Кубанский государственный технологический университет

Кафедра защиты информации

### **КУРСОВАЯ РАБОТА**

По дисциплине:

языки программирования

На тему:

**Сетевые сканеры и анализаторы**

Краснодар 2008

**ВВЕДЕНИЕ**

Данный курсовой проект является обзорной статьей о передаче информации в локальных сетях и методах её отслеживания (нахождения). Собранный мною материал предназначен для опытных пользователей, работающих с локальными сетями, в частности – для сетевых администраторов.

**Сканирование**

Цель сканирования — выявить открытые "окна" и "двери". В предварительно собранной информации содержатся сведения об адресах подсетей и отдельных компьютеров, полученных с помощью запросов whois и переноса зоны.

Информация, собранная на этом этапе, очень ценна для взломщика, поскольку содержит такие данные, как имена и фамилии сотрудников, номера телефонов, диапазоны IP-адресов, адреса DNS-серверов и почтовых серверов. Теперь можно приступать к выявлению тех компьютеров, которые подключены к сети и достижимы из Internet. Для этого будут использоваться разнообразные средства и приемы, такие как ping-прослушивание, сканирование портов и различные методы, позволяющие автоматизировать выполнение этих задач.

Необходимо отметить, что факт наличия IP-адреса в перенесенной зоне еще не означает, что к соответствующему узлу можно получить доступ через Internet. Необходимо проверить каждый конкретный компьютер в отдельности и выяснить, подключен ли он к Internet и имеются ли на нем порты, находящиеся в состоянии ожидания запросов. Нам приходилось встречать немало неправильно настроенных DNS-серверов, которые предоставляли всем желающим адреса обслуживаемых ими частных сетей (например, 10.10.10.0). Поскольку такие адреса не маршрутизируются по Internet, вы понапрасну будете тратить время, пытаясь связаться с ними. Более подробная информация о том, какие адреса являются маршрутизируемыми, приведена в RFC 1918(http://www.ietf.org/rfc/rfcl918.txt).

Теперь давайте перейдем ко второму этапу сбора информации — сканированию.

Прослушивание сети с помощью утилиты ping

Одним из основных этапов в определении структуры сети является ее автоматизированное прослушивание с помощью утилиты ping по диапазону IP-адресов или адресам подсетей. Цель такого прослушивания — определить, имеется ли у отдельных компьютеров подключение к Internet. Утилита ping отправляет пакеты ICMP ECHO (тип 8) указанному компьютеру и ожидает ответного пакета ICMP ECHO\_REPLY (тип 0). Получение такого ответа говорит о том, что компьютер в данный момент подключен к Internet. Хотя при некоторой настойчивости с помощью утилиты ping можно определить количество постоянно подключенных к Internet компьютеров в небольшой и даже средней сети, ручной перебор сетевых адресов будет малоэффективен, если необходимо обследовать корпоративную сеть крупной организации.

Для выполнения ping-прослушивания можно воспользоваться любым из .G® \*-» многочисленных средств, разработанных как для системы UNIX, так и для i-""""" Windows NT. В мире UNIX одним из самых надежных и проверенных методов такого прослушивания является использование утилиты fping (http:/ packetstorm security.com/Exploit\_Code\_Archive/fping.tar.gz). В отличие от других подобных утилит, которые перед переходе к тестированию следующего компьютера ожидают ответа на ранее посланный запрос, утилита fping рассылает все запросы одновременно, а затем ожидает ответа сразу от всех узлов. Именно поэтому утилита fping обеспечивает гораздо более высокую скорость прослушивания большого диапазона IP-адресов, чем обычная утилита ping. Утилита fping была написана специально для использования в сценариях оболочек совместно с утилитой gping (http://www.hacking-exposed.com/tools/tools.html), которая входит в пакет, распространяемый вместе с fping. Утилита gping генерирует список IP-адресов, передаваемых на вход fping для прослушивания. Листинг использования утилиты gping для сетей класса А, В или С может показаться слегка непонятным, поэтому рассмотрим его подробнее.

[tsunami ]'$ gping

usage: gping aO aN bO bN cO cN dO dN

gping a bO bN cO cN dO dN

gping a b cO cN dO dN

gping a b с dO dN

gping abcd

В качестве параметров утилите gping необходимо передать диапазон IP-адресов. На основании этого диапазона будет генерироваться листинг, в котором адреса перебираются друг за другом. Каждый октет передаваемого IP-адреса должен отделяться от остальных пробелами. Например, если мы собираемся генерировать IP-адреса для сети класса С, нам необходимо просто добавить 254 в качестве последнего параметра. Это позволит утилите перебрать все адреса от 192.168.1.1 до 192.168.1.254. Предположим, что эта сеть не содержит подсетей и использует маску подсети 255.255.255.0. Кроме того, мы не будем проверять адрес самой сети 192.168.1.0 и адрес широковещательной рассылки 192.168.1.255. Следует избегать применения утилиты ping к адресам широковещательной рассылки, поскольку это может привести к отказу, или так называемому состоянию DoS (denial of service), если ответный пакет одновременно будет сгенерирован многимиузлами (более подробная информация о том, как установить маску подсети узла, приведена в документации по запросам ICMP). С использованием утилиты gping можно сгенерировать перечень адресов, которые затем будут использоваться утилитой fping.

[tsunami] gping 192 168 1 1 254

192.168.1.1

192.168.1.2

192.168.1.3

192.168.1.4

192.168.1.5

192.168.1.251

192.168.1.252

192.168.1.253

192.168.1.254

Теперь в нашем распоряжении имеется список всех узлов, которые могут находиться в исследуемой сети класса С. Осталось лишь перенаправить вывод утилиты gping на вход утилиты fping, которая выполнит прослушивание сети и определит, какие компьютеры в данный момент подключены к сети.

[tsunami]? gping 192 168 1 1 254 | fping -a

192.168.1.254 is alive

192.168.1.227 is alive

192.168.1.224 is alive

192.168.1.3 is alive

192.168.1.2 is alive

192.168.1.1 is alive

192.168.1.190 is alive

Параметр -а утилиты fping предназначен для включения режима, в котором выводится информация обо всех активных в данный момент компьютерах сети. Если нужно, утилита может выводить и информацию об именах узлов. Этот режим включается с помощью параметра -d. По нашему мнению, параметр -а лучше всего использовать в сценариях оболочки, а параметр -d — при исследовании сети на предмет поиска определенных узлов. Среди других параметров необходимо упомянуть -f, который позволяет вводить адреса из заранее подготовленного файла, а также -h, с помощью которого можно получить перечень всех параметров утилиты и режимов их использования. Еще одной утилитой, о которой мы будем много говорить в этой книге, является утилита nmap, созданная хакером по имени Федор (Fyodor) (www.insecure.org/nmap). Более подробно эта утилита будет рассматриваться ниже в этой главе, однако будет нелишним упомянуть, что, кроме всех остальных возможностей, данная утилита также позволяет выполнить прослушивание сети. Для включения соответствующего режима необходимо указать параметр -SP.

[tsunami] nmap -sP 192.168.1.0/24

Starting nmap V. 2.53 by fyodor@insecure.org

(www.insecure.org/nmap/)

Host (192.168.1.0) seems to be a subnet broadcast

address (returned 3 extra pings).

Host (192.168.1.1) appears to be up.

Host (192.168.1.10) appears to be up.

Host (192.168.1.11) appears to be up.

Host (192.168.1.15) appears to be up.

Host (192.168.1.20) appears to be up.

Host (192.168.1.50) appears to be up.

Host (192.168.1.101) appears to be up.

Host (192.168.1.102) appears to be up.

Host (192.168.1.255) seems to be a subnet broadcast

address (returned 3 extra pings).

Nmap run completed — 256

IP addresses (10 hosts up) scanned in 21 seconds

Что касается приверженцев Windows, они также не остались без внимания. В частности, имеется такая бесплатная утилита, как Finger (рис. 1.1), написанная хакерами из группы Rhino9 (http://www.nmrc.org/files/snt/). Эта утилита является одной из самых быстрых в своем классе. Как и fping, утилита Finger одновременно рассылает несколько ICMP-пакетов ECHO, а затем ожидает поступления ответов. Кроме того, Finger позволяет также получать имена узлов и сохранять результаты своей работы в файле. Такой же скоростью, как и Finger, обладает коммерческий продукт Ping Sweep, предлагаемый компанией SolarWinds (www.solarwinds.net). Поразительная скорость работы Ping Sweep объясняется тем, что данная программа позволяет устанавливать время задержки между передаваемыми пакетами (delay time). Установив это значение равным 0 или 1, можно просканировать всю сеть класса С и получить имена ее узлов менее чем за 7 секунд. Однако при использовании этих средств соблюдайте осторожность, поскольку в этом случае можно значительно снизить пропускную способность какого-нибудь низкоскоростного канала, например канала ISDN с пропускной способностью 128 Кбит/с или Frame Relay (не говоря уже о спутниковом или инфракрасном канале).

Среди других утилит Windows, предназначенных для прослушивания сети, можно отметить WS\_Ping ProPack (www.ipswitch.com) и Netscan Tools (www.nwpsw.com). Хотя возможностей этих утилит вполне достаточно для прослушивания небольших сетей, они значительно медленнее Finger и Ping Sweep. Кроме того, не забывайте, что утилиты с графическим интерфейсом, несмотря на удобство их использования, лишают вас возможности их применения в сценариях и автоматизированных процедурах.

Возможно, вы хотите спросить, как поступать, если исследуемый узел блокирует сообщения ICMP? Хороший вопрос. Такой подход зачастую применяется на тех узлах, администраторы которых заботятся о безопасности. Однако, несмотря на блокировку пакетов ICMP, существуют дополнительные средства и методы, позволяющие определить, подключен ли такой узел к сети или нет. Вместе с тем необходимо отметить, что все эти средства оказываются не такими точными и эффективными, как обычные утилиты семейства ping.

В тех случаях, когда обмен данными по протоколу ICMP заблокирован, в первую очередь применяется метод сканирования портов (port scanning), который более подробно рассматривается ниже в этой главе. Просканировав стандартные порты каждого потенциального IP-адреса сети, можно определить, какие узлы подключены к сети. Если порт открыт (opened mode) или находится в режиме ожидания (listening mode), значит, по данному адресу находится подключенный к Internet узел сети. Недостатками этого метода являются большие временные затраты и некоторая неопределенность результата (если по какому-то адресу не удалось обнаружить ни одного порта, то это еще не означает, что соответствующий узел не подключен к Internet). Одной из утилит, которые можно использовать для сканирования портов, является nmap. Как уже упоминалось, с помощью этой утилиты можно проводить ICMP-прослушивание, однако этим перечень ее возможностей далеко не исчерпывается. В частности, эта утилита позволяет выполнять так называемое TCP-прослушивание сканированием (TCP ping scan). Данный режим включается с помощью параметра -рт и указания номера порта, например 80. Выбор порта с номером 80 обусловлен тем, что в подавляющем большинстве случаев именно он используется узлами сети для обмена данными через пограничные маршрутизаторы или брандмауэры с компьютерами, расположенными в так называемой демилитаризованной зоне (DMZ — demilitarized zone). При использовании указанного параметра утилита рассылает узлам исследуемой сети пакеты АСК, а затем ожидает поступления пакетов RST, что свидетельствует о том, что узел подключен к Internet.

[tsunami] nmap -sP -PT80 192.168.1.0/24

TCP probe port is 80

Starting nmap V. 2.53

Host (192.168.1.0) appears to be up.

Host (192.168.1.1) appears to be up.

Host shadow (192.168.1.10) appears to be up.

Host (192.168.1.11) appears to be up.

Host (192.168.1.15) appears to be up.

Host (192.168.1.20) appears to be up.

Host (192.168.1.50) appears to be up.

Host (192.168.1.101) appears to be up.

Host (192.168.1.102) appears to be up.

Host (192.168.1.255) appears to be up.

Nmap run completed (10 hosts up) scanned in 5 seconds

Как видно из приведенного выше листинга, этот метод определения подключенных к Internet узлов очень эффективен, даже если на них блокируется передача пакетов ICMP. С помощью утилиты nmap имеет смысл провести несколько подобных проверок, тестируя такие стандартные порты как SMTP (25), POP (110), AUTH (110), IMАР (143) или другие порты, которые, по вашим сведениям, могут быть уникальными на каком-либо компьютере исследуемой сети.

Еще одной утилитой, специально предназначенной для TCP-прослушивания, является утилита hping (http://www.kyuzz.org/antirez/). По возможностям она даже превосходит утилиту nmap. Утилита hping позволяет пользователям управлять параметрами протокола TCP, что может обеспечить проникновение отправляемых пакетов даже через некоторые устройства управления доступом. Так, установив порт назначения с помощью параметра -р, можно обойти некоторые устройства управления доступом точно так же, как это было сделано с применением утилиты traceroute в главе 1. Поэтому утилита hping может с успехом служить не только для TCP-прослушивания, но и преодолевать преграды некоторых устройств управления доступом благодаря возможности фрагментации пакетов.

[tsunami] hping 192.168.1.2 -S -p 80 -f

HPING 192.168.1.2 (ethO 192.168.1.2):

S set, 40 data bytes

60 bytes from 192.168.1.2:

flags=SA seq=0 ttl=124 id=17501 win=0 time=46.5

60 bytes from 192.168.1.2:

flags=SA seq=l ttl=124 id=18013 win=0 time=169.1

В некоторых случаях простые устройства управления доступом не могут корректно обрабатывать фрагментированные пакеты, что позволяет им проходить через такие устройства и достигать интересующего взломщика адреса. Обратите внимание, что в случае, когда порт открыт, возвращаются флаги TCP SYN (s) и дек (А). Утилиту hping очень легко использовать в сценариях оболочки с параметром счетчика пакетов -cN, где N — это количество пакетов, которые нужно отправить в Internet, прежде чем переходить к выполнению следующей команды сценария. Хотя данный метод и не обладает такой скоростью, как описанные выше методы ICMP-прослушивания, в некоторых случаях только он может помочь выяснить конфигурацию сети. Более подробно утилита hping рассматривается в главе 11, "Брандмауэры".

Последним из средств прослушивания рассмотрим утилиту icmpenum хакером Симплом Номадом (Simple Nomad) (http://www.nmrc.org/files/sunix/icmpenum-1.1.tgz). Эту утилиту удобно использовать для определения архитектуры сети. Утилита icmpenum позволяет быстро выявить подключенные к сети компьютеры, передавая стандартные ICMP-пакеты ECHO, а также ICMP-запросы TIME STAMP REQUEST и INFO. Если входные пакеты ECHO не пропускаются пограничным маршрутизатором или брандмауэром, то подключенные узлы можно по-прежнему идентифицировать с помощью альтернативных пакетов ICMP.

[shadow] icmpenum -i2 -с 192.168.1.0

192.168.1.1 is up

192.168.1.10 is up

192.168.1.11 is up

192.168.1.15 is up

192,168.1.20 is up

192.168.1.103 is up

В приведенном примере сеть класса С (192.168.1.0) была протестирована с использованием ICMP-запроса TIME STAMP REQUEST. Однако реальная мощь утилиты icmpenum заключается в возможности идентификации узлов с помощью ложных пакетов, что позволяет избежать обнаружения злоумышленника. Это возможно благодаря тому, что утилита icmpenum позволяет генерировать ложные пакеты с использованием параметра -s и пассивно ожидать отклика при указании параметра -р.

Подводя итог, можно отметить, что IСМР- или TCP-прослушивание позволяет точно установить, какие компьютеры сети подключены к Internet. Так, в рассматриваемом примере мы установили, что из 255 потенциальных адресов сети класса С к Internet подключены лишь несколько компьютеров. Выявленные узлы становятся предметом первоочередного внимания в дальнейших исследованиях. Таким образом, мы значительно сузили область поиска, что позволяет сэкономить время и силы для более эффективных действий.

**Идентификация запущенных TCP- и UDP-служб**

Использование хорошей утилиты сканирования портов — важнейший этап сбора информации об исследуемой сети. Хотя для этих целей существует много различных программ, ориентированных как на платформу UNIX, так и на платформу Windows NT, мы ограничимся рассмотрением лишь самых популярных и проверенных временем сканеров.

strobe

Утилита strobe — это общепризнанный и популярный TCP-сканер портов, написанный Джулианом Ассанжем (Julian Assange) (ftp://ftp.FreeBSD.org/pub/ FreeBSD/ports/distfiles/strobe-l.06.tgz). Она стала известной уже довольно давно и, вне всякого сомнения, считается одной из самых быстрых и надежных утилит этого класса. К основным возможностям утилиты strobe относится оптимизация системных и сетевых ресурсов, а также сканирование исследуемой системы с максимальной эффективностью. Помимо высокой эффективности, утилита strobe версии 1.04 и выше может собирать идентификационные маркеры (если, конечно, они имеются), связанные с каждым проверяемым портом. Эта информация может оказаться полезной при определении операционной системы, а также запущенных на компьютере службах. В данных, выводимых утилитой strobe, имеется информация о каждом прослушанном порте TCP.

[tsunami] strobe 192.168.1.10

strobe 1.03 © 1995 Julian Assange (proff@suburbia.net).

192.168.1.10 echo 7/tcp Echo [95,JBP]

192.168.1.10 discard 9/tcp Discard [94,JBP]

192.168.1.10 sunrpc 111/tcp rpcbind SUN RFC

192.168.1.10 daytime 13/tcp Daytime [93,JBP]

192.168.1.10 chargen 19/tcp ttytst source

192.168.1.10 ftp 21/tcp File Transfer [Control] [96,JBP]

192.168.1.10 exec 512/tcp remote process execution;

192.168.1.10 login . 513/tcp remote login a la telnet;

192.168.1.10 cmd 514/tcp shell like exec, but automatic

192.168.1.10 ssh 22/tcp Secure Shell

192.168.1.10 telnet 23,/tcp Telnet [112,JBP]

192.168.1.10 smtp 25/tcp Simple Mail Transfer [102,JBP]

192.168.1.10 nfs 2049/tcp networked file system

192.168.1.10 lockd 4045/tcp

192.168.1.10 unknown 32772/tcp unassigned

192.168.1.10 unknown 32773/tcp unassigned

192.168.1.10 unknown 32778/tcp unassigned

192.168.1.10 unknown 32799/tcp unassigned

192.168.1.10 unknown 32804/tcp unassigned

Хотя в большинстве случаев утилита strobe предоставляет точные данные, все же важно помнить о некоторых ее ограничениях. Во-первых, данная утилита выполняет TCP-сканирование, не поддерживая сканирование по протоколу UDP. Поэтому в некоторых случаях можно получить лишь половину требуемой информации. Во-вторых, при соединении с каждым портом утилита strobe выполняет лишь TCP-сканирование подключением. Хотя именно этим и объясняется высокая надежность получаемых результатов, в то же время использование утилиты strobe очень легко выявить на исследуемой системе. Поэтому необходимо рассмотреть и другие утилиты сканирования, лишенные указанных недостатков.

udp\_scan

Для UDP-сканирования, которого не выполняет strobe, можно воспользоваться утилитой udp\_scan, которая изначально входила в пакет SATAN (Security Administrator Tool for Analyzing Networks), написанный Дэном Фармером (Dan Farmer) и Вайетсом Венема (Wietse Venema) в 1995 году. Хотя сам пакет SATAN несколько устарел, входящие в его состав утилиты по-прежнему можно использовать. Кроме того, по адресу http://wwdsilx.wwdsi.com можно получить новую версию пакета SATAN, которая теперь называется SAINT. Несмотря на наличие множества других утилит UDP-сканирования, мы пришли к выводу, что udp\_scan — одна из самых надежных утилит, позволяющая получать достоверные результаты. Правда, необходимо сказать также о том, что, несмотря на высокую надежность утилиты udp\_scan, у нее имеется и один существенный недостаток. Эта утилита не может противостоять контратаке какого-либо из пакетов IDS, который осведомлен о методах, используемых в пакете SATAN для сканирования портов. Таким образом, если сканирование необходимо выполнить более скрытно, поишите какое-нибудь другое средство. Обычно с помощью утилиты udp\_scan проверяются порты с номерами, меньшими 1024, а также некоторые определенные порты с большими номерами.

;tsunami] udp\_scan 192.168.1.1 1-1024

-: 2 : UNKNOWN: 5 3:UNKNOWN: 123:UNKNOWN: 135:UNKNOWN:

netcat

Еще одной прекрасной утилитой является netcat (или nc), написанная Хоббитом (Hobbit, hobbit@avian.org). Эта утилита может выполнять так много различных задач, что была названа нами "швейцарским армейским ножом". Помимо остальных возможностей, о которых мы еще не раз будем говорить на протяжении всей книги, утилита nc позволяет применять основные методы TCP- и UDP-сканирования. Степенью детализации выводимых данных можно управлять с помощью параметров -v и -vv, которые включают, соответственно, режимы подробного и очень подробного отображения результатов. Параметр -z применяется для включения режима нулевого ввода-вывода (zero mode I/O), используемого для сканировании портов, а параметр -..2 позволяет задать для каждого соединения интервал ожидания. По умолчанию утилита пс выполняет TCP-сканирование, а для UDP-сканирования необходимо использовать параметр -и (как показано во втором примере).

tsunami] nc -v -z -w2 192.168.1.1 1-140

192.168.1.1] 139 (?) open

192.168.1.1] 135 (?) open

192.168.1.1] 110 (pop-3) open

[192.168.1.1] 106 (?) open

[192.168.1.1] 81 (?) open

[192.168.1.1] 80 (http) open

[192.168.1.1] 79 (finger) open

[192.168.1.1] 53 (domain) open

[192.168.1.1] 42 (?) open

[192.168.1.1] 25 (smtp) open

[192.168.1.1] 21 (ftp) open

[tsunami] nc -u -v -z -w2 192.168.1.1 1-140

[192.168.1.1] 135 (ntportmap) open

[192.168.1.1] 123 (ntp) open

[192.168.1.1] 53 (domain) open

[192.168.1.1] 42 (name) open

nmap

Рассмотрев простейшие средства сканирования портов, давайте перейдем к обсуждению возможностей безусловного лидера этой категории — утилиты nmap. Данная утилита, разработанная Федором (Fyodor) (http://www.insecure.org/nmap), обладает не только базовыми возможностями TCP- и UDP-сканирования, но и поддерживает все остальные упоминавшиеся выше методы. Очень редко можно найти утилиту, которая предоставляла бы столь богатый набор возможностей в одном пакете. Итак, запустим утилиту и посмотрим, какие возможности она предоставляет.

[tsunami]# nmap -h

nmap V. 2.53 Использование:

nmap [Тип(ы) сканирования] [Параметры]

<Список узлов или подсетей>

Некоторые стандартные типы сканирования

(При использовании параметров, отмеченных символом '\*',

требуются привилегии root)

-sT TCP-сканирование подключением

(устанавливается по умолчанию) \* -sS TCP-сканирование

с помощью сообщений SYN (среди всех методов TCP-сканирования является наилучшим)

\* -sU UDP-сканирование

-sP ping-прослушивание

(выполняется поиск всех достижимых узлов)

\* -sF,-sX,-sN сканирование с помощью сообщений FIN,

по методу "рождественской елки" и

нуль-сканирование,

соответственно (рекомендуется использовать только опытным пользователям)

-SR/-I сканирование с использованием демона

RPC/identd (применяется совместно с другими типами сканирования)

Некоторые стандартные параметры

(являются необязательными, могут комбинироваться друг с другом):

\* -О режим изучения пакетов TCP/IP

с целью определения типа удаленной операционной системы

-р <диапазон> — диапазон портов,

которые будут сканироваться.

Пример диапазона: '1-1024,1080,6666,31337'

-F Выполняется сканирование портов,

перечисленных в файле /etc/services

-v Режим вывода подробной информации.

Рекомендуется всегда использовать этот параметр.

Для включения режима вывода очень

подробной информации используйте параметр -vv

-РО Отключение проверки активности узла с помощью утилиты ping (применяется для сканирования таких узлов, как www.microsoft.com и аналогичных)

\* -Ddecoy\_hostl,decoy2[,...]

Скрытое сканирование с указанием нескольких ложных адресов узлов

-Т <Paranoici I Sneaky|Polite|Normal[Aggressive IInsane>

Принятая политика ожидания отклика от удаленного узла

-n/-R Никогда не выполнять разрешение имен DNS/

Всегда выполнять [по умолчанию: имена разрешаются

при необходимости] -oN/-oM <logfile>

Вывести результаты сканирования в файл <logfile> в

удобочитаемом/машинном формате -iL <inputfile>

Взять IP-адреса или имена узлов из файла <inputfiie>.

Для использования стандартного потока ввода stdin укажите '-'

\* -S <your\_IP>/-e <devicename>

позволяет указать исходный IP-адрес или устройство

--переход в интерактивный режим

(затем для получения справки нужно нажать клавишу h)

[tsunami] nmap -sS 192.168.1.1

Starting nmap V. 2.53 by fyodor@insecure.org

Interesting ports on (192.168.1.11):

(The 1504 ports scanned but

not shown below are in state: closed)

Port State Protocol Service

21 open tcp ftp

25 open tcp smtp

42 open tcp nameserver

53 open tcp domain

79 open tcp finger

80 open tcp http

81 open tcp hosts2-ns

106 open tcp popSpw

110 open tcp pop-3

135 open tcp loc-srv

139 open tcp netbios-ssn

443 open tcp https

Помимо вышеуказанных, утилита nmap предоставляет и другие полезные возможности, заслуживающие детального обсуждения. Так, в приведенном выше примере мы ис-ПОЛЬЗОЕИЛИ параметры командной строки, при которых осуществлялось сканирование одного узла. Однако утилита nmap с такой же легкостью позволяет сканировать и всю сеть. Как легко заметить, nmap поддерживает описания диапазонов адресов в нотации CIDR (Classless Inter-Domain Routing — бесклассовая маршрутизация доменов Internet), описанной в RFC 1519 (среди всех методов TCP-сканирования является наилучшим)

\* -sU UDP-сканирование

-sP ping-прослушивание (выполняется поиск всех достижимых узлов)

\* -sF,-sX,-sN сканирование с помощью сообщений FIN,

по методу "рождественской елки" и нуль-сканирование, соответственно (рекомендуется использовать только опытным пользователям)

-SR/-I сканирование с использованием демона RPC/identd (применяется совместно с другими типами сканирования)

Некоторые стандартные параметры (являются необязательными, могут комбинироваться друг с другом):

\* -О режим изучения пакетов TCP/IP

с целью определения типа удаленной операционной системы

-р <диапазон> — диапазон портов, которые будут сканироваться.

Пример диапазона: '1-1024,1080,6666,31337'

-F Выполняется сканирование портов, перечисленных в файле /etc/services

-v Режим вывода подробной информации.

Рекомендуется всегда использовать этот параметр.

Для включения режима вывода очень подробной информации используйте параметр -vv

-РО Отключение проверки активности узла с помощью утилиты ping (применяется для сканирования таких узлов, как www.microsoft.com и аналогичных)

\* -Ddecoy\_hostl,decoy2[,...]

Скрытое сканирование с указанием нескольких ложных адресов узлов

-Т <Paranoici I Sneaky|Polite|Normal[Aggressive IInsane>

Принятая политика ожидания отклика от удаленного узла

-n/-R Никогда не выполнять разрешение имен DNS/

Всегда выполнять [по умолчанию: имена разрешаются при необходимости] -oN/-oM <logfile>

Вывести результаты сканирования в файл <logfile> в удобочитаемом/машинном формате -iL <inputfile>

Взять IP-адреса или имена узлов из файла <inputfiie>.

Для использования стандартного потока ввода stdin укажите '-'

\* -S <your\_IP>/-e <devicename>

позволяет указать исходный IP-адрес или устройство --переход в интерактивный режим (затем для получения справки нужно нажать клавишу h)

[tsunami] nmap -sS 192.168.1.1

Starting nmap V. 2.53 by fyodor@insecure.org

Interesting ports on (192.168.1.11):

(The 1504 ports scanned but

not shown below are in state: closed)

Port State Protocol Service

21 open tcp ftp

25 open tcp smtp

42 open tcp nameserver

53 open tcp domain

79 open tcp finger

80 open tcp http

81 open tcp hosts2-ns

106 open tcp popSpw

110 open tcp pop-3

135 open tcp loc-srv

139 open tcp netbios-ssn

443 open tcp https

Помимо вышеуказанных, утилита nmap предоставляет и другие полезные возможности, заслуживающие детального обсуждения. Так, в приведенном выше примере мы использовали параметры командной строки, при которых осуществлялось сканирование одного узла. Однако утилита nmap с такой же легкостью позволяет сканировать и всю сеть. Как легко заметить, nmap поддерживает описания диапазонов адресов в нотации CIDR (Classless Inter-Domain Routing — бесклассовая маршрутизация доменов Internet), описанной в RFC 1519 (http://www.ietf.org/rfc/rfcl519.txt). В этом формате очень легко задавать диапазоны адресов вида 192.168.1.1-192.168.1.254. Полученную информацию можно сохранить в обычном текстовом файле с помощью параметра -о. При указании параметра -oN результаты будут сохранены в удобочитаемом формате.

[tsunar.i]# ranap -sF 192.168.1.0/24 -oN outfile

Если выводимые данные нужно сохранить в файле, в котором в качестве разделителей используются символы табуляции (например, чтобы впоследствии программно анализировать полученную информацию), используйте параметр -оМ. В любом случае при сканировании сети, скорее всего, будет получено очень много информации, поэтому имеет смысл сохранить результаты в любом из форматов. В некоторых случаях целесообразно сохранять их сразу в обоих форматах, используя как параметр -ON, так и -оM.

Предположим, что после сбора предварительных данных о сети организации мы пришли к выводу, что в качестве основного брандмауэра в ней используется простое устройство, выполняющее фильтрацию пакетов. В этом случае можно воспользоваться параметром -f утилиты nmap, чтобы включить режим фрагментации пакетов. Очевидно, что это приведет к отделению заголовков TCP-пакетов от самих пакетов, что затруднит для устройств управления доступом или систем IDS возможность выявления попытки сканирования. В большинстве случаев современные устройства фильтрации пакетов и брандмауэры прикладного уровня, прежде чем осуществлять анализ пакетов IP, помещают все фрагменты в очередь. Однако при использовании более старых моделей устройств управления доступом или устройств, в которых соответствующие функции были отключены для повышения производительности, дефрагментация не выполняется и пакеты передаются дальше во внутреннюю сеть в том виде, в котором они поступают.

Если архитектура системы безопасности исследуемой сети и ее узлов была хорошо продумана, то эта система без особого труда выявит сканирование, осуществляемое с помощью приведенных выше примеров. Для таких случаев утилита nmap предоставляет дополнительные возможности маскирования, предназначенные для заполнения системных журналов исследуемого узла избыточной информацией. Данный режим включается с помощью параметра -D. Главная идея данного подхода состоит в том, чтобы во время выполнения реального сканирования создать видимость одновременного сканирования из других указанных в командной строке адресов. Для того чтобы воспрепятствовать такому сканированию, системе безопасности исследуемого узла придется проверить все записи, чтобы выяснить, какие из полученных IP-адресов источников сканирования являются реальными, а какие — фиктивными. При использовании данного метода нужно удостовериться в том, что IP-адреса, выступающие в качестве маскировочных, принадлежат реальным узлам, которые в момент сканирования подключены к Internet. В противном случае исследуемая система будет не в состоянии обработать все сообщения SYN, в результате чего возникнет условие DoS.

[tsunami] nmap -sS 192.168.1.1 -D 10.1.1.1

www.target\_web.com,ME -p25,139,443

Starting nmap V. 2.53 by fyodor@insecure.org

Interesting ports on (192.168.1.1):

Port State ' Protocol Service

25 open tcp smtp

443 open tcp https

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

В приведенном примере параметры, введенные в командной строке утилиты nmap, обеспечивают сканирование в режиме, затрудняющем обнаружение реального адреса сканирующего узла.

Еще одним полезным методом является сканирование с целью идентификации запущенных процессов (подробнее о нем говорится в RFC 1413, http://www.ieff.org/rfc/rfcl413.txt). Этот тип сканирования, называемый ident-сканированием, предназначен для определения пользователя путем установления TCP-соединения с портом 113. Очень часто в ответ приходит сообщение, содержащее идентификатор владельца процесса, связанного с данным портом. Однако этот метод годится лишь для исследования систем UNIX.

[tsunami] nmap -I 192.168.1.10

Starting nmap V. 2.53 by fyodor@insecure.org

Port State Protocol Service Owner

22 open tcp ssh root

25 open tcp smtp root

80 open tcp http root

110 open tcp pop-3 root

113 open tcp auth root

6000 open tcp Xll root

В приведенном выше фрагменте показано, как идентифицируются владельцы всех обнаруженных процессов. Опытный читатель должен обратить внимание на то, что Web-сервер принадлежит не пользователю nobody, как это должно быть в соответствии с элементарными правилами обеспечения безопасности, а пользователю root, что является вопиющим нарушением. Выполнив идентификацию процессов и установив такой интересный факт, можно заключить, что взломщик, которому удастся проникнуть через систему защиты Web-сервера, получит полный контроль над данным компьютером.

Последний метод, на котором мы остановимся, называется сканированием с прорывом по FTP (FTP bounce scanning). Этот метод впервые был описан Хоббитом (Hobbit). В своей статье, опубликованной в электронном бюллетене Bugtraq в 1995 году, он осветил некоторые скрытые недостатки протокола FTP (RFC 959, http://www.ietf.org/rfc/rfc0959.txt). Кратко данный метод можно описать как скрытное подключение через FTP-сервер, используя поддержку proxy-серверов, реализованную на этом FTP-сервере. Как отмечает Хоббит в вышеупомянутой статье, прорыв по FTP "можно использовать практически для неотслеживаемой отправки электронной почты и сообщений в группы новостей, взлома серверов различных сетей, заполнения диска, попыток прорыва через брандмауэры и другой вредоносной деятельности, которая при этом может оставаться практически незамеченной". Добавим, что с помощью прорыва по FTP можно сканировать порты, чтобы скрыть свой адрес, и, что еще более важно, обходить устройства управления доступом.

Конечно, утилита nmap поддерживает и этот режим сканирования (параметр -b). Однако для его выполнения необходимо соблюдение нескольких условий. Во-первых, на FTP-сервере должен быть каталог, доступный для чтения/записи со стороны любого пользователя, например /incoming. Во-вторых, FTP-сервер должен принять от утилиты nmap заведомо неправильную информацию о порте с помощью команды PORT. Хотя этот метод очень эффективен для проникновения через устройства управления доступом, а также для сокрытия своего адреса, у него есть один существенный недостаток — слишком низкая скорость работы. Кроме того, многие современные FTP-серверы просто запрещают выполнение таких операций.

Однако применение различных средств для сканирования портов — это только половина задачи. Теперь нужно разобраться с тем, как проанализировать данные, полученные с помощью каждой из утилит. Независимо от применяемого средства, необходимо идентифицировать открытые порты, поскольку их перечень позволит определить операционную систему удаленного узла. Например, если на узле открыты порты 135 и 139, то, скорее всего, этот узел работает под управлением операционной системы Windows NT. Обычно Windows NT опрашивает порты 135 и 139, тогда как Windows 95/98 — лишь порт 139.

Например, изучив результаты, полученные во время работы утилиты strobe, которая рассматривалась выше в этой главе, можно заключить, что исследовавшийся в рассматриваемом примере узел работает под управлением операционной системы из семейства UNIX. Данный вывод можно сделать на основании того, что на исследуемом узле открыты порты с номерами 111 (portmapper), 512-514 (службы Berkley R), 2049 (NFS), а также порты с номерами 3277Х, что характерно именно для систем семейства UNIX. Более того, можно также предположить, что данная операционная система относится к семейству Solaris — именно этой системе присуще использование служб RPC вместе с портам из этого диапазона. Нужно подчеркнуть, что это лишь предположения, поскольку в действительности установленная операционная система, если с ее настройкой поработал опытный администратор безопасности, может лишь "выдавать себя" за Solaris, а в действительности не иметь с ней ничего общего.

Итак, после завершения TCP- и (или) UDP-сканирования портов уже можно выдвинуть предположения о типе операционной системы, работающей на исследуемом узле, и, следовательно, о том, как можно проникнуть на этот узел. Например, если на сервере Windows NT открыт порт 139, то такой узел подвергается очень высокой степени риска. Подробнее о скрытых недостатках системы защиты Windows NT, а также о том, как с помощью порта 139 можно проникнуть в систему, в которой не приняты адекватные контрмеры для защиты этого порта, рассказывается в главе 5.

Рассматривавшаяся в качестве примера система UNIX, скорее всего, также подвергается большому риску, поскольку выявленные нами работающие службы предоставляют в распоряжение удачливого взломщика очень большие возможности. Например, использование служб удаленного вызова процедур (RPC — Remote Procedure Call) и поддержки сетевой файловой системы (NFS — Network File System) являются двумя основными методами проникновения через систему защиты сервера UNIX (подробнее см. главу 8).

С другой стороны, если служба RPC не находится в режиме ожидания запросов, то проникнуть через ее систему защиты практически невозможно. Именно поэтому так важно помнить, что чем больше служб работает на компьютере, тем большему риску он подвергается.

**Сканирование с помощью утилит**

Утилита cheops (произносится "ки-опс", http://www.marko.net/cheops/), изображенная на рис. 1.2, представляет собой программу с графическим интерфейсом, предназначенную для полномасштабного исследования сети. При этом в одном пакете объединены утилиты ping, traceroute, средства сканирования портов, а также определения типа операционной системы (с помощью queso). Кроме этого, cheops позволяет получить графическое схематическое изображение исследуемой сети и связанных с ней сетей, что значительно облегчает понимание ее архитектуры.

Вторая утилита, на которой мы остановимся, называется tkined и входит в состав пакета Scotty (http://wwwhome.cs.utwente.nl/~schoenw/scotty/). По существу, эта утилита является редактором сетевого редактирования, написанным на языке Tel, который, объединяет различные средства управления сетью, обеспечивающие сбор всевозможной информации об архитектуре и работе сети. Утилита tkined обладает большой гибкостью и позволяет проводить исследование сети с представлением результатов в графической форме. Кроме утилиты tkined, в состав пакета Scotty входит немало других интересных средств, заслуживающих самого пристального внимания.

**Контрмеры: защита от средств автоматического сбора информации**

Поскольку в автоматизированных средствах, подобных Scotty, tkined и cheops, объединены приемы, о которых рассказывалось выше в данной главе, для защиты от них можно применять контрмеры, аналогичные тем, которые применяются для соответствующих средств сбора информации, сканирования и т.д.

**Пассивное исследование стека**

Основные принципы пассивного исследования стека аналогичны концепциям, лежащим в основе его активного исследования. Однако в данном случае вместо передачи пакетов для определения используемой операционной системы взломщик осуществляет мониторинг сетевого трафика. Таким образом, наблюдая за сетевым трафиком между различными компьютерами, можно определить тип и версию удаленной операционной системы. Большие исследования в этой области были проведены Ланцом Спитзнером (Lance Spitzner). На их основе была написана книга, которую можно найти по адресу http://www.enteract.com/~lspitz/finger.html. Кроме ТОГО, по адресу http://www.subterrain.net/projects/siphon можно также найти утилиту siphon, предназначенную для пассивного исследования портов и идентификации операционной системы. Теперь познакомимся с тем, как же выполняется пассивное исследование стека.

Параметры, используемые для пассивного исследования стека

Для определения типа и версии операционной системы можно использовать самые разнообразные признаки. Однако сейчас мы ограничимся рассмотрением лишь нескольких атрибутов, связанных с сеансом сетевого взаимодействия с помощью протокола TCP/IP.

* атрибутов TTL (Time-to-Live — время жизни). Устанавливает ли операционная система значение TTL для исходящих пакетов?
* Windows Size (размер окна). Какой размер окна используется?
* DF (Don't Fragment — бит фрагментации). Устанавливается ли операционной системой признак DF?
* TOS (Type-of-Service — тип службы). Устанавливается ли операционной системой значение TOS. Если да, то какое?

Проанализировав каждый из атрибутов и сравнив полученные результаты со значениями из имеющейся базы данных, можно определить удаленную операционную систему. Поскольку этот метод не гарантирует получения правильного ответа на основе каждого из атрибутов в отдельности, для получения более надежных результатов атрибуты можно комбинировать. Именно такой подход и используется утилитой siphon.

Вот как работает описанный метод. Если с помощью утилиты telnet установить удаленное соединение между узлами 192 .168 .1.10 и 192 .168 .1.11 , то с использованием утилиты siphon можно определить тип удаленной операционной системы.

[shadow]# telnet 192.168.1.11

С помощью нашей любимой утилиты snort можно частично просмотреть пакеты, передаваемые в процессе сетевого взаимодействия.

06/04-11:23:48.297976 192.168.1.11:23 -> 192.168.1.10:2295

TCP TTL:255 TOS:OxO 10:58934 DF

\*\*S\*\*\*A\* Seq: OxD3B709A4 Ack: OxBE09B2B7 Win: 0x2798

TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS: 1460

При этом видно, что упоминавшиеся выше атрибуты принимают следующие значения:

* TTL= 255
* Размер окна = 2798
* Бит DF = Yes A TOS = 0

Теперь обратимся к базе данных утилиты siphon — файлу osprints.conf:

[shadow]f grep -i Solaris osprints.conf

# Window:TTL:DF:Operating System DF = 1 for ON, 0 for OFF.

2328:255:l:Solaris 2.6 - 2.7

2238:255:l:Solaris 2.6 - 2.7

2400:255:l:Solaris 2.6 - 2.7

2798:255:l:Solaris 2.6 - 2.7

FE88:255:l:Solaris 2.6 - 2.7

87CO:255:l:Solaris 2.6 - 2.7

FAFO:255:0:Solaris 2.6 - 2.7

FFFF:255:1:Solaris 2.6-2.7

Из приведенного фрагмента видно, что в четвертой записи содержатся те же значения, которые были получены с использованием утилиты snort. Таким образом, с помощью утилиты siphon можно точно определить исследуемую операционную систему.

[crush]# siphon -v -i xl0 -о fingerprint.out

Running on: 'crush' running FreeBSD 4.0-RELEASE on a(n) i386

Using Device: xlO

Host Port . TTL DF Operating System 192.168.1.11 23 255 ON Solaris 2.6 - 2.7

Итак, в качестве удаленной была определена система Solaris 2.6. И это .не вызвало особых проблем. Пассивное исследование стека взломщик может использовать для выбора потенциальных жертв. Для этого достаточно понаблюдать за соответствующим Web-узлом и проанализировать сетевой трафик либо воспользоваться утилитой siphon. Несмотря на то что описанный метод является достаточно эффективным, он все же имеет некоторые ограничения. Во-первых, в приложениях, генерирующих свои собственные пакеты (например, шпар), не применяются те же признаки, что и самой операционной системой. Поэтому полученные результаты могут оказаться неточными. Во-вторых, на удаленном узле можно без проблем изменить атрибуты соединения.

Solaris: ndd -set /dev/ip ip\_def\_ttl 'число'

Linux: echo 'число' > /proc/sys/net/ipv4/ip\_default\_ttl

NT:

HKEY\_LOCAL\_MACHINESystemCurrentControlSet

ServicesTcpipParameters

**Активное исследование стека**

Прежде чем перейти к рассмотрению возможностей утилит nmap и queso, необходимо вкратце пояснить, в чем же состоит суть исследования стека TCP/IP. Исследование стека (stack fingerprinting) — это очень мощная технология, позволяющая быстро определить тип и версию операционной системы узла с высокой степенью вероятности.

Очевидно, что разные разработчики по-разному подходят к реализации стека TCP/IP. В частности, многие разработчики по-своему трактуют рекомендации документов RFC, что впоследствии проявляется в логике работы тех или иных сетевых служб. Таким образом, зная о существующих различиях и проверив реакцию служб изучаемой системы на различные ситуации, можно практически однозначно определить тип и версию соответствующей операционной системы. Для достижения максимальной достоверности при исследовании стека требуется по крайней мере один порт, находящийся в режиме ожидания запросов. С помощью утилиты nmap можно выдвинуть предположение об используемой операционной системе даже при отсутствии таких портов, однако степень его достоверности в этом случае будет невысокой. Полное описание процесса исследования стека можно найти в статье Федора (Fyodor), впервые публикованной в журнале Phrack Magazine.

Ниже приведен перечень тестов, которые можно использовать в процессе исследования стека для определения типа и версии операционной системы.

* Передача пакетов FIN (FIN probe). Пакет FIN отсылается в открытый порт. Как уже упоминалось, согласно документу RFC 793 исследуемая система не должна отвечать на такое сообщение. Однако многие реализации стека (например, Windows NT) отвечают на них, отправляя пакет FIN/ACK.
* Попытка установки флагов (bogus flag probe). Отсылается пакет SYN с установленным флагом в заголовке TCP, значение которого не определено спецификацией протокола. Некоторые операционные системы, например Linux, в ответном пакете устанавливают этот же флаг.
* Изучение начальной последовательности (Initial Sequence Number (ISN) sampling). Основная задача этого теста — попытаться определить характерные признаки начальной последовательности, генерируемой узлом при получении запроса на установку соединения, которые характерны для той или иной реализации TCP.
* Мониторинг бита фрагментации ("don't fragment bit" monitoring). Этот бит устанавливается некоторыми операционными системами для повышения производительности. Проверка данного бита может помочь в определении типа операционной системы, для которой характерно такое поведение.
* Исходный размер окна TCP (TCP initial window size). Для некоторых реализаций стека протоколов TCP/IP данный параметр уникален, что способствует точности определения типа операционной системы.
* Значение АСК (дек value). В различных реализациях стека IP по-разному задается значение поля АСК. В одних случаях возвращается полученный от вас номер последовательности, а в других — значение номера последовательности, увеличенное на 1.
* Обработка сообщений об ошибках ICMP (ICMP error message quenching). Некоторые операционные системы следуют рекомендациям документа RFC 1812 (www. ietf.org/rfc/rfcl812.txt) и ограничивают скорость передачи сообщений об ошибках. Поэтому, отправляя UDP-пакеты на какой-либо порт (обычно с большим номером), вполне реально измерить количество сообщений об ошибках, поступившее за определенный период, и определить таким образом тип операционной системы.
* Измерение длины сообщений ICMP (ICMP message quoting). При возникновении ошибок ICMP разными операционными системами передаются сообщения различной длины. Проанализировав полученное сообщение, можно сделать некоторые предположения об исследуемой операционной системе.
* Проверка целостности ответных сообщений об ошибках ICMP (ICMP error message-echoing integrity). В некоторых реализациях стека используется изменение заголовка IP при возврате сообщений об ошибках ICMP. Проверив тип изменений, внесенных в заголовок, можно сделать некоторые предположения об операционной системе исследуемого узла.
* Тип службы (TOS — type of service). Можно проверять поле TOS для сообщений "ICMP port unreachable" (порт недоступен). В большинстве реализаций это поле имеет значение 0, однако иногда используются и другие значения.
* Обработка фрагментации (fragmentation handling). Как отмечают Томас Пташек (Thomas Ptacek) и Тим Ньюсхам (Tim Newsham) в своей известной статье Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection (http://www. clark.net/~roesch/idspaper.html), различные стеки обрабатывают перекрывающиеся сообщения по-разному. При сборке фрагментированньк пакетов некоторые стеки записывают новые данные поверх старых и наоборот. Проверив, каким образом были собраны тестовые пакеты, можно сделать предположение об исследуемой операционной системе.
* Параметры TCP (TCP options). Параметры TCP определены в документе RFC 793 и недавно изданном RFC 1323 (www.ietf.org/rfc/rfcl323.txt). Нововведения, описанные в RFC 1323, нашли отражение только в самых последних реализациях стеков. Отправляя пакет с набором различных параметров, таких как по operation, maximum segment size, window scale factor, timestamp и так далее, можно сделать вывод о типе и версии операционной системы.

Для того чтобы воспользоваться утилитой nmap и выполнить все перечисленные тесты (за исключением обработки фрагментации и обработки сообщений об ошибках ICMP), достаточно указать в командной строке параметр -о. Давайте посмотрим, как будет выглядеть полученный результат.

[tsunami] nmap -0192.168.1.10

Starting nmap V. 2.53 by fyodor@insecure.org

Interesting ports on shadow (192.168.1.10) :

Port State Protocol Service

7 open tcp echo

9 open tcp discard

13 open tcp daytime

19 open tcp chargen

21 open tcp ftp

22 open tcp ssh

23 open tcp telnet

25 open tcp smtp

37 open tcp time

111 open tcp sunrpc

512 open tcp exec

513 open tcp login

514 open tcp shell

2049 open tcp nfs

4045 open tcp lockd

TCP Sequence Prediction: Class=randorn positive increments

Difficulty=26590 (Worthy challenge)

Remote operating system guess: Solaris 2.5, 2.51

Как видно, при включении режима исследования стека утилиты nmap можно легко получить достаточно точное определение типа и версии операционной системы. Даже если на изучаемом узле не открыто ни одного порта, утилита nmap поможет сделать довольно точное предположение об используемой операционной системе.

[ tsunami]# nmap -p80 -0 10.10.10.10

Starting nmap V. 2.53 by fyodor@insecure.org

Warning: No ports found open on this machine,

OS detection will be

MUCH less reliable

No ports open for host (10.10.10.10)

Remote OS guesses: Linux 2.0.27 - 2.0.30, Linux 2.0.32-34,

Linux 2.0.35-36, Linux 2.1.24 PowerPC,

Linux 2.1.76, Linux 2.1.91 - 2.1.103, Linux 2.1.122 - 2.1.132; 2.2.0-prel - 2.2.2, Linux 2.2.0-pre6 - 2.2.2-ac5

Nmap run completed -- 1 IP

address (1 host up) scanned in 1 second

Как видно из приведенного листинга, утилита nmap даже без открытых портов, правильно определила операционную систему Linux.

Одной из примечательных особенностей утилиты nmap является то, что листинг сигнатур хранится в отдельном файле с именем nmap-os-fingerprints. При появлении каждой новой версии утилиты этот файл также обновляется, и на момент написания данной книги в нем содержались сотни сигнатур. Если вы хотите добавить новые сигнатуры и повысить таким образом эффективность утилиты nmap, обратитесь ПО адресу http://www.insecure.org:80/cgi-bin/nmap-submit.cgi.

Хотя на момент написания данной книги утилита nmap, по-видимому, позволяет наиболее точно выполнить исследование стека TCP/IP, она, тем не менее, является далеко не первой программой, в которой реализована соответствующая технология. До того как Федор встроил в утилиту nmap средства определения операционной системы, для этих же целей уже была создана утилита queso (http://www.apostols.org/projectz/). Необходимо отметить, что утилита queso не позволяет выполнять сканирование портов и может определять тип операционной системы только при наличии в исследуемой системе открытого порта (по умолчанию используется порт 80). Если порт 80 закрыт, необходимо задать другой открытый порт, как показано в следующем примере, в котором с помощью утилиты queso осуществляется попытка определить тип операционной системы через порт 25.

[tsunami] queso 10.10.10.20:25

10.10.10.20:25 \* Windoze 95/98/NT

Контрмеры: защита от определения операционной системы

**Обнаружение попыток определения операционной системы**

Многие из упоминавшихся выше средств выявления сканирования с успехом могут служить и для обнаружения попыток определения типа операционной системы, Хотя они не проинформируют вас о том, что выполнялось специальное сканирование с помощью утилиты nmap или queso, с их помощью все же удастся распознать сам факт такого особого сканирования, например с установкой флага SYN.

**Предупреждение попыток определения операционной системы**

Хотелось бы посоветовать какое-нибудь средство, позволяющее противодействовать попыткам определения операционной системы, однако, к сожалению, вынуждены констатировать, что решить эту проблему весьма непросто. Конечно, можно изменить исходный код операционной системы (естественно, если он имеется в вашем распоряжении) или поменять ее параметры, влияющие на характеристики стека, однако такое вмешательство может значительно изменить функциональность ОС. Например, в системе FreeBSD 4.x имеется параметр ядра TCP\_DROP\_SYNFIN, который можно применить для игнорирования пакетов SYN+FIN, используемых утилитой nmap в целях исследования стека. Установка этого параметра поможет пресечь попытки определения типа операционной системы, однако в то же время нарушит поддержку RFC 1644 (TCP Extensions for Transactions).

Вместо этого мы предлагаем создавать такие сети, в которых сканированию могли бы подвергнуться лишь надежные и хорошо защищенные proxy-серверы и брандмауэры, а не компьютеры внутренней сети. В этом случае, даже если взломщику и удастся разведать тип операционной системы того или иного узла, проникновение через устройства защиты будет значительно затруднено.

**Пассивное определение операционной системы**

Из предыдущих разделов видно, насколько эффективными оказываются средства активного исследования стека, такие как утилиты nmap и queso. Важно не забывать о том, что рассмотренные выше приемы являются активными по своей природе. При этом для определения специфических особенностей сетевого стека и используемой операционной системы каждому узлу нужно передавать тестовые пакеты. Поскольку все активные методы предполагают передачу пакетов, идентификации операционной системы. Другими словами, активное исследование является далеко не самым скрытым методом, к которому может прибегнуть взломщик.

**Типы сканирования**

Прежде чем перейти к описанию конкретных средств, используемых для сканирования портов, необходимо уделить немного времени обзору методов сканирования, известных в настоящее время. Одним из пионеров реализации различных методов сканирования является ранее упоминавшийся Федор (Fyodor). Многочисленные приемы сканирования были реализованы им в утилите nmар. Многие из описанных в данной книге методов сканирования были предложены самим Федором.

TCP-сканирование подключением (TCP connect scan). При таком типе сканирования осуществляется попытка подключения по протоколу TCP к интересующему нас порту с прохождением полной процедуры согласования параметров (handshake), состоящей в обмене сообщениями SYN, SYN/ACK и АСК. Попытки такого сканирования очень легко выявляются. На рис. 1.3 показана диаграмма обмена сообщениями в процессе согласования параметров.

 TCP-сканирование с помощью сообщений SYN (TCP SYN scan). Этот метод называется также сканированием с незавершенным открытием сеанса (half-open scanning), так как при его использовании полное TCP-соединение не устанавливается. Вместо этого на исследуемый порт отправляется сообщение SYN. Если в ответ поступает сообщение SYN/ACK, это означает, что данный порт находится в состоянии LISTENING. Если же ответ приходит в виде сообщения RST/ACK, то, как правило, это говорит о том, что исследуемый порт отключен. Получив ответ, компьютер, выполняющий сканирование, отправляет исследуемому узлу сообщение RST/ACK, поэтому полное соединение не устанавливается. Этот метод обеспечивает более высокую скрытность по сравнению с полным подключением. Многие системы не регистрируют такие попытки, поэтому они довольно часто могут оставаться незамеченными.

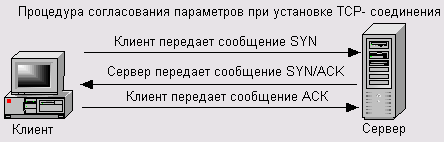


Рис. 1.3. При установке TCP-соединения происходит обмен тремя сообщениями: (1) клиент отправляет серверу пакет SYN, (2) получает от сервера пакет SYN/ACK и (3) отправляет серверу пакет АСК

* TCP-сканирование с помощью сообщений FIN (TCP FIN scan). В этом случае исследуемой системе отправляется пакет FIN. Согласно документу RFC 793 (http://www.ieff.org/rfc/rfc0793.txt), в ответ узел должен отправить пакет RST для всех закрытых портов. Данный метод срабатывает только для стека протоколов TCP/IP, реализованного в системе UNIX.
* TCP-сканирование по методу "рождественской елки" (TCP Xmax Tree scan). При использовании данного метода на исследуемый порт отправляются пакеты FIN, URG и PUSH. Согласно документу RFC 793, исследуемый узел в ответ должен отправить сообщения RST для всех закрытых портов.
* TCP нуль-сканирование (TCP Null scan). Этот метод состоит в отправке пакетов с отключенными флагами. Согласно RFC 793, исследуемый узел должен ответить отправкой сообщения RST для всех закрытых портов.
* TCP-сканирование с помощью сообщений АСК (TCP ACK scan). Этот метод позволяет получить набор правил, используемых брандмауэром. Такое сканирование поможет определить, выполняет ли брандмауэр простую фильтрацию пакетов лишь определенных соединений (пакетов с установленным флагом АСК) или обеспечивает расширенную фильтрацию поступающих пакетов.
* TCP-сканирование размера окна (TCP Windows scan). Такой метод позволяет выявить открытые, а также фильтруемые/нефильтруемые порты некоторых систем (например, AIX и FreeBSD), в зависимости от полученного размера окна протокола TCP.
* TCP-сканирование портов RFC (TCP RFC scan). Этот метод применим только для систем UNIX и используется для выявления портов RFC (Remote Procedure Call — удаленный вызов процедур), связанных с ними программ и их версий.

А UDP-сканирование (UDP scan). Данный метод заключается в отправке на исследуемый узел пакетов по протоколу UDP. Если в ответ поступает сообщение о том, что порт ICMP недоступен (ICMP port unreachabie), это означает, что соответствующий порт закрыт. С другой стороны, если такого сообщения нет, можно предположить, что данный порт открыт. В связи с тем, что протокол UDP не гарантирует доставки, точность данного метода очень сильно зависит от множества факторов, влияющих на использование системных и сетевых ресурсов. Кроме того, UDP-сканирование — очень медленный процесс, что особенно сказывается при попытках сканирования устройств, в которых реализован мощный алгоритм фильтрации пакетов. Поэтому, планируя использовать UDP-сканирование, приготовьтесь к тому, что результаты могут оказаться ненадежными.

Некоторые реализации IP-протокола обладают одним неприятным свойством: пакеты RST отправляются обратно для всех сканируемых портов независимо от того, находятся ли соответствующие порты в режиме ожидания запросов. Учитывайте этот факт при использовании описанных методов. Однако в то же время сканирование подключением и сканирование с использованием сообщений SYN могут применяться для всех узлов.

**Утилиты сканирования портов для системы Windows**

В предыдущих разделах были рассмотрены утилиты сканирования портов с точки зрения пользователя UNIX, однако неужели не существует аналогичных средств, доступных для пользователей Windows? Конечно же, это не так. Следующие утилиты сканирования портов являются лучшими среди подобных средств, поскольку обладают высокой скоростью, точностью и широким набором функциональных возможностей.

NetScanTools Pro 2000

Одним из наиболее универсальных средств исследования сетей, доступных в настоящее время, является пакет NetScanTools Pro 2000 (NSTP2K), содержащий самые разнообразные утилиты, объединенные общим интерфейсом. С использованием NSTP2K можно генерировать запросы DNS, включая nslookup, dig и axf r, запросы whois, осуществлять ping-прослушивание, сканировать таблицы имен NetBIOS, отслеживать сообщения SNMP и выполнять многие другие задачи. Более того, с использованием пакета NetScanTools Pro 2000 можно выполнять несколько задач одновременно. Например, можно выполнять сканирование портов одной сети и осуществлять ping-прослушивание другой сети (хотя мы не можем ручаться за правильность таких действий по отношению к большим сетям).

В состав пакета NetScanTools Pro 2000 включен также один из лучших сканеров портов Windows. Все необходимые параметры можно установить во вкладке Port Probe. К преимуществам утилиты сканирования NSTP2K. можно отнести возможность гибкого задания параметров исследуемых узлов и портов (и IP-адреса и список портов могут быть импортированы из текстовых файлов), возможность TCP- и UDP-сканирования (хотя соответствующие режимы нельзя установить отдельно для каждого порта), а также высокую скорость благодаря реализации многопоточности. К недостаткам утилиты сканирования пакета NSTP2K можно отнести некоторую громоздкость получаемых результатов, что затрудняет их анализ с помощью сценариев, и, кроме того, графический интерфейс делает невозможным применение этой утилиты в сценариях. Нам хотелось бы высказать следующее пожелание: было бы очень удобно, чтобы результаты, полученные с использованием одной утилиты пакета NSTP2K (скажем, NetScanner), можно было бы напрямую передавать другой утилите (например, Port Probe).

В общем, пакет NSTP2K (http://www.nwpsw.com) представляет собой профессионально разработанный программный продукт, который регулярно обновляется посредством сервисных пакетов, однако все же остается несколько дорогостоящим по сравнению с предоставляемыми им возможностями. Можно также познакомиться с менее робастной версией Netscan Tools (в настоящее время доступна версия 4), являющейся пробной 30-дневной версией пакета NSTP2K. Однако предоставляемые ею возможности не столь широки, как у пакета Pro 2000 (например, она не позволяет выполнять UDP-сканирование).

При использовании пакета NSTP2K не забудьте отключить сервер идентификации во вкладке IDENT Server, чтобы не запрещать прослушивание порта 113.

SuperScan

Утилита SuperScan, написанная Робином Кейром (Robin Keir) (http: //members. home.com/rkeir/software.html), является еще одной быстрой и гибкой утилитой TCP-сканирования портов и имеет гораздо более привлекательную стоимость — она распространяется бесплатно! Как и пакет NSTP2K, утилита SuperScan позволяет гибко задавать перечень IP-адресов исследуемых узлов и сканируемых портов. Особенно удобно использовать режим Extract from file. Лучше всего особенности его применения описаны в справочной системе. Вот небольшой фрагмент из справочной системы утилиты SuperScan, из которого видно, что она позволяет сэкономить значительную часть времени.

"Режим [The "Extract from file" feature scans] позволяет просматривать содержимое любого текстового файла и извлекать из него корректные IP-адреса и имена узлов. При поиске корректных имен программой выполняются достаточно интеллектуальные действия. Однако перед обработкой файла из него нужно удалить потенциально неоднозначные фрагменты текста, воспользовавшись внешним текстовым редактором. На кнопках Browse и Extract можно щелкать столько раз, сколько различных файлов имеется в вашем распоряжении. При этом в список имен исследуемых узлов программой будут добавлены все новые имена. Все повторяющиеся элементы будут автоматически удалены. После нахождения всех имен узлов щелкните на кнопке Resolve, чтобы преобразовать их в числовые IP-адреса и выполнить подготовку к этапу сканирования портов." Эта утилита предоставляет также один из наиболее обширных списков портов, с которым нам когда-либо приходилось встречаться. (Авторам книги нравится список под названием henss.lst. Более того, в исходном англоязычном названии книги первые буквы составляют аббревиатуру HENSS&S, откуда можно заключить, что авторы — просто фанаты этого списка.) Кроме того, порты можно выделить или отменить их выделение вручную. Не лишний раз повторить, что утилита SuperScan помимо всех перечисленных возможностей обладает также и высокой скоростью.

NTOScanner

Утилита NTOScanner компании NTObjectives, Inc. (http://www.ntobjectives.com) представляет собой быструю программу TCP-сканирования с графическим интерфейсом, которая при ручной установке соответствующего режима позволяет также собирать маркеры портов, находящихся в состоянии ожидания запросов. Однако эта утилита имеет некоторые неудобства при задании исследуемых узлов и портов, а также требует, чтобы перед сканированием сетей класса С к узлам было применено ping-прослушивание. Утилита NTOScanner чрезвычайно полезна для определения служб, запущенных на конкретном узле или в сети, достижимой с помощью ICMP-пакетов.

WinScan

Утилита WinScan компании Scan Mathias of Prosolve (http://www.prosolve.com) является свободно распространяемой программой TCP-сканирования портов, реализованной в двух версиях: с графическим интерфейсом (winscan.exe) и для использования в командной строке (scan.exe). Мы регулярно обращаемся к версии для командной строки в файлах сценариев, поскольку при сканировании сетей класса С она позволяет получить удобные для анализа результаты. При использовании утилит Win32 strings, tee и tr компании Mortice Kern Systems, Inc. (http://www.mks.com) следующая консольная команда NT будет выполнять сканирование сети для портов из диапазона 0-1023 и формировать результат в виде строк с полями, разделенными двоеточиями, в формате IP-адрес:имя\_службы: порт/протокол (для облегчения восприятия строка была разделена на две части).

scan.exe -n 192.168.7.0 -s 0 -е 1023 -f|strings|findstr/с:"/top"|

 tr \011\040 : | tr -s ::|tee -ia results.txt

Параметр -f при медленных соединениях лучше не использовать, поскольку полученные результаты могут оказаться не очень надежными. При запуске приведенной выше команды будут получены примерно следующие данные.

192.168.22.5:nbsession:139/tcp

192.168.22.16:nbsession:139/tcp

192.168.22.32:nbsession:139/tcp

Большое спасибо Патрику Хейму (Patrick Heim) и Ясону Глассбергу (Jason Glassberg) за предоставление этой интересной команды.

ipEye

Не думаете ли вы, что для выполнения нетрадиционного сканирования потребуется система Linux и утилита nmap? He торопитесь с выводами, поскольку утилита ipEye Арни Видстрома (Arne Vidstrom)'(http: //ntsecurity.nu) позволяет выполнить сканирование требуемых портов, а также TCP-сканирование с помощью сообщений SYN, FIN и с использованием метода "рождественской елки", из командной строки Windows. На использование этой прекрасной утилиты накладывается лишь несколько ограничений. Они заключаются в том, что ее можно использовать только в среде Windows 2000 и одновременно сканировать один узел. Вот пример запуска утилиты ipEye для выполнения TCP-сканирования с помощью сообщений SYN порта с номером 20. При этом предпринята попытка избежать правил фильтрации, используемых маршрутизатором. Приведенный пример аналогичен запуску утилиты nmap с параметром -д (для краткости полученные результаты отредактированы).

С:\Toolbox>ipeye.exe

192.168.234.110 -syn -p 1 1023 -sp 20

ipEye 1.1 - (с) 2000, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)

- http://ntsecurity.nu/toolbox/ipeye/

1-52 [closed or reject]

53 [open]

54-87 [closed or reject]

88 [open]

89-134 [closed or reject]

135 [open]

136-138 [closed or reject]

139 [open]

636 [open]

637-1023 [closed or reject]

1024-65535 [not scanned]

WUPS

Утилита Windows UDP Port Scanner (WUPS) разработана теми же авторами (http://ntsecurity.nu). Эта утилита представляет собой надежное, с графическим интерфейсом и относительно быстрое средство сканирования UDP-портов, несмотря на то, что позволяет одновременно сканировать заданную последовательность портов лишь одного узла. Как видно из рис. 2.6, утилита WUPS является надежным средством для быстрого UDP-сканирования каждого требуемого узла и, следовательно, значительно облегчает выполнение этой утомительной задачи.

**Защита от сканирования портов**

В табл. 1.4 приведен перечень различных популярных утилит сканирования, а также типы сканирования, которые эти утилиты позволяют выполнять.

**Контрмеры: защита от сканирования портов**

Выявление факта сканирования

Как правило, взломщики прибегают к сканированию TCP- и UDP-портов удаленного компьютера, чтобы установить, какие из них находятся в состоянии ожидания запросов. Поэтому выявить факт сканирования — значит, установить, в каком месте и кем будет предпринята попытка взлома. Основные методы выявления факта сканирования состоят либо в использовании специальной программы, предназначенной для выявления вторжений на уровне сети (IDS), такой как NFR, либо в использовании механизма защиты на уровне отдельного узла.

Таблица 1.4. Популярные утилиты сканирования портов и их возможности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Утилита | Сканирование | | | Ресурс |
| UNIX | TCP | UDP | Скрытое |  |
| Strobe | X |  |  | ftp://f tp.FreeBSD.org/  pub/FreeBSD  /ports/distfiles/strobe-1.06.tgz |
| Tcp\_scan | X |  |  | http://wwdsilx.  wwdsi.com/saint/ |
| Udp\_scan |  | X |  | http://wwdsilx.  wwdsi.com/saint/ |
| Nmap | X | X | X | http://www. inscure . org/nmap |
| Netcat | X | X |  | http://www.10pht.com/  users/10pht/nc110.tgz |
| Windows  Netcat | X | X |  | http://www.10pht.com/users/  10pht/ncllnt.zip |
| NetScanTools Pro 2000 | X | X |  | http://www.nwpsw.com |
| SuperScan | X |  |  | http://members.home.com/  rkeir/software.html |
| NTOScanner | X |  |  | http://www.ntobjectives.com |
| WinScan | X |  |  | http://www.prosolve.com |
| IpEye | X |  |  | http://ntsecurity.nu |
| WUPS |  | X |  | http://ntsecurity.nu |
| Fscan | X | X |  | http://www.foundstone.com |

Метод UDP-сканирования утилиты netcat не работает в операционной системе Windows NT, поэтому пользователям этой ОС не стоит доверять полученным результатам.

# Алгоритм выявления факта сканирования

# Автор - Стюарт Мак-Клар (Stuart McClure)

# Данный алгоритм выявляет неудачные попытки сканирования портов,

# которые приводят к генерации сообщений ACK/RST.

В процессе настройки

# можно поэкспериментировать со значениями

# maxcount и maxtime.

port\_schema = library\_schema:new( 1, ["time", "ip", "ip",

"int"],

scope() };

time = 0;

count = 0;

maxcount =2; t Максимально допустимое количество ACK/RST.

maxtime =5; tt Максимально допустимое время, в течение

# которого может достигаться значение maxcount.

source = 0; port = 0; target = 0;

filter portscan ip ( )

{

if (tcp.is)

(

# Проверка АСК, RST и источника поступления.

if ( byte(ip.blob, 13) == 20 )

# Установлены флаги АСК, RST

{

count = count + 1;

source = ip.dest;

target = ip.source;

port = tcp.sport;

time = system.time;

} } on tick = timeout

( sec: maxtime, repeat )

call checkout;

}

func checkout {

if (count >= maxcount) {

echo("Попытка сканирования порта,

Время: ", time, "\n); record system.time, source, target,

port to the\_recorder\_portscan; count = 0; }

else

count = 0;

}

the\_recorder\_portscan=recorder("

bin/histogram packages/sandbox/portscan.cfg",

"port\_schema"

);

Для выявления попыток сканирования портов можно также воспользоваться утилитой snort (www.snort.org) (см. также http://spyjurenet.com/linuxrc.org/projects/ snort/). Как вы могли догадаться, эта утилита является одной из предпочитаемых нами программ IDS (заметим, что ее версии 1.x не позволяют обнаруживать фрагментацию пакетов). Вот пример листинга, содержащего данные о попытке сканирования портов.

[\*\*] spp\_portscan: PORTSCAN DETECTED from 192.168.1.10 [\*\*]

05/22-18:48:53.681227

[\*\*] spp\_portscan: portscan status from 192.168.1.10: 4 connections

across 1 hosts: TCP(O), UDP(4) [\*\*]

05/22-18:49:14.180505

[\*\*] spp\_portscan: End of portscan from 192.168.1.10 [\*\*]

05/22-18:49:34.180236

Для платформы UNIX существует немало утилит, таких, например, как Утилита: scanlogd (http://www.openwall.com/scanlogd/) компании Solar Designer, которые выявляют и регистрируют попытки сканирования. Кроме того, утилиту Psionic PortSentry, созданную в рамках проекта Abacus (http://www.psionic.com/abacus), можно настроить не только на регистрацию, но и на принятие контрмер при выявлении факта активного сканирования. Один из способов борьбы с попытками сканирования портов заключается в автоматической установке для ядра правил фильтрации, когда к уже существующим добавляется новое правило, запрещающее доступ со стороны узла-нарушителя. Такое правило можно задать в конфигурационном файле утилиты PortSentry. При этом одно и то же правило может отличаться в различных системах. Для системы Linux 2.2.x, в которой имеется поддержка ядра брандмауэра, в файл portsentry.conf нужно добавить запись, аналогичную следующей.

# New ipchain support for Linux kernel version 2.102+

KILL\_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -1"

Утилита PortSentry совместима с большинством версий UNIX, включая Solaris. Независимо от того, применяете ли вы какие-либо утилиты или нет, необходимо помнить, что массированные попытки сканирования портов, инициируемые каким-либо узлом или какой-нибудь сетью, могут означать, что кто-то изучает вашу сеть. Всегда обращайте самое пристальное внимание на такие действия, поскольку за ними может последовать полномасштабное вторжение. И наконец, не забывайте о том, что имеется возможность активного противостояния или блокирования попыток сканирования портов. Все эти вопросы рассматриваются в статье, опубликованной компанией Solar Designer (http://www. openwall.com/scanlogd/P53-13.gz). В этой статье содержатся дополнительные советы по разработке и использованию систем выявления попыток сканирования.

Большинство брандмауэров не только могут, но и должны настраиваться на режим обнаружения попыток сканирования. Однако одни брандмауэры справляются с этой задачей лучше, другие хуже. Например, некоторые брандмауэры умеют выявлять скрытое сканирование. Однако многие из них, поддерживая режим выявления SYN-сканирования, абсолютно игнорируют FiN-сканирование. Самой большой проблемой при выявлении факта сканирования является задача анализа огромных системных журналов, накапливаемых при ежедневной работе серверов сети. Для упрощения решения этой задачи можно воспользоваться утилитой Psionic Logcheck (http: //www.-psionic.com/abacus/logcheck/). Кроме того, мы рекомендуем настроить утилиты таким образом, чтобы они реагировали на обнаруженные попытки сканирования в реальном времени, отсылая сообщения по электронной почте. Везде, где это только возможно, устанавливайте пороговые значения для количества регистрируемых событий (threshold logging), чтобы взломщик не завалил ваш почтовый ящик грудой сообщений, в которых будет так же трудно найти информацию, как и в системных журналах. Кроме того, в этом случае может также возникнуть условие DoS. При использовании пороговых значений все предупреждения будут группироваться, а не обрабатываться по одному. Как минимум необходимо настроить систему безопасности на выдачу отчетов о самом факте выявленной попытки сканирования. Для брандмауэра Firewall-1 с этой целью можно использовать утилиту Ланца Спитцнера (Lance Spitzner) alert, sh (http://www. enteract.com/~lspitz/intrusion.html). Эта утилита будет выявлять и отслеживать попытки сканирования портов, работая под управлением Firewall-1 в качестве средства защиты, заданного пользователем.

Для платформы Windows NT также имеется несколько утилит, предназначенных для выявления попыток сканирования. Прежде всего, необходимо отметить такую утилиту, как Genius 2.0, разработанную компанией Independent Software (http://www. indiesoft.com) для платформ Windows 95/98 и Windows NT. Этот программный продукт предоставляет гораздо больше возможностей, чем простое средство обнаружения TCP-сканирования портов. Однако необходимо отметить, что даже с учетом этого, имеет смысл его использовать. Утилита Genius отслеживает многочисленные запросы к открытым портам и при обнаружении попыток сканирования отображает на экране предупреждающее диалоговое окно, в котором содержится IP-адрес взломщика и доменное имя его узла.

Утилита Genius позволяет выявлять как попытки обычного сканирования, т.е. с установлением TCP-соединения, так и SYN-сканирования.

Еще одним детектором сканирования для системы Windows, заслуживающем отдельного упоминания, является программа BlackICE компании Network ICE (http://www.-networkice.com). Данная программа представляет первое основанное на использовании агентов средство выявления вторжений, которое можно использовать как в Windows 9х, так и в NT. В момент написания данной книги этот программный продукт был коммерческим, хотя в ближайшем будущем компания обещает подготовить свободно распространяемую версию. И наконец, программа ZoneAlarm (http://www.zonelabs.com/zonealarm.htm) хорошо подходит для платформы Windows и может применяться в качестве средства IDS на уровне брандмауэра.

**Предотвращение сканирования**

Вряд ли можно помешать кому-либо предпринять попытку сканирования портов на вашем компьютере, однако вполне реально свести к минимуму связанный с этим риск. Для этого нужно заблокировать все службы, в работе которых нет необходимости. В среде UNIX данная задача решается с помощью добавления символов комментария в соответствующие строки файла /etc/inetd. corif, а также отключения автоматического запуска ненужных служб в сценарии начальной загрузки. В системе Windows NT также целесообразно отключить все ненужные службы. Однако сделать это сложнее, поскольку из-за сетевой архитектуры Windows NT по крайней мере Порт 139 должен работать постоянно. Тем не менее, остальные службы можно отключить, запустив аплет Services панели управления. Здесь же стоит упомянуть о том, что компанией Tiny Software (www.tinysoftware.com) распространяется модуль ядра, позволяющий выполнять фильтрацию входящих пакетов. С помощью этого модуля можно защитить большинство важных портов. Что же касается других операционных систем и устройств, то нам остается лишь посоветовать как можно внимательнее прочитать соответствующие справочные руководства. Постарайтесь найти в них информацию о том, какие порты вам действительно необходимы и как отключить остальные, чтобы свести риск к минимуму.

**Активное определение операционной системы**

Итак, мы убедились, что существует множество различных приемов и средств сканирования портов. Вспомните, что при сканировании портов преследуется две основные цели. Во-первых, нужно установить, какие TCP- и UDP-порты на исследуемом компьютере находятся в состоянии ожидания запросов. Во-вторых, необходимо определить тип операционной системы, используемой на удаленном узле. Информация об операционной системе понадобится на последующих этапах, при составлении схемы уязвимых участков. Об этом речь пойдет в последующих главах. Важно помнить, что при этом необходимо быть особенно точным и внимательным к мелочам. Именно поэтому очень важно абсолютно правильно установить тип удаленной операционной системы. При определении типа ОС очень полезной оказывается косвенная информация, получаемая, например с помощью сбора маркеров, о которых мы поговорим в главе 3. При этом будет собрана информация о таких службах, как FTP, telnet, SMTP, HTTP, POP и других. Сбор маркеров — это один из самых простых методов определения типа операционной системы, а также версий работающих под ее управлением служб. Нетрудно догадаться, что существуют различные средства, призванные помочь в решении этой задачи. Среди доступных можно отметить две утилиты, позволяющие получить самые точные результаты, — уже хорошо нам известная nmар и утилита queso. Точность результатов, выдаваемых обеими утилитами, объясняется, прежде всего, тем, что обе они предоставляют возможность исследования стека протоколов TCP/IP (stack fingerprinting).

**Заключение**

Целью данной курсовой работы было обзорной статьей о передаче информации в локальных сетях и методах её отслеживания (нахождения).

В ходе курсовой работы мною было проанализировано программное обеспечение, которое позволяет осуществлять сканирование ресурсов локальной сети. Так же мной были изучены технологии, с помощью которых осуществляется сканирование. Вся информация была представлена в форме курсового проекта.

**Список литературы**

1. Мурлина В.А. Информатика и программирование. Методическое указание к курсовой работе по дисциплине «Информатика и программирование» для студентов всех форм обучения специальности 080801 – «прикладная информатика в экономике» факультета компьютерных технологий и автоматизированных систем.- Краснодар: издательство КубГТУ, 2004. – 49с.

2. С.В. Пеников «Локальные сети», издательство «АПВ», 2004 - 151 с.

3. Г.Т. Сизов «Системы и сети», издательство «Интер» 2005 – 204 с.