F файл	Использует файл (а не сеть) для ввода данных. Это удобно для анализа событий «постфактум»
-n	Не преобразовывает адреса в имена
-q	Печатает быстрый вывод. Печатается меньше протокольной информации, поэтому строки оказываются короче
-T тип	Заставляет интерпретировать пакеты, выбранные заданным в выражении фильтром, в соответствии с указанным типом
-t	Не печатает метку времени в каждой строке
-tt	Печатает неформатированную метку времени в каждой строке
-ttt	Печатает интервал времени между пакетами
-tttt	Печатает в каждой строке дату, а затем метку времени в подразумеваемом формате
-v	Использует чуть более подробный вывод, включающий время жизни, идентификатор, общую длину и поля опций каждого пакета

5.1.4. Выражения TcpDump

Выражения TcpDump определяют выбор отображаемых сетевых пакетов. Именно здесь происходит реальная работа TcpDump. Выдаются только те объекты, которые соответствуют выражению; если выражения не заданы, отображаться будут все пакеты. Выражение TcpDump состоит из одной или нескольких директив, называемых примитивами, которые, в свою очередь, состоят из идентификатора и следующего за ним квалификатора. В табл. 5.3 перечислены три различных вида квалификаторов, а в табл. 5.4 – доступные комбинации примитивов.

Таблица 5.3

Квалификаторы TcpDump

Квалификатор	Описание
Тип	Определяет, к чему относится идентификатор, заданный как имя или номер. Возможными типами служат host, net и port. Например, host foo, net 128.3 или port 20
Направление	Определяет направление трафика от определенного идентификатора. Возможными направлениями служат src; dst; src or dst и src and dst (src обозначает исходный адрес, dst – целевой)
Протокол	Позволяет определить протокол для фильтрации. Возможными протоколами являются ether, fddi, tr, ip, ipv6, arp, rarp, decnet, tcp и udp. Если протокол не задан, то допустимы все протоколы, совместимые с остальной частью выражения. При помощи фильтров с этим квалификатором можно определить, какая машина делает чрезмерное количество arp-запросов и отбрасывания на фильтре udp-запросов, которых немало во многих сетях, так как DNS использует udp

Таблица 5.4

Допустимые комбинации примитивов

Комбинация	Описание
dst host хост	Показывает только трафик, адресованный хосту, который может быть задан IP-адресом или именем
src host хост	Показывает только трафик, исходящий из хоста
host хост	Показывает как исходящий, так и входящий трафик хоста
ether dst Ethernet-хост	Показывает трафик, предназначенный для указанного Ethernet-хоста, который может быть задан либо именем, либо MAC-адресом
ip proto протокол	Перехватывает трафик заданного протокола. Допустимыми протоколами служат icmp, icmpv6, igmp, igrp, pim, ah, esp, vrrp, udp и tcp. Имена tcp, udp и icmp должны помещаться между двумя обратными косыми чертами, чтобы они не читались как ключевые слова. Пример: ip proto \tcp\
decnet хост	Фильтрует трафик DECnet с исходным или целевым адресом хоста
ip	Сокращенный вариант описанной выше комбинации ether proto ip. Ловит трафик, соответствующий Ethernet-протоколу ip
ip6	Сокращенный вариант описанной выше комбинации ether proto ip6. Ловит трафик, соответствующий Ethernet-протоколу ip6
arp	Сокращенный вариант описанной выше комбинации ether proto arp. Ловит трафик, соответствующий Ethernet-протоколу arp
tcp	Сокращенная форма комбинации ip proto tcp

5.1.5. Примеры применения TcpDump

Ниже представлены примеры применения TcpDump:

1. **Просмотр всего входящего и исходящего трафика определенного хоста.** Если вы хотите отслеживать только входящий и исходящий трафик определенного хоста, то можно отфильтровать все остальное с помощью простого выражения «host». Например, чтобы следить за хостом с IP-адресом 192.168.1.1, нужно выполнить следующую инструкцию:

```
TcpDump -n host 192.168.1.1
```

2. **Наблюдение за входящим и исходящим трафиком определенного порта.** Если вы хотите проследить за использованием определенного приложения, можно применить TcpDump для улавливания всего трафика, направляемого в определенный порт TCP/UDP. Если приложением, за которым вы пытаетесь наблюдать, является Telnet (порт 23), то это можно сделать с помощью следующего выражения TcpDump:

```
TcpDump -n port 23
```

3. **Выявление вредоносной рабочей станции.** Если возникли сетевые проблемы, и вы подозреваете, что вредоносный компьютер пытается нарушить вашу сеть, можно применить TcpDump для быстрого прослеживания виновника. Вне зависимости от того, будет ли это неисправная сетевая плата или ПК с троянской программой, вызывающей атаку на доступность, TcpDump поможет прояснить ситуацию. Сначала попробуйте просто запустить TcpDump без фильтрации и посмотреть, что порождает большую часть трафика. Используйте опции `-a` и `-e` для генерации имен и MAC-адресов:

```
TcpDump -ae
```

Отметим, что можно объединять две буквы с одним дефисом. Если вывод на экране проскальзывает слишком быстро, используйте опцию `-c 1000`, чтобы остановиться после получения 1000 пакетов.

5.1.6. Wireshark

Wireshark (ранее Ethereal) – программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Имеет графический пользовательский интерфейс. В июне 2006 г. проект был переименован в Wireshark из-за проблем с торговой маркой. Функциональность, которую предоставляет Wireshark, очень схожа с возможностями программы TcpDump, однако Wireshark имеет графический пользовательский интерфейс и гораздо больше возможностей по сортировке и фильтрации информации.

Главное окно Wireshark изображено на рис. 5.1.

Wireshark – это утилита, перехватывающая сетевой трафик с одного или нескольких сетевых интерфейсов. Можно задать набор правил, определяющих интересующие вас пакеты. Затем эти пакеты откладываются в сторонку (точнее сказать, в буфер) и всесторонне анализируются. Перехваченные пакеты можно также сохранить в файл и загрузить из него. Обычно Wireshark переводит сетевой интерфейс в так называемый «неразборчивый» режим и принимает все пакеты, а не только адресованные вашему компьютеру. Для перевода в «неразборчивый» режим нужны привилегии суперпользователя, поэтому Wireshark обычно запускается от имени root.

Начнем с простого примера работы Wireshark: рассмотрим результат обращения к веб-странице на сервере 192.168.1.67 из браузера на клиенте

192.168.1.69. В верхней части изображения (на экране выделяется зеленым цветом) показан весь обмен пакетами (см. рис. 5.1). Каждая строка – это один пакет. Пакеты с первого по третий – открытие TCP-соединения, четвертый – запрос HTTP GET, а шестой – ответ на него. Пакеты 7–10 показывают завершение соединения на обоих концах. В колонке Time показано время, прошедшее с захвата первого пакета в секундах. Оно пригодится, например, для анализа задержек из-за тайм-аутов DNS. В нашем случае на это понадобилось менее 3 мс.

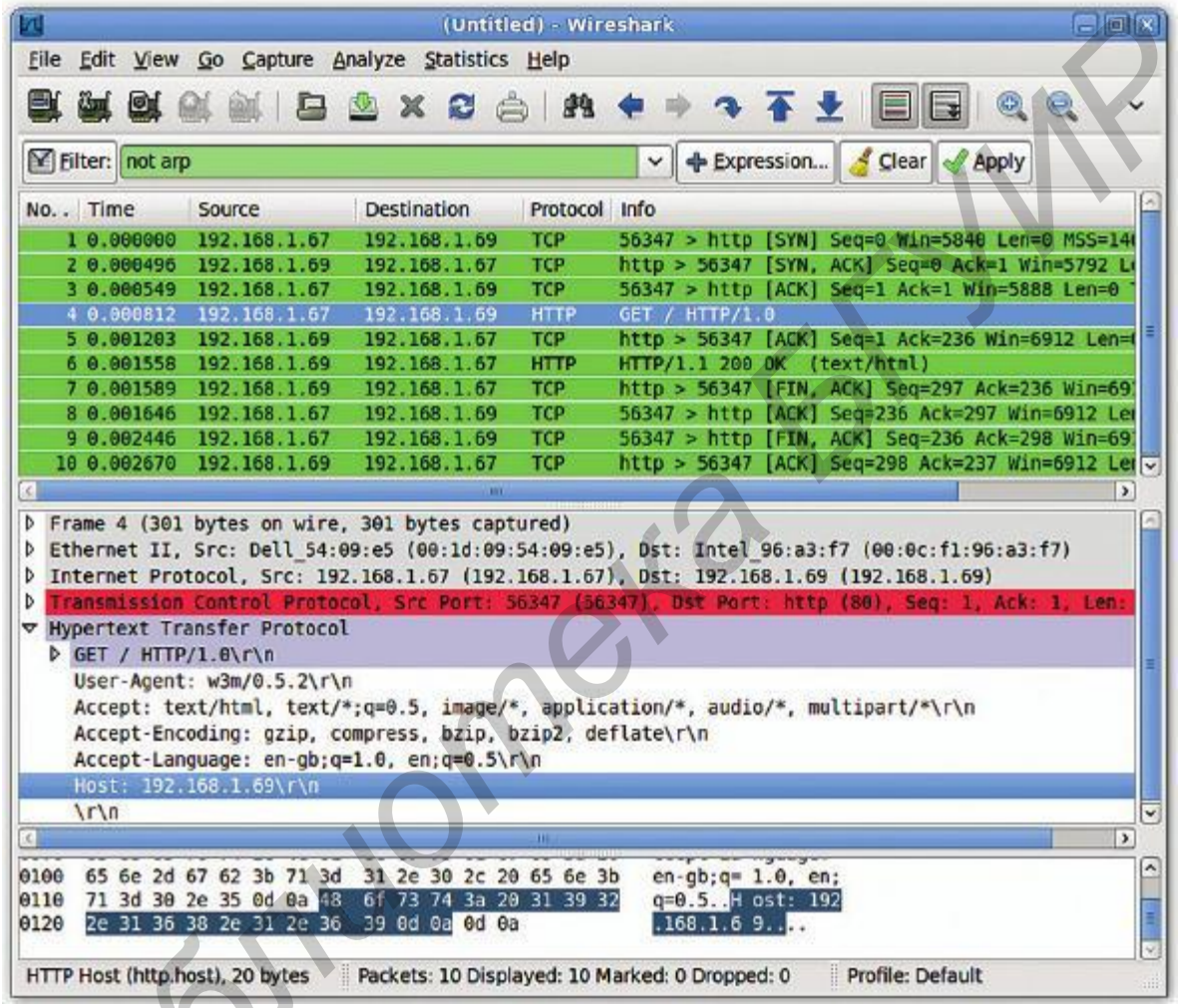

Рис. 5.1. Главное окно Wireshark

Четвертый пакет на рисунке выделен для подробного анализа. В средней панели мы видим общую информацию о заголовках внутри него для каждого уровня стека протоколов. С помощью маленьких стрелок слева можно раскрыть любой уровень, показав его более подробно. Мы сделали это с заголовком уровня приложения – в данном случае это HTTP-пакет. Теперь мы видим, что это запрос HTTP GET, и видим поля заголовка HTTP-запроса.

В нижней части панели отображается содержимое пакета байт за байтом в шестнадцатеричном формате и в ASCII. Подсвеченная часть показывает поле заголовка HTTP, выделенное выше, – в данном случае поле Host.

5.1.6.1. Фильтры

Фильтры – одна из самых мощных возможностей Wireshark. Фильтр – это один или несколько тестов содержимого пакета, позволяющих понять, интересен ли он вам. Фильтрация выполняется в две стадии. Фильтры захвата определяют пакеты, которые будут удержаны в буфере захвата, а фильтры отображения определяют, какие пакеты будут показываться. Язык фильтров богат, и фильтровать можно практически по любому полю любого из протоколов, что позволяет сконцентрироваться на трафике, который вам интересен.

Редактор выражений фильтров поможет просмотреть доступные поля и построить выражения фильтров (рис. 5.2).

Рис. 5.2. Редактор выражений фильтров

Наблюдательные студенты могли заметить, какой фильтр использовался в последнем примере. Он состоит из простого правила not arp и используется, чтобы подавить ARP-рассылку с широкополосного маршрутизатора.

Фильтры могут оперировать с заданными протоколами, такими, как IP, TCP, UDP, ARP и т. д. Они могут выполнять сравнение на равенство и неравенство и числовое сравнение значений полей. Для полей со строковыми значениями можно анализировать подстроки с помощью оператора contains и проверять их на соответствие регулярным выражениям оператором matches. Можно даже сравнить содержимое заданной части пакета с помощью синтаксиса [смещение:длина]. Отдельные условия можно объединить операторами and, not и or. Все это образует универсальный и мощный язык фильтров.

В Wireshark есть графическая утилита, помогающая создавать фильтры отображения. Чтобы запустить ее, нажмите кнопку Expression на панели инструментов Filter главного окна. С его помощью определяется фильтр для отображения только тех пакетов, которые содержат код ответа HTTP 404 («Файл не найден»). Результирующее правило фильтрации: http.response.code == 404.

Полей, по которым можно фильтровать, множество, а как узнать их имена? При перемещении по пакету в главном окне Wireshark имя поля выбранного элемента отображается в строке состояния. Имена полей можно использовать в выражениях фильтров. Вглядевшись в строку состояния первого экранного снимка, вы увидите, что выбранному на панели выше полю соответствует http.host. Полный список имеется на сайте http://www.wireshark.org/docs/dfref.

В двух табл. 5.5 и 5.6 далее показано несколько примеров фильтров захвата и отображения, способных дать представление об их возможностях.

Таблица 5.5

Фильтры захвата

Описание цели захвата	**Описание фильтра**
Только трафик, идущий от или к заданному IP	host 192.168.1.44
Только трафик, идущий от или к заданной подсети	net 192.168.1.0/24
Только DNS-трафик (порт 53)	port 53
Все, кроме ARP и DNS	port not 53 and not arp

Таблица 5.6

Фильтры отображения

Описание цели просмотра	**Описание фильтра**
Только трафик между машинами в локальной подсети	ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16
Только трафик от MAC-адресов устройств Dell	eth.addr[0:3]==00:06:5B
Только HTTP-запросы с URI, заканчивающимися на foo	http.request.uri matches “foo$”
Трафик, имеющий отношение к Windows	smb || nbns || dcerpc || nbss || dns

Существуют также фильтры, которые позволяют задать цветовые правила для Wireshark. Редактор цветовых правил поможет создать новые правила (используя тот же синтаксис, что и у фильтров) и цвета для них, а также импортировать или экспортировать набор цветовых правил. Также можно применять пользовательские цветовые правила для подсветки пакетов на фоне другого трафика. Цвет каждого пакета определяется первым соответствующим ему правилом (рис. 5.3). На сайте http://www.wireshark.org есть набор готовых цветовых правил, с которых можно начать.

В окне параметров захвата можно указать интерфейс, пакеты с которого будут захватываться, переключить его в «неразборчивый» режим, задать фильтр захвата и установить предельные значения для числа пакетов, объема данных или времени.

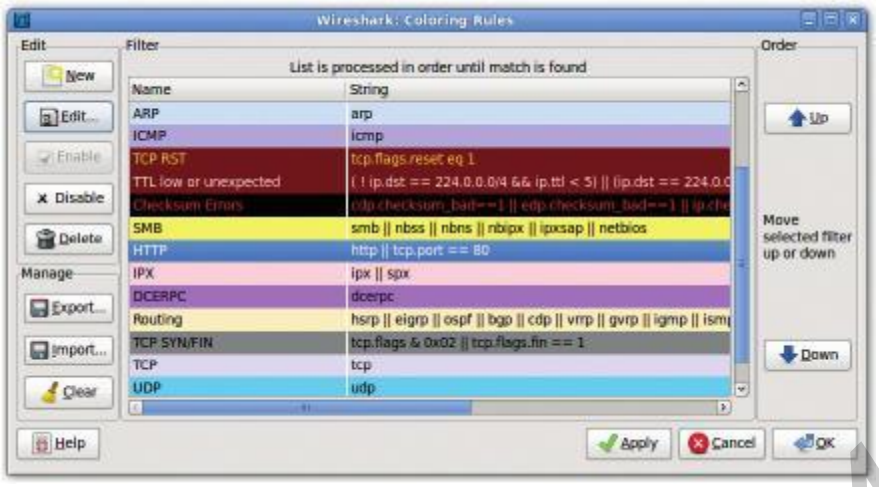

Рис. 5.3. Цветовое правило связывает выражение фильтра с отображаемым цветом

5.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустить две различные ОС (какие именно, уточнить у преподавателя).
2. На обеих ОС установить и сконфигурировать сетевые анализаторы.
3. Поочередно просмотреть весь входящий и исходящий трафик от ваших тестовых машин.
4. На одной из тестовых машин установить ssh или telnet. При желании можно установить и какой-нибудь коммуникатор ICQ, уточнить порт, через который он работает и пронаблюдать весь входящий и исходящий трафик.
5. Просмотреть весь трафик от тестовой машины. Записать его и проанализировать.
6. Полученные результаты конфигурирования в виде скриншотов прикрепить к отчету и обосновать.
7. Полученные анализы сетевого трафика с комментариями прикрепить к отчету.

5.3. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Реализация решения задачи и скриншоты конфигурации сетевых анализаторов.
5. Выводы.

5.4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. На каких уровнях модели OSI находятся сетевые анализаторы?
2. Что такое Ethernet?
3. Почему сетевой протокол Ethernet приобрел большую популярность, чем Token Ring?
4. Что регламентирует Ethernet?
5. Что такое «коммутированная сеть Ethernet»?
6. Что такое TcpDump?
7. Что определяют «выражения» в TcpDump?
8. Какую информацию получит администратор сети, выполнив следующую команду: `TcpDump -n port 23`?
9. Что такое Windump?
10. Что такое Wireshark?
11. Укажите достоинства Wireshark по сравнению с TcpDump.
12. Посредством чего Wireshark может декомпозировать и сортировать пакеты?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Lambert, M. TcpDump / M. Lambert, T. Mariam. – Книга по требованию, 2010. – 148 с.
2. Archibald, N. Nessus, Snort, & Ethereal Power Tools: Customizing Open Source Security Applications / N. Archibald, G. Ramirez. – Книга по требованию, 2005. – 445 с.

Св. план 2014, поз. 12

Учебное издание

Логин Владимир Михайлович
Цырельчук Игорь Николаевич
Хорошко Виталий Викторович

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ БЕЗОПАСНОСТИ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *М. В. Гуртатовская*

Подписано в печать 16.02.2015. Формат 60х84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 3,84. Уч.-изд. л. 4,0. Тираж 50 экз. Заказ 215.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6