Л6. Методы сканирования и уклонения в Kali Linux.

Рассмотрим следующие темы.

- Описание метода обнаружения цели.
- Как с помощью инструментов Kali Linux распознать целевую машину.
- Шаги, которые необходимо выполнить для поиска операционных систем целевых машин (получение отпечатков операционной системы).
- Автоматическое сканирование с помощью Striker.
- Сокрытие с помощью Nipe.

Чтобы было понятнее, в качестве целевой машины мы будем использовать виртуальную машину.

Технические условия

Ваша система должна соответствовать следующим техническим условиям.

- Минимальные требования к оборудованию: 6 Гбайт оперативной памяти, четырехъядерный процессор 2,4 ГГц и жесткий диск 500 Гбайт.
- Kali Linux 2018.
- Виртуальная машина для тестирования, например Metasploitable или Bad Store (см. главу 2).

Начинаем с обнаружения цели

После того как информация о целевой сети или машине была собрана с помощью сторонних источников, можно приступать к обнаружению целевой машины. Цели обнаружения следующие.

- Найти в целевой сети доступные машины. Если целевая машина недоступна, мы не можем проводить на ней тест на проникновение и перейдем к следующей машине.

- Определить операционную систему, установленную на целевой машине.
- Использовать собранную таким образом информацию в процессе сопоставления уязвимостей.

Для процесса обнаружения целей мы можем использовать инструменты, предоставляемые Kali Linux. Некоторые из них доступны в меню Information Gathering (Сбор информации). Другие приложения придется запускать из командной строки.

В этой главе мы опишем лишь несколько важных инструментов из каждой категории. Инструменты выбраны в зависимости от их функциональности, популярности и поставленных перед исследователем целей.

В этой главе в качестве целевой системы мы используем установленную ранее Metasploitable 2. Каждую из предложенных команд вы можете опробовать в этой операционной системе.

Идентификация целевой машины

Инструменты этой категории используются для определения целевых машин, к которым испытатель на проникновение может получить доступ. Прежде чем начать процесс идентификации, мы должны знать условия и соглашения, предъявляемые нашим клиентом.

Если в соглашении требуется скрыть все действия по тестированию на проникновение, мы все опыты должны проводить скрытно. Методы скрытности могут также применяться для тестирования функциональности *системы обнаружения вторжений (IDS)* или *системы предотвращения вторжений (IPS)*. Если такие требования не обговаривались, проведение испытания на проникновение скрывать не следует.

ping

ping — самый известный и часто применяемый инструмент, который используется для проверки доступности конкретного хоста. Он работает следующим образом: сначала целевой машине или сети отправляется пакет эхо-запроса протокола *ICMP* (*Internet Control Message Protocol* — протокол межсетевых управляющих сообщений). Если целевая машина доступна и брандмауэр не блокирует пакет эхо-запроса ICMP, он вышлет пакет эхо-ответа ICMP.

Запрос проверки связи ICMP и эхо-ответ ICMP — это два сообщения ICMP управления. Чтобы узнать о других управляющих сообщениях ICMP, обратитесь по адресу https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol#Control_messages.

В меню Kali Linux команды `ping` нет. Чтобы выполнить ее, откройте терминал и введите `ping` с нужными параметрами.

Чтобы выполнить тестирование целевого устройства, введите команду `ping` и IP-адрес целевого устройства (рис. 5.1).

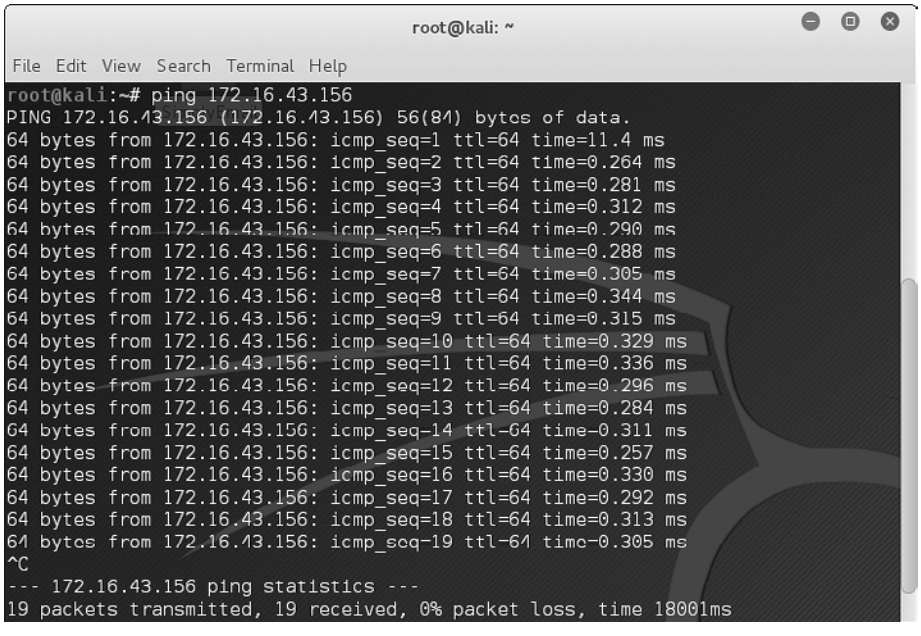


Рис. 5.1. Команда ping выполняется

По умолчанию этот тест будет идти непрерывно. Чтобы его остановить, нажмите сочетание клавиш Ctrl+C.

Инструмент ping имеет несколько параметров. Ниже показаны наиболее популярные.

- `-c` (счет) — число отправленных эхо-запросов.
- `-I` (IP-адрес интерфейса) — это IP-адрес целевой машины. Аргументом может быть числовой IP-адрес (например, 192.168.56.102) или имя устройства (например, eth0). Данный параметр можно применять для проверки связи с локальным адресом IPv6.
- `-s` (размер пакета) — указывает количество отправляемых байтов данных. По умолчанию размер пакета составляет 56 байт, что в сочетании с 8 байтами данных заголовка ICMP преобразуется в 64 байта данных ICMP.

Посмотрим, как эти параметры применяются на практике. Предположим, испытание на проникновение вы начинаете с внутреннего теста. Клиент предоставил вам список IP-адресов целевых серверов и доступ к локальной сети по кабелю.

Первое, что вам следует сделать перед запуском основного теста на проникновение, — проверить, доступны ли с вашей машины целевые серверы. Для этого вам вполне подойдет команда `ping`.

Допустим, IP-адрес целевого сервера — 172.16.43.156, в то время как IP-адрес вашего компьютера — 172.16.43.150. Для проверки доступности целевого сервера введите следующую команду:

```
ping -c 1 172.16.43.156
```

Вместо IP-адреса целевой машины ping также принимает имена хостов.

На рис. 5.2 показан результат, который мы получим после выполнения этой команды.

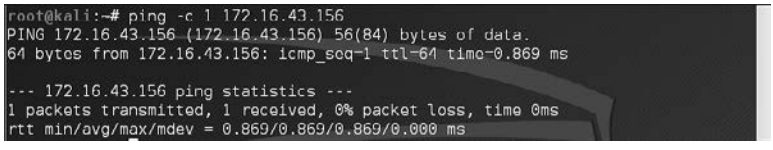


Рис. 5.2. Результат выполнения команды ping

Мы видим, что пакет эхо-запроса ICMP был передан назначению (IP-адрес = = 172.16.43.156). В ответ компьютеру (IP-адрес = 172.16.43.150) был возвращен эхо-ответ. На передачу пакета, прием целевым компьютером и обратный ответ было потрачено 0,869 миллисекунды. Потери пакетов нет.

Посмотрим, какие сетевые пакеты передаются и принимаются нашей машиной. Для захвата пакетов мы используем анализатор сетевого протокола Wireshark (рис. 5.3).

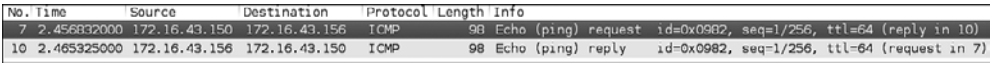


Рис. 5.3. Анализируем захваченные пакеты

На рис. 5.3 видно, что наш компьютер (172.16.43.150) отправил целевому компьютеру (172.16.43.156) один пакет эхо-запроса ICMP. Целевой компьютер на этот эхо-запрос передал нашей машине пакет с эхо-ответом. Более подробно Wireshark мы рассмотрим в главе 9.

Если на целевом компьютере используется IP-адрес протокола IPv6, например fe80::20c:29ff:fe:18:f:08, то для проверки его доступности мы можем воспользоваться инструментом ping6. Вам для работы с локальным адресом следует добавить в команду параметр `-I`:

```
# ping6 -c 1 fe80::20c:29ff:fe18:f08 -I eth0
PING fe80::20c:29ff:fe18:f08(fe80::20c:29ff:fe18:f08) from
fe80::20c:29ff:feb3:137 eth0: 56 data bytes
64 bytes from fe80::20c:29ff:fe18:f08: icmp_seq=1 ttl=64 time=7.98 ms
--- fe80::20c:29ff:fe18:f08 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.988/7.988/7.988/0.000 ms
```

На рис. 5.4 показаны пакеты, отправленные для выполнения запроса ping6. Здесь видно, что ping6 использует для запроса и ответа протокол *ICMPv6*.

Рис. 5.4. Выполнение запроса ping6

Для блокировки ping-запроса нужно настроить брандмауэр так, чтобы он отвечал на эхо-запросы только с определенного хоста, а эхо-запросы с остальных хостов игнорировал.

fping

Разница между *ping* и *fping* заключается в том, что инструмент fping может отправлять ping нескольким хостам одновременно. В командной строке можно указать несколько целевых компьютеров или использовать файл, содержащий хосты для проверки связи.

В fping по умолчанию ping отслеживает ответ от целевого компьютера. Если целевой компьютер отправляет ответ, он будет отмечен и удален из списка назначения. Если целевой компьютер не отвечает в течение определенного срока, он будет помечен как недоступный. По умолчанию fping попытается отправить три пакета эхо-запроса ICMP для каждой цели.

Для доступа к fping используется следующая команда:

```
# fping -h
```

В ответ мы получим инструкцию по использованию этой команды и доступные параметры.

Следующие сценарии дадут вам представление, как можно использовать fping.

Если нам нужно одновременно опросить несколько целевых машин с IP-адресами 72.16.43.156, 172.16.43.150 и 172.16.43.155, мы можем ввести следующую команду:

```
fping 172.16.43.156 172.16.43.150 172.16.43.155
```

Ниже показан результат ее выполнения:

```
# fping 172.16.43.156 172.16.43.150 172.16.43.155
172.16.43.156 is alive
172.16.43.150 is alive
ICMP Host Unreachable from 172.16.43.150 for ICMP Echo sent to
172.16.43.155
ICMP Host Unreachable from 172.16.43.150 for ICMP Echo sent to
172.16.43.155
ICMP Host Unreachable from 172.16.43.150 for ICMP Echo sent to
172.16.43.155
ICMP Host Unreachable from 172.16.43.150 for ICMP Echo sent to
172.16.43.155
172.16.43.155 is unreachable
```

Мы также можем генерировать список хостов автоматически, без определения один за другим IP-адресов и идентификации работающих и отвечающих на запросы компьютеров. Предположим, мы хотим найти включенные и отвечающие на запросы хосты в сети 172.16.43.0/24. Для этого следует использовать параметр `-g` и задать сеть для проверки, как в команде:

```
# fping -g 172.16.43.0/24
```

Если мы хотим изменить количество попыток проверки связи, нужно использовать параметр `-r` (ограничение повторных попыток), как показано далее. По умолчанию инструмент выполняет три попытки отправки пакетов.

```
fping -r 1 -g 172.16.43.149 172.16.43.160
```

На что мы получим такой ответ:

```
# fping -r 1 -g 172.16.43.149 172.16.43.160
172.16.43.150 is alive
172.16.43.156 is alive
172.16.43.149 is unreachable
172.16.43.151 is unreachable
172.16.43.152 is unreachable
172.16.43.153 is unreachable
172.16.43.154 is unreachable
172.16.43.155 is unreachable
172.16.43.157 is unreachable
172.16.43.158 is unreachable
172.16.43.159 is unreachable
172.16.43.160 is unreachable
```

Чтобы собрать полную статистику, добавьте параметр `-s`:

```
fping -s www.yahoo.com www.google.com www.msn.com
```

На эту команду мы получим следующий ответ:

```
#fping -s www.yahoo.com www.google.com www.msn.com
www.yahoo.com is alive
www.google.com is alive
www.msn.com is alive
      3 targets
      3 alive
      0 unreachable
      0 unknown addresses
      0 timeouts (waiting for response)
      3 ICMP Echos sent
      3 ICMP Echo Replies received
      0 other ICMP received
28.8 ms (min round trip time)
30.5 ms (avg round trip time)
33.6 ms (max round trip time)
      0.080 sec (elapsed real time)
```

hping3

Средство *hping3* представляет собой генератор и анализатор сетевых пакетов командной строки. Благодаря возможности генерировать пользовательские сетевые пакеты hping3 можно задействовать для протокола TCP/IP при выполнении таких тестов, как сканирование портов, проверка правил брандмауэра и тестирование производительности сети.

Как утверждает разработчик, есть еще несколько вариантов использования hping3:

- тестирование правил брандмауэра;
- тестирование IDS;
- использование известных уязвимостей в стеке TCP/IP.

Чтобы получить доступ к hping3, перейдите в консоль и введите команду `hping3`. Вы можете задавать команды `hping3` несколькими способами: введя в командную строку, через интерактивную оболочку или с помощью сценария.

Без заданных в командной строке параметров `hping3` отправляет нулевой TCP-пакет на порт под номером 0. Для выбора другого протокола укажите в командной строке следующие параметры.

Короткий параметр	Длинный параметр	Значение
-0	--raw-ip	Отправляет необработанные IP-пакеты
-1	--icmp	Отправляет пакеты ICMP
-2	--udp	Передает пакеты UDP
-8	--scan	Выбирает режим сканирования
-9	--listen	Включает режим прослушивания

При использовании протокола TCP мы можем применять TCP-пакеты без каких-либо дополнительных флагов (это предусмотрено по умолчанию) или выбрать один из следующих вариантов.

Параметр	Имя флага
-S	sun
-A	ack
-R	rst
-F	fin
-P	psh
-U	urg
-X	xmas: flags fin, urg, psh set
-Y	ymas

Рассмотрим возможные случаи использования hping3.

Отправьте один пакет эхо-запроса ICMP на машину с IP-адресом 192.168.56.101. Для этого укажите следующие параметры: `-1` (чтобы выбрать протокол ICMP) и `-c1` (для установки набора пакетов в один пакет):

```
hping3 -1 172.16.43.156 -c 1
```

В ответ мы получим следующее:

```
# hping3 -1 172.16.43.156 -c 1
HPING 172.16.43.156 (eth0 172.16.43.156): icmp mode set, 28 headers + 0 data
bytes
len=46 ip=172.16.43.156 ttl=64 id=63534 icmp_seq=0 rtt=2.5 ms
--- 172.16.43.156 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 2.5/2.5/2.5 ms
```

Из полученных выходных данных мы можем определить, что целевая машина подключена к сети, работает и отвечает на эхо-запрос ICMP.

Чтобы проверить, правильны ли наши выводы, мы с помощью tcpdump захватили трафик. Захваченные пакеты показаны на рис. 5.5.

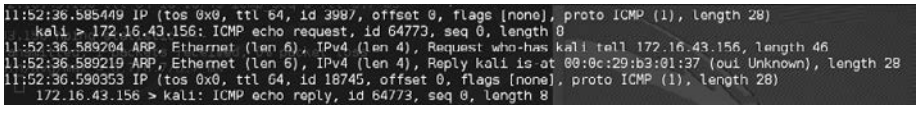


Рис. 5.5. Пакеты, захваченные с помощью tcpdump

Мы видим, что цель ответила пакетом эхо-ответа ICMP.

Мы можем вводить параметры не только в командную строку. Инструмент hping3 способен работать и в интерактивном режиме. Запустите терминал и введите в командную строку `hping3`. Появится приглашение, в котором можно вводить Tcl-команды.

Чтобы получить больше информации по командам Tcl, обратитесь к следующим ресурсам: http://www.invece.org/tclwise/ и http://wiki.tcl.tk/.

Ниже приведен соответствующий сценарий Tcl:

```
hping3> hping send {ip(daddr=172.16.43.156)+icmp(type=8,code=0)}
```

Если терминал не запущен, откройте окно терминала и для получения ответа от целевого сервера введите следующую команду:

```
hping recv eth0
```

После этого откройте еще одно окно терминала и введите в командную строку запрос. На рис. 5.6 показан ответ, который вы должны получить.

```
hping3> hping recv eth0
ip(ihl=0x0,ver=0x0,tos=0x00,totlen=0,id=0,fragoff=0,mf=0,df=0,rf=0,ttl=0,proto=0
,cksum=0x0000,saddr=0.0.0.0,daddr=0.0.0.0)
```

Рис. 5.6. Ответ на отправленный запрос

Можно также использовать hping3 для проверки правил брандмауэра. Предположим, у вас есть следующие правила брандмауэра.

- Принимать любые TCP-пакеты, направленные на порт 22 (SSH).
- Принимать любые TCP-пакеты, относящиеся к установлению соединения.
- Отбросить любые другие пакеты.

Чтобы проверить эти правила, для передачи пакета эхо-запроса ICMP введите в hping3 следующую команду:

```
hping3 -1 172.16.43.156 -c 1
```

В ответ вы получите такой код:

```
# hping3 -1 172.16.43.156 -c 1
HPING 172.16.43.156 (eth0 172.16.43.156): icmp mode set, 28 headers + 0 data bytes
--- 172.16.43.156 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Мы видим, что целевая машина не ответила на наш ping-запрос.

Отправьте TCP-пакет на порт 22 с флагом SYN. В ответ вы получите результат, показанный на рис. 5.7.

```
root@kali:~# hping3 172.16.43.156 -c 1 -S -p 22 -s 6060
HPING 172.16.43.156 (eth0 172.16.43.156): S set, 40 headers + 0 data bytes
len=46 ip=172.16.43.156 ttl=64 DF id=0 sport=22 flags=SA seq=0 win=5840 rtt=5.3 ms

--- 172.16.43.156 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 5.3/5.3/5.3 ms
```

Рис. 5.7. Ответ на запрос, отправленный на порт 22 с флагом SYN

На рис. 5.7 видно, что брандмауэр целевой машины позволяет пакету SYN достигать порта 22. Давайте проверим, разрешено ли UDP-пакету достигать порта 22 (рис. 5.8).

```
root@kali:~# hping3 -2 172.16.43.156 -c 1 -S -p 22 -s 6060
HPING 172.16.43.156 (eth0 172.16.43.156): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=172.16.43.156 name=UNKNOWN
status=0 port=6060 seq=0

--- 172.16.43.156 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 26.8/26.8/26.8 ms
```

Рис. 5.8. Проверяем, разрешено ли UDP-пакету достигать порта 22

На рис. 5.8 видно, что брандмауэр целевой машины не позволяет UDP-пакету достичь порта 22.

Возможности hping3 не ограничиваются вышеописанными операциями. В этой главе мы рассмотрели только несколько возможностей применения этого инструмента. Если вы хотите узнать больше, обратитесь к документации hping3, расположенной по адресу http://wiki.hping.org.

Получение отпечатков ОС

После того как мы установили, что целевая машина нам отвечает, мы можем узнать, какая операционная система на ней используется. Этот метод широко известен как *снятие отпечатков пальцев (fingerprinting) операционной системы*. Существует два метода снятия отпечатков пальцев: активный и пассивный.

При выполнении *активного* метода приложение отправляет целевой машине сетевые пакеты, а затем анализирует полученный ответ и определяет, какая операционная система установлена. Преимущество такого метода заключается в том, что процесс проходит быстро. Но есть и недостатки: например, целевая машина может заметить попытку получить информацию о своей операционной системе.

Чтобы избежать обнаружения, следует воспользоваться *пассивным* методом снятия отпечатков ОС. Этот метод был впервые разработан Михалом Залевски (Michal Zalewsky), когда он создавал инструмент под названием *p0f*. Основным преимуществом метода является то, что при работе этого инструмента уменьшается взаимодействие между целевой и испытательной машиной и скрытность снятия отпечатков значительно увеличивается. Наиболее существенный недостаток пассивного метода в том, что на процесс снятия отпечатков затрачивается гораздо больше времени.

В этом разделе мы рассмотрим несколько инструментов, которые можно использовать для снятия отпечатков ОС.

p0f. Инструмент p0f предназначен для пассивного снятия отпечатков операционной системы. Его можно применять для идентификации ОС на следующих компьютерах.

- Машины, которые подключаются к вашей испытательной машине (режим SYN, выбранный по умолчанию).
- Машины, к которым подключаетесь вы (режим SYN + ACK).
- Машины, к которым не удается подключиться (режим RST+).
- Машины, связь с которыми вы можете контролировать.

Инструмент p0f анализирует TCP-пакеты, отправленные в ходе сетевого обмена. Далее собирается статистика специальных пакетов, которые по умолчанию не стандартизированы ни одной корпорацией. Например, ядро Linux использует 64-байтовую ping-датаграмму, а Windows — 32-байтовую ping-датаграмму или

значение *времени жизни пакета (TTL)*. Для Windows значение TTL равно 128, а для Linux зависит от дистрибутива. Эту информацию p0f применяет для определения операционной системы удаленного компьютера.

При использовании входящего в состав Kali Linux инструмента p0f мы не смогли снять отпечатки операционной системы на удаленной машине. Мы выяснили, что в инструменте p0f не обновлена база данных отпечатков. К сожалению, нам не удалось найти последнюю версию базы данных отпечатков. Поэтому мы задействовали p0f версии 3.06b. Для использования этой версии скачайте файл TARBALL (http://lcamtuf.coredump.cx/p0f3/releases/p0f-3.06b.tgz) и скомпилируйте код, запустив сценарий build.sh. По умолчанию файл базы данных отпечатков (p0f.fp) располагается в текущем каталоге.

Если вы хотите изменить путь хранения файла и поместить его, например, в каталог /etc/p0f/p0f.fp, отредактируйте файл config.h и перекомпилируйте p0f. Если путь хранения не изменить, то для определения расположения файла базы данных отпечатков потребуется использовать параметр -f.

Чтобы получить доступ к p0f, откройте консоль и введите команду `p0f -h`. Она выведет на экран инструкцию по использованию приложения и описание всех параметров. Воспользуемся инструментом p0f для идентификации операционной системы, установленной на удаленной машине, к которой мы подключимся. Для этого нужно ввести в консоли следующую команду:

```
p0f -f /etc/p0f/p0f.fp -o p0f.log
```

Команда будет читать базу данных из файла и сохранит сведения в файле `p0f.log`. Затем вы увидите следующую информацию:

```
--- p0f 3.07b by Michal Zalewski <lcamtuf@coredump.cx> ---
[+] Closed 1 file descriptor.
[+] Loaded 320 signatures from '/usr/share/p0f/p0f.fp'.
[+] Intercepting traffic on default interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file 'p0f.log' opened for writing.
[+] Entered main event loop.
```

Теперь необходимо выполнить сетевые операции по созданию TCP-подключения, например просмотр удаленного компьютера или подключение целевого удаленного компьютера к нашему компьютеру. Для примера мы установили подключение к сайту HTTP, расположенному на компьютере 2.

При успешном снятии отпечатков с помощью p0f информацию об операционной системе целевой машины вы увидите в консоли. Кроме того, она сохранится в файле журнала (p0f.log).

Это сокращенный вариант того, что будет отображено в консоли:

```
.-[ 172.16.43.150/41522 -> 172.16.43.156/80 (syn+ack) ]-
|
| server   = 172.16.43.156/80
| os       = Linux 2.6.xe
| dist     = 0
| params   = none
| raw_sig  = 4:64+0:0:1460:mss*4,5:mss,sok,ts,nop,ws:df:0
```

На рис. 5.9 показан файл журнала с полученной информацией.

Рис. 5.9. Информация, записанная в журнал

Основываясь на предыдущем результате, мы узнали, что целью является операционная система Linux 2.6.

На рис. 5.10 приводятся данные, полученные с целевого компьютера.

Рис. 5.10. Информация об ОС, полученная с целевого компьютера

Сравнивая данные, мы поймем, что p0f правильно получил информацию об ОС целевого компьютера. Удаленная машина работает под управлением Linux версии 2.6.

Завершите работу p0f, нажав сочетание клавиш Ctrl+C.

Введение в сканирование портов

Самый простой метод сканирования портов тот, что используется на целевых компьютерах для определения состояния портов протоколов TCP и UDP. Открытый порт означает, что в целевом компьютере существует сетевая служба, которая прослушивает порт, и она доступна. Закрытый порт показывает, что службы, прослушивающей данный порт, нет.

После того как состояние порта будет определено, злоумышленник проверит версию используемого сетевой службой программного обеспечения и обнаружит уязвимости этой версии. Предположим, что сервер А имеет программное обеспечение веб-сервера версии 1.0. Несколько дней назад был выпущен бюллетень по безопасности. Информация об уязвимости в веб-серверах версии 1.0 была опубликована. Если злоумышленник узнает, какая версия веб-сервера используется, информация об уязвимости может быть задействована для атаки на этот сервер. Это простой пример того, что может сделать злоумышленник после получения информации о доступных на компьютере службах.

Прежде чем мы углубимся в мир сканирования портов, немного обсудим теорию протоколов TCP/IP.

Изучаем протокол TCP/IP

В состав протоколов TCP/IP включены десятки различных протоколов. Наиболее важные из них — TCP и IP. Протокол IP обеспечивает адресацию, маршрутизацию датаграмм и другие функции для подключения одной машины к другой. Протокол TCP отвечает за управление соединениями и обеспечивает надежную передачу данных между процессами на двух машинах. Протокол IP в модели *Open Systems Interconnection (OSI)* расположен на сетевом уровне 3, тогда как TCP — на транспортном уровне (уровень 4) OSI.

Кроме TCP, вторым ключевым протоколом на транспортном уровне является UDP. Конечно, вы можете спросить, в чем разница между этими двумя протоколами. Если коротко, TCP имеет следующие характеристики.

- *Это протокол, ориентированный на подключение*. Прежде чем TCP приступит к передаче данных, клиент и сервер устанавливают между собой TCP-соединение, используя механизм трехстороннего подтверждения связи.
  - Клиент инициирует соединение, отправляя на сервер пакет, содержащий флаг SYN (synchronize). Обратите внимание: в поле порядкового номера сегмента SYN находится *начальный порядковый номер (Initial Sequence Number, ISN)*. Этот номер выбирается случайным образом.
  - Сервер отвечает собственным сегментом SYN, содержащимся в ISN. Сервер подтверждает SYN клиента, отправляя флаг ACK (подтверждение), хранящий значение клиентского ISN + 1.
  - Клиент подтверждает полученное от сервера сообщение, отправив флаг ACK, содержащий серверный ISN + 1. После этого клиент и сервер могут обмениваться данными.
  - Чтобы прервать соединение, TCP должен выполнить такие шаги:
    1) клиент отправляет пакет, содержащий набор флагов FIN (finish);
    2) сервер передает пакет ACK (подтверждение), чтобы сообщить клиенту, что сервер получил пакет FIN;
    3) после того как сервер приложений подготовился к закрытию, он отправляет пакет FIN;
    4) затем клиент для подтверждения получения пакета FIN сервера отправляет пакет ACK. Обычно каждая сторона (клиент или сервер) может завершить связь, отправив пакет FIN.
- *Это надежный протокол*. TCP использует порядковый номер и подтверждение для идентификации пакетных данных. Получатель отправляет подтверждение после получения пакета. Если пакет потерян или подтверждения от получателя пакета нет, TCP еще раз автоматически ретранслирует этот пакет. Если пакеты поступают не по порядку, TCP изменит порядок пакетов перед отправкой приложению.
- *Приложения, которым необходимо передавать файлы или важные данные, используют протокол TCP*. Это такие приложения, как, например, протокол HTTP и протокол FTP.

Протокол UDP имеет противоположные протоколу TCP характеристики.

- UDP не устанавливает соединение. Иначе говоря, для отправки данных клиенту и серверу в начале передачи не нужно устанавливать UDP-соединение.
- Протокол предпримет все имеющиеся у него способы, чтобы отправить пакет в пункт назначения. Если же пакет потеряется, UDP не будет отправлять его повторно.

Протокол UDP используют приложения, для которых потеря некоторых пакетов не является критической. Это такие приложения, как потоковое видео и другие мультимедийные приложения. Другими известными приложениями, задействующими UDP, являются *система доменных имен (DNS)*, *протокол DHCP* и *протокол SNMP (Simple Network Management Protocol)*.

Для взаимодействия приложений применяются порты. При адресации указываются номера портов, через которые и происходит передача данных. Программный процесс прослушивает определенный порт на сервере, а клиентская машина посылает данные на порт сервера, где он должен быть обработан. Номера портов имеют 16-разрядный адрес, и число может варьироваться от 0 до 65 535. Чтобы избежать хаотичного использования портов, предусмотрены следующие универсальные соглашения о диапазонах их номеров.

- *Известные номера портов (от 0 до 1023):* номера портов этого диапазона зарезервированы и обычно используются серверными процессами, выполняемыми системным администратором или привилегированным пользователем. Например, серверные приложения занимают следующие номера портов: SSH (порт 22), HTTP (порт 80), HTTPS (порт 443).
- *Зарегистрированные номера портов (от 1024 до 49 151):* чтобы зарегистрировать один из этих номеров портов для своего клиент-серверного приложения, пользователям следует отправить запрос в *Internet Assigned Number Authority (IANA)*.
- *Частные, или динамические, номера портов (49 152–65 535):* любой пользователь может задействовать в этом диапазоне номера портов, не регистрируя их в IANA.

Теперь, когда мы кратко обсудили различия между TCP- и UDP-протоколами, опишем форматы сообщений TCP и UDP.

Тонкости форматов сообщений TCP и UDP

Сообщение TCP называется *сегментом*. Сегмент TCP состоит из заголовка и области данных. Заголовок TCP часто имеет размер 20 байт (без параметров). Его можно описать с помощью схемы, показанной на рис. 5.11.

Ниже приводится краткое описание каждого поля.

- Source Port (Порт источника) и Destination Port (Порт назначения) имеют длину по 16 бит. Порт источника — это порт на машине, передающей пакет. Порт назначения — порт на целевой машине, принимающей данный пакет.
- Sequence Number (Порядковый номер) (32 бита) при нормальной передаче хранит порядковый номер первого байта данных.
- Acknowledgment Number (Номер подтверждения) (32 бита) содержит порядковый номер отправителя, увеличенный на единицу.
- H. Len. (4 бита) — размер TCP-заголовка в 32-разрядных словах.

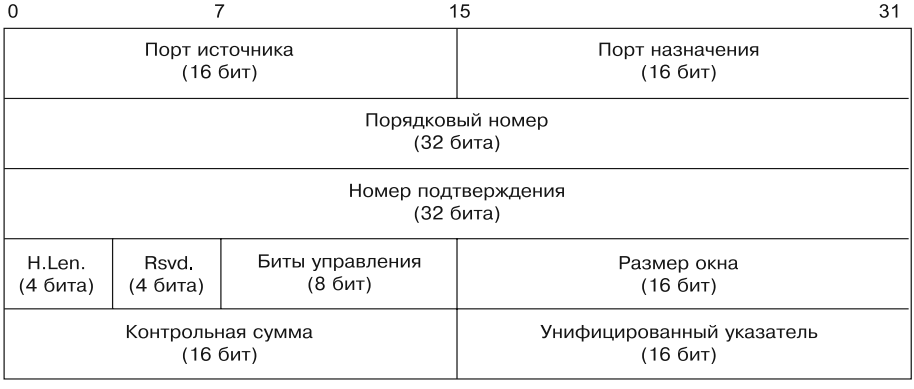


Рис. 5.11. Описание заголовка протокола TCP

- Rsvd. — зарезервирован для использования. Это четырехбитное поле с нулевым значением.
- Control Bits или Control flags (Биты управления) — содержит восемь однобитных флагов. В первоначальной спецификации (RFC 793) (можно загрузить по адресу http://www.ietf.org/rfc/rfc793.txt) TCP имеет шесть флагов.
- SYN — флаг синхронизации порядковых номеров. Используется во время установки сеанса.
- ACK — указывает, что поле подтверждения в заголовке TCP является значимым. Если в пакете содержится этот флаг, то он является подтверждением ранее полученного пакета.
- RST — сбрасывает соединение.
- FIN — указывает, что у стороны больше нет данных для отправки, и используется для корректного сброса соединения.
- PSH — указывает, что эти данные буферизованы и должны быть переданы приложению без ожидания дополнительных данных.
- URG — указывает, что это важное поле срочного указателя в заголовке TCP. Срочный указатель относится к важным номерам последовательности данных.

  Позже в RFC 3168 были добавлены еще два расширенных флага. Его можно загрузить по адресу http://www.ietf.org/rfc/rfc3168.txt.

  - CWR (Congestion Window Reduced — уменьшенное окно перегрузки) — данный флаг сообщает получателю данных, что очередь ожидающих отправки пакетов уменьшена из-за перегрузки сети.
  - ECN-Echo (Explicit Connection Notification-Echo — явное уведомление об эхо-подключении) — означает, что сетевое подключение перегружено.
- Window Size (Размер окна) (16 бит) указывает количество байтов, которые принимающая сторона готова принять.

- Checksum (Контрольная сумма) (16 бит) используется для проверки на ошибки заголовка TCP и данных.

Флаги могут быть установлены независимо друг от друга.

Чтобы получить дополнительные сведения о TCP, обратитесь к RFC 793 и RFC 3168.

При сканировании портов с помощью отправленного на целевую машину пакета SYN злоумышленник может столкнуться со следующими проблемами.

- Целевая машина отвечает пакетом SYN + ACK. Если порт открыт, мы получим этот пакет. Такое поведение определено в спецификации TCP (RFC 793) и обозначает, что если порт открыт, то пакет SYN должен отправляться с пакетом ACK (SYN + ACK), не рассматривая полезную нагрузку самого пакета SYN.
- Если порт закрыт, целевая машина отправляет обратно пакет с набором битов RST и ACK.
- Если порт недоступен и, скорее всего, заблокирован межсетевым экраном, целевая машина передает сообщение ICMP.
- Если от целевой машины ответ не поступает, то, скорее всего, сетевая служба не прослушивает выбранный порт или брандмауэр блокирует наш пакет SYN в автоматическом режиме.

Для испытателя на проникновение представляет интерес ситуация, когда порт открыт. Это значит, что на нем есть доступный сервис, который можно тестировать.

Для более эффективной атаки необходимо понять нюансы поведения TCP-портов.

Ниже мы расскажем о различных вариантах поведения UDP-портов, которые будут обнаружены при сканировании. Но сначала следует рассмотреть заголовок UDP (рис. 5.12).

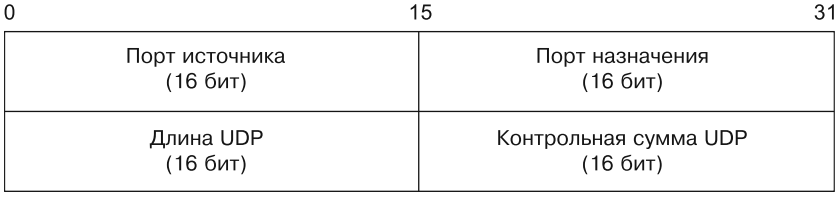


Рис. 5.12. Описание заголовка протокола UDP

Ниже приводится краткое описание каждого поля в заголовке UDP.

Как и заголовок TCP, заголовок UDP имеет исходный порт и порт назначения, каждый из которых имеет длину 16 бит. Исходный порт — это порт на отправляющей машине, которая передает пакет, в то время как порт назначения — порт на целевой машине, которая получает пакет.

- UDP Length (Длина UDP) — длина заголовка UDP.
- UDP Checksum (Контрольная сумма UDP) (16 бит) используется для проверки на ошибки заголовка UDP и данных.

Обратите внимание, что в заголовке UDP нет полей «Порядковый номер», «Номер подтверждения» и «Управляющие биты».

Во время сканирования UDP-портов на целевой машине злоумышленник может столкнуться со следующими ситуациями.

- Целевая машина отвечает пакетом UDP. Если мы этот пакет получим, значит, данный порт открыт.
- Целевой компьютер отправляет относительно тестируемого порта сообщение ICMP. Это значит, что порт закрыт. Однако если отправленное сообщение не является недоступным сообщением ICMP, это означает, что порт фильтруется брандмауэром.
- Если целевая машина в ответ не посылает никаких пакетов, это может указывать на одну из следующих ситуаций.
- Порт закрыт.
- Входящий UDP-пакет заблокирован.
- Ответ заблокирован.

Сканирование UDP-портов менее надежно по сравнению со сканированием TCP-портов, так как UDP-порт может быть открыт, но служба, прослушивающая этот порт, ищет конкретные полезные данные UDP и не отправляет никаких ответов.

Теперь, когда мы кратко рассмотрели теорию сканирования портов, применим ее на практике. В следующих разделах мы разберем несколько инструментов, которые можно использовать для выполнения сетевого сканирования.

Сетевой сканер

В этом разделе мы рассмотрим несколько инструментов, которые можно использовать для поиска открытых портов, идентификации удаленной операционной системы и перечисления служб на удаленной машине.

Перечисление служб — это метод, применяемый для поиска версии службы, доступной на определенном порте в целевой системе. Эти сведения важны, потому что, владея нужной информацией, испытатель на проникновение может искать уязвимости безопасности, которые существуют для этой версии службы.

В основном для служб используются стандартные порты. Но для некоторых служб системные администраторы могут изменять порты, применяемые по умолчанию. Например, по умолчанию служба SSH может быть привязана к порту 22.

Но системный администратор может изменить ее привязку и назначить порт 2222. В таком случае, если испытатель на проникновение просканирует порты, назначенные по умолчанию, порт для SSH не будет обнаружен. У испытателя могут возникнуть трудности с проприетарными приложениями, работающими на нестандартных портах. С помощью средств перечисления служб эти две проблемы можно решить и службу, привязанную к нестандартному порту, реально обнаружить.

Что такое Nmap

Nmap — это многофункциональный сканер портов, популярный в сообществе по IT-безопасности. Приложение написано и поддерживается Федором (Fyodor). Nmap — очень гибкий и качественный инструмент, который должен быть у каждого тестера на проникновение.

Nmap выполняет различные функции.

- **Обнаружение хостов.** Nmap можно использовать для поиска работающих хостов в целевых системах. По умолчанию Nmap для обнаружения хоста отправляет эхо-запрос ICMP, пакет TCP SYN на порт 443, пакет TCP ACK на порт 80 и запрос метки времени ICMP.
- **Обнаружение службы/версии.** После обнаружения портов Nmap может дополнительно проверить протокол службы, имя и номер версии приложения, используемого на целевом компьютере.
- **Обнаружение операционной системы.** Nmap отправляет ряд пакетов на удаленный хост и проверяет ответы. Затем он сравнивает эти ответы со своей базой данных отпечатков операционной системы и, если есть совпадение, выводит подробную информацию. Если Nmap не может определить операционную систему, он предоставляет URL-адрес, на который можно отправить отпечаток для обновления базы данных отпечатков ОС. Конечно, если вы знаете операционную систему, используемую в целевой системе, вам следует сразу отправить отпечаток.
- **Трассировка сети.** Это действие выполняется для определения порта и протокола, которые, вероятнее всего, достигнут целевой системы. Трассировка Nmap начинается с высокого значения TTL и уменьшается до тех пор, пока значение TTL не достигнет нуля.
- **Nmap Scripting Engine.** С помощью этой функции Nmap может быть расширен. Если вы хотите добавить в сканер не включенную по умолчанию проверку, напишите ее с помощью обработчика сценариев Nmap. В настоящее время проводятся проверки на наличие уязвимостей в сетевых службах и перечисление ресурсов в целевой системе.

Всегда рекомендуется проверять наличие новых версий. Если новая версия Nmap для Kali Linux найдена, для обновления выполните следующие команды:

```
apt-get update
apt-get install nmap
```

Запустить Nmap можно двумя способами. Первый — открыть меню Applications (Приложения), зайти в подменю Information Gathering (Сбор информации) и щелкнуть на названии нужного приложения. Второй способ — запустить терминал и ввести в консоль команду:

```
nmap
```

Будет запущен сканер Nmap, и вы увидите описание этого приложения. Возможно, для новичка описание покажется сложным.

К счастью, для сканирования удаленной машины требуется только один параметр. Это IP-адрес целевой машины или, если DNS правильно настроен, имя целевого хоста. Данная операция выполняется с помощью следующей команды:

```
nmap 172.16.43.156
```

Ниже приведен однозначный результат сканирования:

```
Nmap scan report for 172.16.43.156
Host is up (0.00025s latency).
Not shown: 977 closed ports
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
23/tcp   open   telnet
25/tcp   open   smtp
53/tcp   open   domain
80/tcp   open   http
111/tcp  open   rpcbind
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
512/tcp  open   exec
513/tcp  open   login
514/tcp  open   shell
1099/tcp open   rmiregistry
1524/tcp open   ingreslock
2049/tcp open   nfs
2121/tcp open   ccproxy-ftp
3306/tcp open   mysql
5432/tcp open   postgresql
5900/tcp open   vnc
6000/tcp open   X11
6667/tcp open   irc
8009/tcp open   ajp13
8180/tcp open   unknown
MAC Address: 00:0C;29:18:0F:08 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 1.7 seconds
```

Из этого вывода мы видим, что целевая машина очень уязвима для атаки, так как у нее много открытых портов.

Прежде чем мы продолжим использовать Nmap, посмотрим на состояния портов, которые могут быть идентифицированы сканером. Существует шесть состояний портов.

- Open (Открыт) — означает, что существует приложение, принимающее TCP-подключение, UDP-датаграмму или ассоциацию SCTP.
- Closed (Закрыт) — хотя порт и доступен, его не слушает ни одно приложение.
- Filtered (Фильтр) — Nmap не может определить, открыт порт или нет, потому что существует устройство фильтрации пакетов, блокирующее запросы.
- Unfiltered (Нефильтрованный) — означает, что порт доступен, но Nmap не может определить, открыт он или закрыт.
- Open|Filtered (Открыт|Отфильтрован) — Nmap не может определить, открыт порт или отфильтрован. Так происходит, когда сканирование открытых портов не дает ответа. Может быть достигнуто путем установки межсетевого экрана для сбрасывания пакетов.
- Closed|Filtered (Закрыт|Отфильтрован) — Nmap не может определить, закрыт порт или отфильтрован.

Далее мы опишем несколько вариантов, которые обычно используются во время тестирования на проникновение, и рассмотрим их применение на практике.

Спецификация цели

Все команды Nmap мы будем выполнять в командной строке. Мы также рассмотрим параметры и аргументы спецификации целевого хоста. Вместо имени хоста рекомендуем использовать его IP-адрес. В таком случае Nmap не нужно получать разрешение DNS, что ускоряет процесс сканирования портов.

В текущей версии Nmap поддерживает следующие спецификации IPv4-адресов.

- Поддерживается один узел, например 172.16.43.156.
- С помощью нотации CIDR поддерживается вся сеть смежных узлов, например 172.16.43.0/24. Эта спецификация будет включать 256 IP-адресов в диапазоне от 172.16.43.0 до 172.16.43.255.
- Поддерживается адресация октетного диапазона, например 172.16.2–4,6.1. Эта адресация будет включать четыре IP-адреса: 172.16.2.1, 172.16.3.1, 172.16.4.1 и 172.16.6.1.
- Поддерживается множество спецификаций хоста, например 172.16.43.1, 172.168.3–5,9.1.

Для IPv6-адреса Nmap поддерживает только полный формат IPv6 и имя хоста, например `fe80::a8bb:ccff:fedd:eeff%eth0`.

Кроме целевой спецификации, полученной из командной строки, Nmap с помощью параметра `-iL <inputfilename>` принимает целевое определение из текстового файла. Эта функция полезна, если у вас уже есть IP-адреса из другой программы.

Убедитесь, что записи в этом файле соответствуют целевому формату, поддерживаемому сканером Nmap. Записи должны быть разделены пробелами, табуляцией или символами перехода на новую строку.

Следующий код демонстрирует пример такого файла:

```
172.16.1.1-254
172.16.2.1-254
```

Теперь просканируем сеть 172.16.430/24. Наша задача — увидеть пакеты, отправленные Nmap. Для мониторинга отправленных пакетов можно использовать утилиту захвата пакетов, например tcpdump.

Запустите консоль и введите следующую команду:

```
tcpdump -nnX tcp and host 172.16.43.150
```

IP-адрес 172.16.43.150 принадлежит нашей машине, на которой запускается Nmap. Необходимо настроить его в соответствии с конфигурацией.

На этом же компьютере запустите еще одну консоль и введите следующую команду:

```
nmap 172.16.43.0/24
```

В консоли tcpdump вы увидите следующее:

```
    22:42:12.107532 IP 172.16.43.150.49270 >172.16.43.156.23: Flags [S],
seq 239440322, win 1024, options [mss 1460], length 0
      0x0000:  4500 002c eb7f 0000 3006 ad2e c0a8 3866  E..,....0.....8f
      0x0010:  c0a8 3867 c076 0017 0e45 91c2 0000 0000  ..8g.v...E......
      0x0020:  6002 0400 4173 0000 0204 05b4            `...As......
```

Из информации о пакете мы видим, что атакующая машина отправила пакет с флагом SYN с порта 49 270 на порт 23 целевой машины, предназначенный для Telnet. Если Nmap запускается привилегированным пользователем, например root в Kali Linux, то флаг SYN устанавливается по умолчанию.

На рис. 5.13 показан пакет, отправленный атакующей машиной на другие машины и порты в целевой сети.

Если удаленный компьютер ответит, вывод будет следующим:

```
22:36:19.939881 IP 172.16.43.150.1720 >172.16.43.156.47823: Flags [R.], seq 0,
ack 1053563675, win 0, length 0
  0x0000: 4500 0028 0000 4000 4006 48b2 c0a8 3867  E..(..@.@.H...8g
  0x0010: c0a8 3866 06b8 bacf 0000 0000 3ecc 1b1b  ..8f........>...
  0x0020: 5014 0000 a243 0000 0000 0000 0000       P....C........
```

Обратите внимание, что отправленный флаг обозначается символом R. Этот флаг можно сбросить. Он означает, что порт 1720 на целевой машине закрыт. Мы можем проверить это с предыдущим результатом Nmap.

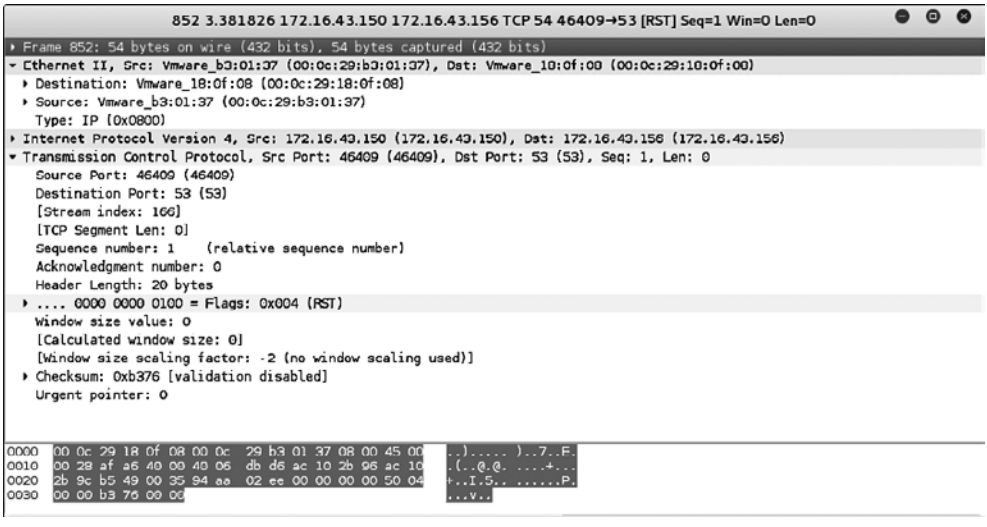


Рис. 5.13. Анализ пакета

Если же порт открыт, вы увидите следующий сетевой трафик:

```
22:42:12.108741 IP 172.16.43.156.23 >172.16.43.150.49270:Flags [S.], seq
1611132106, ack 239440323, win 5840,options [mss 1460], length 0
  0x0000: 4500 002c 0000 4000 4006 48ae c0a8 3867   E..,..@.@.H...8g
  0x0010: c0a8 3866 0017 c076 6007 ecca 0e45 91c3   ..8f...v`....E..
  0x0020: 6012 16d0 e1bf 0000 0204 05b4 0000
```

Здесь показано, что пакет в предыдущем коде подтверждает порядковый номер, показанный в ответе выше. То есть этот пакет имеет номер подтверждения 239 440 323, в то время как предыдущий пакет имел порядковый номер 239 440 322.

Параметры сканирования TCP

Для использования большинства параметров сканирования TCP Nmap требует привилегированного пользователя. Это может быть учетная запись корневого уровня в Unix или учетная запись администратора в Windows. Учетная запись применяется для отправки и получения необработанных пакетов. По умолчанию Nmap будет выполнять проверку TCP SYN, но если у сканера нет привилегированного пользователя, он будет использовать проверку TCP connect. В Nmap предусмотрены следующие виды сканирования.

- **TCP-сканирование подключения (-sT)** — параметр завершает трехстороннее подтверждение связи с каждым целевым портом. Если соединение установлено успешно, порт считается открытым. Трехстороннее рукопожатие — это медленный тип сканирования, поэтому, скорее всего, он будет внесен в журнал

целевой машины. Данный параметр сканирования применяется по умолчанию, если Nmap запускается пользователем без привилегий.

- Поскольку необходимо выполнить трехстороннее рукопожатие для каждого порта, этот тип развертки медленный и, скорее всего, будет внесен в журнал целевой машины. Если Nmap запускается пользователем без каких-либо привилегий, данный параметр сканирования выбирается по умолчанию.
- **SYN-сканирование (-sS)** — также известен как *полуоткрытый* или *скрытый SYN*. С помощью этого параметра Nmap отправляет SYN-пакет, а затем ожидает ответа. Ответ SYN/ACK означает, что порт прослушивается службой, а в случае ответа RST/ACK становится ясно, что порт не прослушивается. Если ответа нет или сообщение об ошибке ICMP недоступно, порт считается фильтрованным. Это быстрый тип сканирования. И еще одна деталь: так как трехстороннее рукопожатие не завершается, оно незаметно. Если Nmap запускается с правами привилегированного пользователя, данный параметр устанавливается по умолчанию.
- **TCP NULL-сканирование (-sN), FIN-сканирование (-sF), XMAS-сканирование (-sX)** — NULL-сканирование не устанавливает все биты управления. Сканирование FIN устанавливает только флаг FIN, а XMAS-сканирование устанавливает флаги FIN, PSH и URG. Если в ответ получен пакет RST, то порт считается закрытым, а отсутствие ответа означает, что порт открыт/отфильтрован.
- **TCP-сканирование Маймона (-sM)** — такое сканирование протокола было предложено Уриэлем Маймоном (Uriel Maimon). Оно отправит пакет с установленным флагом FIN/ACK. BSD-подобные системы при открытом порте этот пакет отбросят, а если порт закрыт, будет дан ответ RST.
- **TCP ACK-сканирование (-sA)** — этот тип сканирования используется для определения состояния брандмауэра и фильтрации портов. Сетевой пакет данного типа отправляет только бит ACK. Если в ответ получим RST, значит, цель не фильтруется.
- **TCP Window-сканирование (-sW)** — этот тип сканирования проверяет поля TCP Window ответа первого пакета. Открытый порт выдаст положительное значение окна TCP. Закрытый порт покажет нулевое значение окна TCP.
- **TCP Idle-сканирование (-sI)** — при использовании данного метода пакеты не отправляются целевой машине. Будет проведено сканирование указанного вами зомби-хоста. IDS сообщит, что атаку проводит зомби.

Кроме того, с помощью параметра `scanflags` Nmap позволяет вам создать собственное TCP-сканирование. Аргумент для этого параметра может быть числовым, например 9 для PSH и FIN, или описываться символьными именами. Для описания набора параметров следует после `scanflags` в любом порядке ввести комбинацию `URG`, `ACK`, `PSH`, `RST`, `SYN`, `FIN`, `ECE`, `CWR`, `ALL` или `NONE`. Например, параметр `--scanflags URGACKPSH` установит флаги URG, ACK и PSH.

Сканирование UDP

В то время когда для сканирования TCP предусмотрено много типов, для сканирования UDP — только один (`-sU`). Несмотря на то что проверка UDP менее надежна, чем проверка TCP, испытателю на проникновение не следует игнорировать ее, потому что на портах UDP также могут находиться интересные службы.

При сканировании портов UDP наибольшую проблему представляет скорость сканирования. Ядро Linux ограничивает отправку сообщения о недоступности порта ICMP одним сообщением в секунду, поэтому сканирование UDP 65 536 портов продолжится более 18 часов.

Для ускорения сканирования можно использовать следующие методы.

- Параллельное выполнение сканирования UDP.
- Сканирование сначала самых популярных портов.
- Сканирование за брандмауэром.
- Установку параметра `--host-timeout`, чтобы пропустить медленные хосты.

Эти методы позволяют сократить время, необходимое для сканирования UDP-портов.

Рассмотрим сценарий, в котором мы хотим найти, какие UDP-порты открыты на целевой машине. Для ускорения процесса сканирования мы будем проверять только порты 53 (DNS) и 161 (SNMP). Для этого используется следующая команда:

```
nmap -sU 172.16.43.156 -p 53,161
```

Ниже приведен результат ее выполнения:

```
Nmap scan report for 172.16.43.156
Host is up (0.0016s latency).
PORT    STATE  SERVICE
53/udp  open   domain
161/udp closed snmp
```

Спецификация порта Nmap

В конфигурации по умолчанию Nmap для каждого протокола случайным образом будет сканировать только 1000 наиболее распространенных портов. Файл `nmap-services` содержит оценку популярности для выбора «топовых» портов.

Чтобы изменить эту конфигурацию, Nmap предоставляет несколько параметров.

- `-p` *диапазон_портов* — сканируются только определенные порты. Для сканирования портов с 1 по 1024 введите команду `-p 1-1024`. Для сканирования портов с 1 по 65 535 используется команда `-p-`.
- `-F` (fast) — применяется для сканирования только 100 общих портов.

- `-r` (don't randomize port) — задает последовательное сканирование портов (от первого до последнего).
- `--top-ports <1 или больше>` — будет сканировать только *N* портов с самым большим значением, которое будет найдено в файле `nmap-service`.

Для поиска портов 22 и 25 с помощью метода проверки TCP NULL следует использовать такую команду:

```
nmap -sN -p 22,25 172.16.43.156
```

Результат сканирования отобразится в таком виде:

```
Nmap scan report for 172.16.43.156
Host is up (0.00089s latency).
PORT    STATE         SERVICE
22/tcp  open|filtered ssh
25/tcp  open|filtered smtp
MAC Address: 00:0C:29:18:0F:08 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Ниже приведены фрагменты дампа пакета:

```
23:23:38.581818 IP 172.16.43.150.61870 >172.16.43.156.22: Flags [], win 1024,
length 0
  0x0000: 4500 0028 06e4 0000 2f06 92ce c0a8 3866  E..(..../.....8f
  0x0010: c0a8 3867 f1ae 0016 dd9e bf90 0000 0000  ..8g............
  0x0020: 5000 0400 2ad2 0000                      P...*...

23:23:38.581866 IP 172.16.43.150.61870 >172.16.43.156.25: Flags [], win 1024,
length 0
  0x0000: 4500 0028 1117 0000 3106 869b c0a8 3866  E..(....1.....8f
  0x0010: c0a8 3867 f1ae 0019 dd9e bf90 0000 0000  ..8g............
  0x0020: 5000 0400 2acf 0000                      P...*...

23:23:39.683483 IP 172.16.43.150.61871 >172.16.43.156.25: Flags [], win 1024,
length 0
  0x0000: 4500 0028 afaf 0000 2706 f202 c0a8 3866  E..(....'.....8f
  0x0010: c0a8 3867 f1af 0019 dd9f bf91 0000 0000  ..8g............
  0x0020: 5000 0400 2acc 0000                      P...*...

23:23:39.683731 IP 172.16.43.150.61871 >172.16.43.156.22: Flags [], win 1024,
length 0
  0x0000: 4500 0028 5488 0000 3506 3f2a c0a8 3866  E..(T...5.?*..8f
  0x0010: c0a8 3867 f1af 0016 dd9f bf91 0000 0000  ..8g............
  0x0020: 5000 0400 2acf 0000                      P...*...
```

Из пакетов, показанных в предыдущем коде, мы видим следующее.

- В первом и втором пакетах атакующая машина проверяет, открыт ли порт 22 на целевой машине. Через некоторое время на целевой машине проверяется порт 25.

- В третьем и четвертом пакетах атакующая машина проверяет, открыт ли порт 25 на целевой машине. Через некоторое время на целевой машине проверяется порт 22.
- Далее атакующая машина в течение некоторого времени ожидает ответ от целевой машины. Если ответа нет, Nmap делает вывод, что эти два порта открыты или отфильтрованы.

Параметры вывода Nmap

Результат сканирования портов с помощью Nmap можно сохранить во внешний файл. Это действие можно применить, если вы хотите обработать результат Nmap другими инструментами. Но в любом случае, будет ли результат сканирования сохранен в файл или нет, результат будет отображен на экране.

Nmap поддерживает несколько выходных форматов.

- *Интерактивный вывод*. Этот формат вывода выбран по умолчанию, и результат отправляется на стандартный вывод.
- *Нормальный выход* (`-oN`). Подобен интерактивному выходу, но не содержит сведений о времени выполнения и предупреждений.
- *Вывод в XML* (`-oX`). Позволяет сохранить отчет в формате HTML, проанализировать в графическом интерфейсе пользователя Nmap или импортировать в базу данных. Рекомендуем вам использовать именно его.
- *Формат вывода* (`-oG`). Хоть этот формат устарел, он довольно популярен. Вывод Grepable состоит из комментариев (строк, начинающихся со знака фунта (#)) и целевых строк. Целевая строка содержит комбинацию из шести помеченных полей, разделенных символами табуляции, за которыми следует двоеточие. Это такие поля, как Host, Ports, Protocols, Ignored, OS, Seq, Index, IP ID Seq и Status. Такой формат выходных данных следует использовать, если они будут обрабатываться с помощью команд UNIX наподобие `grep` и `awk`.

Для сохранения результатов Nmap сразу в трех форматах (normal, XML и grepable) вы можете использовать параметр -oA.

Чтобы сохранить результат сканирования в формате HTML (`myscan.XML`), выполните следующую команду:

```
nmap 172.16.43.156 -oX myscan.xml
```

Ниже приведен фрагмент XML-файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href=file:///usr/bin/../share/nmap/nmap.xsl
 type="text/xsl"?>
```

```
<!-- Nmap 6.49BETA4 scan initiated Mon Feb 15 18:06:20 2016 as: nmap –oX
 metasploitablescan.xml 172.16.43.156 -->
<nmaprun scanner="nmap" args="nmap -oX metasploitablescan.xml
 172.16.43.156" start="1455588380" startstr="Mon Feb 15 18:06:20 2016"
 version="6.49BETA4"
<scaninfo type="syn" protocol="tcp" numservices="1000"
 services="1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,
 99-100,106,109-111,113,119,125,135,139,143-144,146,161,163,179,199,211-212,
 222,254-256,259,264,280,301,306,311,340,366,389,406-407,416-417,425,427,
 443-445,458,464-465,481,497,500,512-515,524,541,543-545,548,554-555,563,587,
 593,616-617,625,631,636,646,648,666-668,683,687,691,700,
```

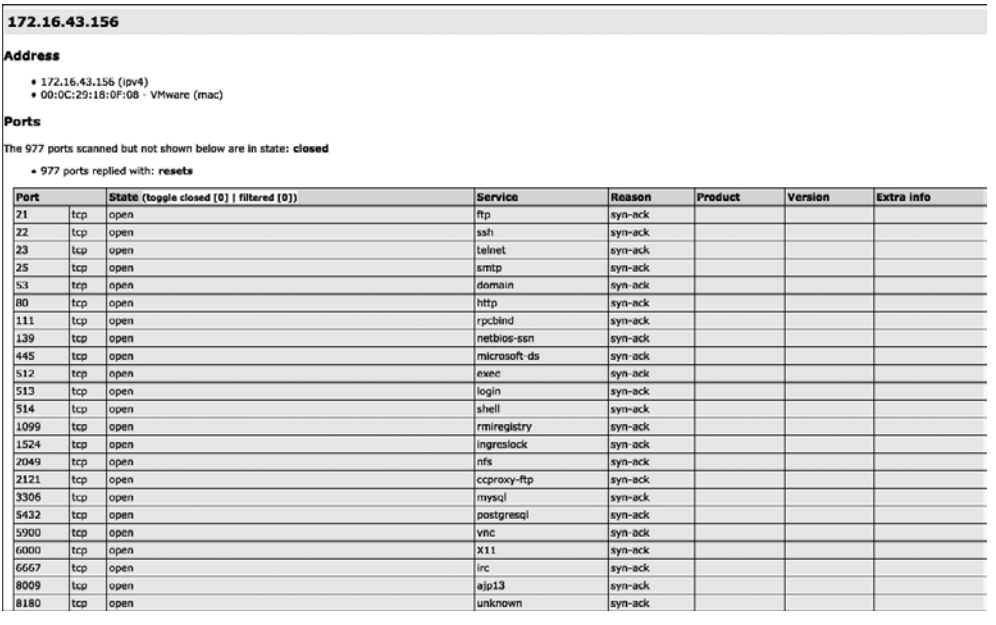
Для краткости из предыдущего фрагмента кода были удалены некоторые порты. В выходном XML-файле вы увидите каждый порт, просканированный Nmap. Ниже показан ответ от каждого отдельно сканируемого порта. Опять же для краткости в отчет включены не все порты:

```
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1455588380" endtime="1455588382"><status state="up"
 reason="arp-response" reason_ttl="0"/>
<address addr="172.16.43.156" addrtype="ipv4"/>
<address addr="00:0C:29:18:0F:08" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="977">
<extrareasons reason="resets" count="977"/>
</extraports>
<port protocol="tcp" portid="21"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="ftp" method="table" conf="3"/></port>
<port protocol="tcp" portid="22"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="ssh" method="table" conf="3"/></port>
<port protocol="tcp" portid="23"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="telnet" method="table" conf="3"/></port>
<port protocol="tcp" portid="25"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="smtp" method="table" conf="3"/></port>
<port protocol="tcp" portid="53"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="domain" method="table" conf="3"/></port>
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="http" method="table" conf="3"/></port>
<port protocol="tcp" portid="111"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="rpcbind" method="table" conf="3"/></port>
<port protocol="tcp" portid="139"><state state="open" reason="syn-ack"
 reason_ttl="64"/><service name="netbios-ssn" method="table"
 conf="3"/></port>
```

Выходные данные XML не очень удобны для просмотра. Чтобы отчет выглядел проще, следует преобразовать XML-файл от Nmap в HTML-формат. Данная операция позволит вам получить для отчета чистые выходные данные. Тогда сотрудники без технического образования смогут понять приведенные в отчете данные. Для конвертации XML-файла можно использовать команду `xsltproc`:

```
xsltproc myscan.xml -o myscan.html
```

На рис. 5.14 приведена часть отчета HTML, отображаемого браузером Firefox ESR, входящего в состав Kali Linux.



172.16.43.156

Address

- 172.16.43.156 (ipv4)
- 00:0C:29:18:0F:08 - VMware (mac)

Ports

The 977 ports scanned but not shown below are in state: **closed**

- 977 ports replied with: **resets**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
23	tcp	open	telnet	syn-ack			
25	tcp	open	smtp	syn-ack			
53	tcp	open	domain	syn-ack			
80	tcp	open	http	syn-ack			
111	tcp	open	rpcbind	syn-ack			
139	tcp	open	netbios-ssn	syn-ack			
445	tcp	open	microsoft-ds	syn-ack			
512	tcp	open	exec	syn-ack			
513	tcp	open	login	syn-ack			
514	tcp	open	shell	syn-ack			
1099	tcp	open	rmiregistry	syn-ack			
1524	tcp	open	ingreslock	syn-ack			
2049	tcp	open	nfs	syn-ack			
2121	tcp	open	ccproxy-ftp	syn-ack			
3306	tcp	open	mysql	syn-ack			
5432	tcp	open	postgresql	syn-ack			
5900	tcp	open	vnc	syn-ack			
6000	tcp	open	X11	syn-ack			
6667	tcp	open	irc	syn-ack			
8009	tcp	open	ajp13	syn-ack			
8180	tcp	open	unknown	syn-ack			

Рис. 5.14. Фрагмент отчета в браузере Firefox ESR

Если вы хотите обработать XML-вывод от Nmap по своему вкусу, можно обратиться к универсальным XML-библиотекам языков программирования. Кроме того, существует несколько библиотек, специально разработанных для работы с Nmap:

- *Perl* — Nmap-Parser (http://search.cpan.org/dist/Nmap-Parser/);
- *Python* — python-nmap (http://xael.org/norman/python/python-nmap/);
- *Ruby* — Ruby Nmap (http://rubynmap.sourceforge.net/);
- *PowerShell* — сценарий для структурного анализа данных Nmap XML (http://www.sans.org/windows-security/2009/06/11/powershell-script-to-parse-nmap-xml-output).

Параметры синхронизации

Nmap поставляется с шестью режимами синхронизации, которые устанавливаются с помощью параметров (`-T`).

- `paranoid (0)` — в этом режиме синхронизации пакет отправляется каждые пять минут. Пакеты отправляются последовательно. Режим используется для предотвращения обнаружения идентификаторов.

- `sneaky (1)` — пакет передается каждые 15 секунд. Параллельно никакие пакеты не передаются.
- `polite (2)` — пакет передается каждые 0,4 секунды и нет никакой параллельной передачи.
- `normal (3)` — в этом режиме пакеты одновременно отправляются группе целей. Данный режим в Nmap выбран по умолчанию. Это компромисс между затраченным на тест временем и сетевой нагрузкой.
- `aggressive (4)` — Nmap будет сканировать целевой компьютер пять минут, после чего перейдет к следующей цели. В этом режиме время ожидания ответа Nmap составляет не более 1,25 секунды.
- `insane (5)` — Nmap сканирует целевой компьютер 75 секунд, после чего переходит к следующей цели. В данном режиме время ожидания Nmap — не более 0,3 секунды.

Если вам не нужен более скрытый режим или более быстрая проверка, лучше выбирать режим по умолчанию.

Полезные параметры Nmap

В этом разделе мы обсудим несколько параметров Nmap, которые весьма полезны при проведении тестов на проникновение.

Определение версии службы

При сканировании портов можно попросить Nmap проверить версию службы. Эта информация очень полезна при последующей идентификации уязвимостей.

Чтобы задействовать такую возможность, при работе сканера укажите параметр `-sV`. Например, мы хотим найти версию программного обеспечения, используемого на порте 22:

```
nmap -sV 172.16.43.156 -p 22
```

На рис. 5.15 приведен результат выполнения этой команды.

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-20 13:54 PDT
Nmap scan report for 172.16.43.156
Host is up (0.00031s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
MAC Address: 00:0C:29:18:0F:08 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
```

Рис. 5.15. Результат сканирования порта 22

Из полученного ответа мы видим, что на порте 22 находится служба SSH, использующая программное обеспечение OpenSSH версии 4.7p1, и протокол SSH 2.0.

Обнаружение операционной системы

Nmap также можно попросить проверить операционную систему, используемую на целевом компьютере. Эта информация очень полезна при идентификации уязвимостей. Чтобы задействовать эту функцию, укажите параметр `-O`.

Например, мы хотим найти операционную систему, используемую на целевой машине:

```
nmap -O 172.16.43.156
```

В результате мы увидим следующие строки (рис. 5.16).

Рис. 5.16. Обнаружение операционной системы на целевой машине

Исходя из приведенной выше информации, мы видим, что удаленная система — это Linux, использующая ядро Linux версий 2.6.9–2.6.33. Если в ядрах Linux есть уязвимости, мы можем ими воспользоваться.

Отключение обнаружения узлов

Если хост блокирует запрос ping, Nmap может предположить, что целевая машина неактивна. По этой причине Nmap не сможет выполнить интенсивное зондирование, такое как сканирование портов, определение версий служб на портах и обнаружение операционной системы. Для решения этой проблемы в Nmap присутствует функция отключения обнаружения хостов, с помощью которой сканер будет считать, что целевая машина доступна, и выполнит интенсивное зондирование.

Для включения этой функции следует указать параметр `-Pn`.

Агрессивное сканирование

Для активации зонда добавьте параметр `-A`. В этом случае вы можете получить следующую информацию:

- обнаружение версии сервиса (`-sV`);
- обнаружение операционной системы (`-O`);
- сканирование сценариев (`-sC`);
- трассировка (`--traceroute`).

Сканирование такого типа может занять некоторое время. Для выполнения агрессивного сканирования введите следующую команду:

```
nmap -A 172.16.43.156
```

На рис. 5.17 приведен сокращенный отчет о выполнении команды.

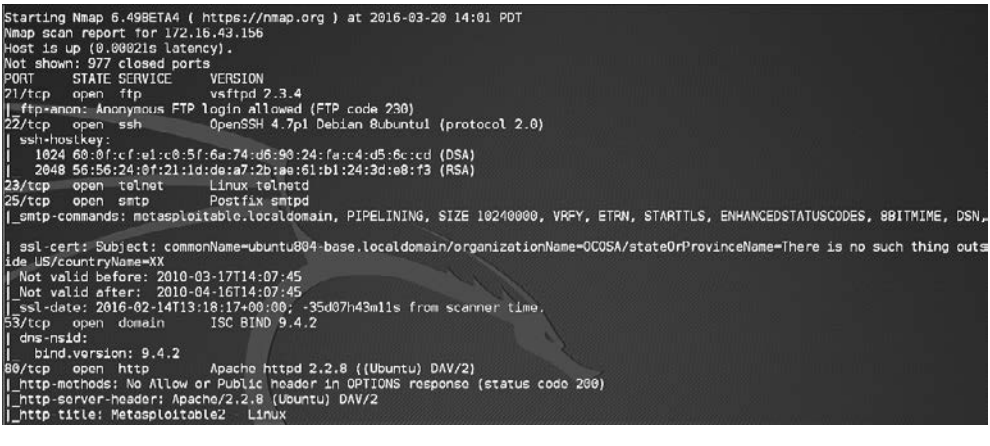


Рис. 5.17. Отчет о выполнении команды с параметром –A

В дополнение к подробной информации о портах, службах и сертификатах мы получаем подробную информацию о веб-сервере Apache, настроенном на целевой машине (рис. 5.18).

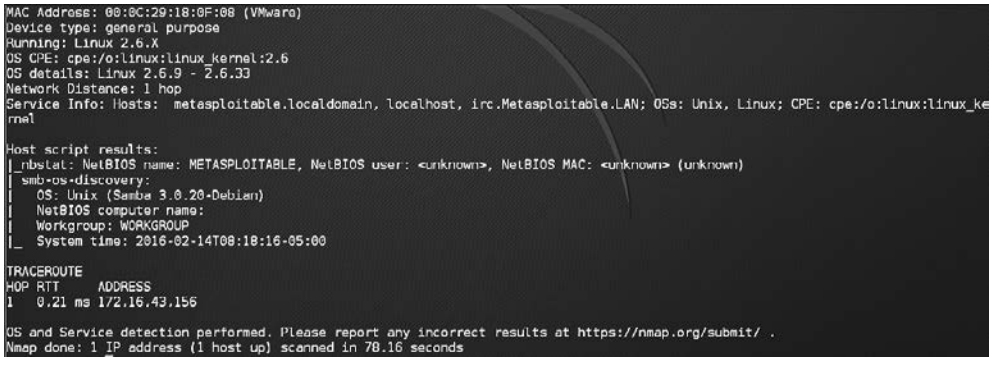


Рис. 5.18. Подробная информация о сервисе Apache

Nmap для сканирования Ipv6

В предыдущем разделе мы упоминали, что в качестве цели для Nmap можно указать цель Ipv6. Рассмотрим, как это сделать.

Ниже приведен Ipv6-адрес задействованного компьютера:

```
Target machine: fe80::20c:29ff:fe18:f08
```

Чтобы просканировать цель с Ipv6, перед целевым IP-адресом введите параметр `-6` и определите целевой адрес Ipv6. Сейчас мы можем указывать только отдельные Ipv6-адреса. Например:

```
nmap -6 fe80::20c:29ff:fe18:f08
```

На рис. 5.19 вы увидите результат, полученный после выполнения данной команды.

Рис. 5.19. Результат сканирования Ipv6

При тестировании Ipv6 мы видим, что количество открытых портов меньше, чем при тестировании Ipv4. Это объясняется тем, что не все службы на удаленном компьютере поддерживают Ipv6.

Сценарный движок Nmap

Nmap уже стал мощным инструментом исследования сети. Если этому приложению добавить возможности обработчика сценариев, оно станет более мощным инструментом. С помощью *Nmap Scripting Engine (NSE)* пользователи могут автоматизировать различные сетевые задачи, например проверку на наличие новых уязвимостей безопасности в приложениях, обнаружение версий приложений или другие возможности, недоступные в обычном Nmap. Сканер Nmap уже включил различные сценарии NSE в свой пакет, но пользователи могут писать и собственные сценарии в соответствии со своими потребностями.

Сценарии NSE основаны на языке программирования Lua (http://www.lua.org). Он встроен в Nmap, и в настоящее время сценарии NSE классифицируются следующим образом.

- `auth` — сценарии этой категории используются для проверки подлинности в целевой системе, например, с помощью метода грубой силы.
- `default` — для выполнения сценариев такого типа необходимо ввести параметры `-sC` или `-A`. Если сценарий будет соответствовать ниже приведенным требованиям, он будет сгруппирован в категорию по умолчанию:
  - сканирование должно быть быстрым;
  - сканирование должно добыть ценную информацию;
  - результаты сканирования должны быть краткими и подробными;
  - сканирование должно быть надежным;
  - тест не должен обнаруживаться целевой системой;
  - тест должен делиться информацией с третьей стороной.
- `discovery` (открытие) — сценарии используются для поиска сети.
- `DoS` — сценарии в этой категории могут вызвать в целевой системе отказ в обслуживании (DoS). Используйте эту возможность осторожно!
- `exploit` (эксплуатация) — сценарии пользуются уязвимостями в безопасности целевой системы. Испытателю на проникновение необходимо разрешение для выполнения таких действий в целевой системе.
- `external` (внешний) — сценарии такого типа предназначены для разглашения информации третьим лицам.
- `fuzzer` (затуманивание) — чтобы замаскировать разведывательную деятельность в целевой системе, применяйте такой сценарий.
- `intrusive` (навязчивый) — эти сценарии могут привести к сбою целевой системы или использовать все ресурсы тестируемой машины.
- `malware` (вредоносная программа) — сценарии проверяют, есть ли в целевой системе вредоносные программы или бэкдоры.
- `safe` (безопасный) — такие сценарии не должны вызывать сбой службы, отказ в обслуживании (DoS) или использовать целевую систему.

- `version` (версия) — в сценарий такого типа включен параметр обнаружения версий (`-sV`), назначение которого — расширенное обнаружение службы в целевой системе.
- `vuln` (уязвимости) — сценарии используются для проверки наличия уязвимостей в целевой системе.

Данные сценарии Nmap находятся в Kali Linux по адресу `/usr/share/nmap/scripts directories`. Сейчас в Nmap версии 7.70 содержится 588 сценариев. В Kali Linux включена именно эта версия Nmap.

Для вызова NSE предусмотрено несколько аргументов командной строки:

- `-sC` или `–script=default` — сканирование выполняется с помощью сценариев, выбранных по умолчанию;
- `--script` <*имя_файла*> | <*категория*> | <*каталоги*> — сканирование выполняется с помощью сценария, определенного в именах файлов, категориях или каталогах;
- `--script-args <args>` — запись предоставляет аргумент сценария. Если используется категория `auth`, примером таких аргументов могут быть имя пользователя или пароль.

Для сканирования порта 172.16.43.156 целевой машины с использованием сценариев по умолчанию введите следующую команду:

```
nmap –sC 172.16.43.156
```

Ниже приведен сокращенный результат ее выполнения:

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-02-22 17:09 PST
Nmap scan report for 172.16.43.156
Host is up (0.000099s latency).
Not shown: 977 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp open ssh
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp open telnet
25/tcp open smtp
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000,
 VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
| ssl-cert: Subject: commonName=ubuntu804-
 base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no
 such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after: 2010-04-16T14:07:45
|_ssl-date: 2016-02-12T05:51:52+00:00; -10d19h17m25s from scanner time.
53/tcp open domain
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp open http
```

```
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-title: Metasploitable2 – Linux
8009/tcp open ajp13
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open unknown
|_http-favicon: Apache Tomcat
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-title: Apache Tomcat/5.5
MAC Address: 00:0C:29:18:0F:08 (Vmware)
Host script results:
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>,
 NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP
|_  System time: 2016-02-12T00:51:49-05:00
Nmap done: 1 IP address (1 host up) scanned in 12.76 seconds
```

Из этого вывода видно, что, используя сценарии по умолчанию, Nmap получает более подробную информацию.

Если же вам требуется выборочная информация о целевой системе, целесообразно самостоятельно выбирать необходимые сценарии для проводимого теста. Например, если мы хотим собрать информацию об HTTP-сервере, следует в NSE выбрать такие HTTP-сценарии, как `http-enum`, `http-headers`, `http-methods` и `http-phpversion`:

```
nmap –script http-enum,http-headers,http-methods,http-php-version –p 80
172.16.43.156
```

На рис. 5.20 показан результат выполнения данной команды.

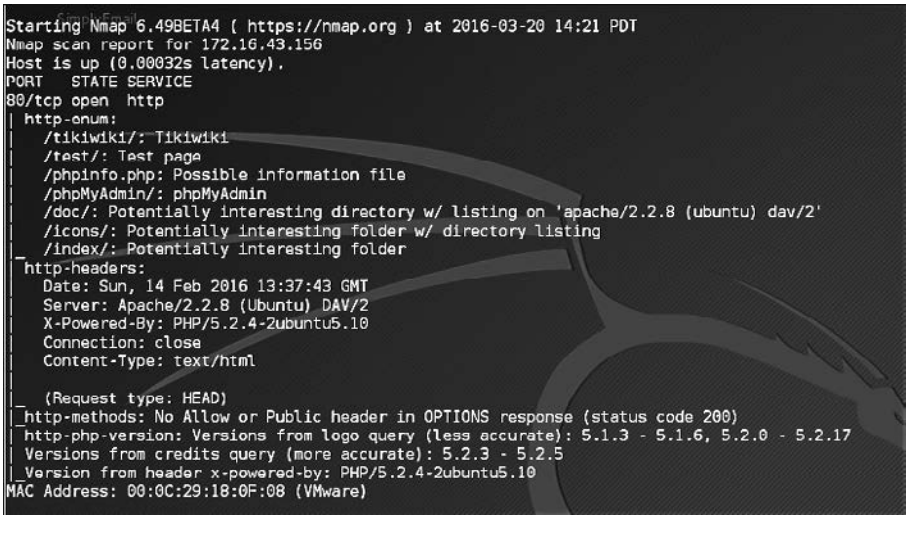


Рис. 5.20. Результат выполнения команды, включающей в себя HTTP-сценарии

Используя четыре HTTP-сценария NSE, мы получаем больше информации о веб-сервере целевой системы.

- Обнаружены несколько интересных каталогов: `Tikiwiki`, `test` и `phpMyAdmin`.
- Найден интересный файл: `phpinfo.php`.
- Мы узнали, что сервер использует PHP версии 5.2.3–5.2.5.

После обсуждения Nmap познакомимся со следующим инструментом сканирования портов.

Мы можем предложить вам полезный сценарий, который называется Nmap Vulscan HCE (http://www.computec.ch/mruef/software/nmap_nse_vulscan-1.0.tar.gz). Он сопоставляет информацию о версии, полученной от целевого компьютера, с базой данных уязвимостей, таких как CVE (http://cve.mitre.org/), VulDB (https://vuldb.com/?), SecurityTracker (http://securitytracker.com/) и SecurityFocus (http://www.securityfocus.com/).

На рис. 5.21 показан пример результата работы сценария CVE.

```
PORT      STATE    SERVICE         REASON      VERSION
22/tcp    open     ssh             syn-ack     OpenSSH 5.8p1 Debian 1ubuntu3
(Ubuntu Linux; protocol 2.0)
| vulscan: scipvuldb - http://www.scip.ch/en/?vuldb (12 findings):
| [7775] Red Hat Linux/Fedora 6 OpenSSH glibc error() privilege escalation
| [4584] OpenSSH up to 5.7 auth-options.c information disclosure
| [4282] OpenSSH 5.x Legacy Certificate Handler buffer overflow
| [2667] OpenBSD OpenSSH up to 4.5 Separation Monitor Designfehler
| [2578] OpenBSD OpenSSH up to 4.4 Signal Handler race condition
| [1999] OpenBSD OpenSSH up to 4.2p1 scp system() Designfehler
| [1724] OpenBSD OpenSSH up to 4.2p1 GSSAPIDelegateCredentials Designfehler
| [1723] OpenBSD OpenSSH up to 4.2p1 Dynamic Port Forwarding Designfehler
| [1083] Nokia IPSO 3.x OpenSSH Designfehler
| [299] OpenBSD OpenSSH 3.7p1/3.7.1p1 PAM Handler Konfigurationsfehler
| [287] OpenBSD OpenSSH up to 3.7.1 buffer_append_space() buffer overflow
| [100] OpenSSH Client IP Restrictions weak authentication
|
| cve - http://cve.mitre.org (69 findings):
| [CVE-2012-6066] freeSSHd.exe in freeSSHd through 1.2.6 allows remote
attackers to bypass authentication via a crafted session, as demonstrated
by an OpenSSH client with modified versions of ssh.c and sshconnect2.c.
| [CVE-2012-5975] The SSH USERAUTH CHANGE REQUEST feature in SSH Tectia
Server 6.0.4 through 6.0.20, 6.1.0 through 6.1.12, 6.2.0 through 6.2.5, and
6.3.0 through 6.3.2 on UNIX and Linux, when old-style password
authentication is enabled, allows remote attackers to bypass authentication
via a crafted session involving entry of blank passwords, as demonstrated
by a root login session from a modified OpenSSH client with an added
input_userauth_passwd_changereq call in sshconnect2.c.
| [CVE-2012-5536] A certain Red Hat build of the pam_ssh_agent_auth module
on Red Hat Enterprise Linux (RHEL) 6 and Fedora Rawhide calls the glibc
error function instead of the error function in the OpenSSH codebase, which
allows local users to obtain sensitive information from process memory or
possibly gain privileges via crafted use of an application that relies on
this module, as demonstrated by su and sudo.
| [CVE-2012-0814] The auth_parse_options function in auth-options.c in sshd
in OpenSSH before 5.7 provides debug messages containing authorized_keys
command options, which allows remote authenticated users to obtain
potentially sensitive information by reading these messages, as
```

Рис. 5.21. Результат выполнения сценария CVE

Параметры Nmap для обхода идентификаторов брандмауэра

Во время тестирования на проникновение мы можем столкнуться с защитой системы в виде брандмауэра и идентификаторов. Если вы используете настройки по умолчанию, ваши действия могут быть обнаружены или результат, полученный от Nmap, будет неточным. Для обхода брандмауэра/идентификаторов следует использовать такие параметры.

- `-f` (fragment packets — «фрагментированные пакеты») — назначение параметра в том, чтобы затруднить обнаружение пакетов. После указания этого параметра Nmap разделит пакет, находящийся после IP-заголовка, на 8 байт или меньше.
- `--mtu` — позволяет выбрать размер пакетов для фрагментации основного пакета. *Максимальный размер блока (MTU)* должен быть кратен восьми, иначе Nmap выдаст ошибку и завершит работу.
- `-D` (decoy — «приманка») — используя этот параметр, Nmap будет отправлять зонды от указанных пользователем поддельных IP-адресов. Смысл действия — скрыть в файлах журнала истинный IP-адрес пользователя. Для генерирования случайного IP-адреса вы можете использовать инструмент `RND` или `RND:number`. В качестве ложной цели используется хост, который выдает поток пакетов. Учтите, что наличие большого количества оправляемых пакетов может привести к перегрузке сети. При сканировании клиентской сети старайтесь этого избегать.
- `--source-port` <*номер_порта*> или `-g` (*ложный_порт_источника*) — этот параметр может быть полезным, если брандмауэр настроен на пропускание всего входящего трафика, поступающего с определенного порта.
- `--data-length` — с помощью этого параметра, заданного по умолчанию, мы можем избежать обнаружения при сканировании. Для этого изменяется длина данных, отправляемых Nmap.
- `--max-parallelism` — обычно значение параметра равно `1`. В этом случае Nmap будет одновременно отправлять на целевой компьютер не более одного зонда.
- `--scan-delay` <*время*> — параметр применяется для обхода идентификаторов/IP-адресов, которые используют пороговое значение для обнаружения активности сканирования портов.

Используя руководство Nmap (http://nmap.org/book/man-bypass-firewalls-ids.html), вы можете поэкспериментировать с другими параметрами маскировки.

Сканирование с Netdiscover

Netdiscover — еще один инструмент обнаружения, встроенный в Kali Linux 2018.2. В настоящее время его версия — .03-pre-beta 7, ее написал Хайме Пенальба (Jaime Penalba). С использованием запросов ARP Netdiscover может выполнять разведку и обнаружение как в беспроводных, так и в коммутируемых сетях.

Чтобы запустить Netdiscover, введите в командную строку терминала команду `netdiscover -h` (рис. 5.22). Параметр `-h` позволяет отобразить все параметры, которые можно применить при использовании этого инструмента. Если вы введете одну команду `netdiscover`, будет запущено сканирование с параметрами по умолчанию.

```
Netdiscover 0.3-pre-beta7 [Active/passive arp reconnaissance tool]
Written by: Jaime Penalba <jpenalbae@gmail.com>


Usage: netdiscover [-i device] [-r range | -l file | -p] [-m file] [-s time] [-n
 node] [-c count] [-f] [-d] [-S] [-P] [-c]
  -i device: your network device
  -r range: scan a given range instead of auto scan. 192.168.6.0/24,/16,/8
  -l file: scan the list of ranges contained into the given file
  -p passive mode: do not send anything, only sniff
  -m file: scan the list of known MACs and host names
  -F filter: Customize pcap filter expression (default: "arp")
  -s time: time to sleep between each arp request (milliseconds)
  -n node: last ip octet used for scanning (from 2 to 253)
  -c count: number of times to send each arp reques (for nets with packet loss)
  -f enable fastmode scan, saves a lot of time, recommended for auto
  -d ignore home config files for autoscan and fast mode
  -S enable sleep time supression between each request (hardcore mode)
  -P print results in a format suitable for parsing by another program
  -N Do not print header. Only valid when -P is enabled.
  -L in parsable output mode (-P), continue listening after the active scan is c
ompleted


If -r, -l or -p are not enabled, netdiscover will scan for common lan addresses.
root@kali:~#
```

Рис. 5.22. Netdiscover запущен с параметром –h

Для сканирования диапазона IP-адресов введите команду с параметром `-r` и добавьте исследуемый IP-диапазон. Для примера мы введем команду `netdiscover -r 0.10.0.0/24`. Добавив параметр `-p`, вы можете выбрать режим пассивного сканирования (рис. 5.23).

Выполнив сканирование, мы обнаружили рабочие станции Dell и HP, устройства Cisco и даже многофункциональные устройства Xerox.

```
28 Captured ARP Req/Rep packets, from 23 hosts.   Total size: 1680
 IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------
172.21.0.53     00:12:d9:ed:d8:3c      1      60  Cisco Systems, Inc
10.10.22.244    00:21:70:32:57:a7      3     180  Dell Inc.
10.10.0.79      00:1a:4b:2f:81:20      2     120  Hewlett Packard
10.10.0.1       cc:16:7e:04:23:e1      1      60  Cisco Systems, Inc
10.10.0.10      00:24:e8:32:c3:b8      1      60  Dell Inc.
10.10.0.50      00:14:38:d8:79:60      1      60  Hewlett Packard Enterprise
10.10.0.52      00:01:e6:39:91:10      1      60  Hewlett Packard
10.10.0.53      00:00:aa:f9:aa:e5      2     120  XEROX CORPORATION
10.10.0.54      fc:3f:db:c3:05:88      1      60  Hewlett Packard
10.10.0.55      9c:93:4e:4b:da:f5      1      60  Xerox Corporation
10.10.0.56      00:23:7d:72:49:56      1      60  Hewlett Packard
10.10.0.74      00:1a:4b:2f:91:cd      1      60  Hewlett Packard
10.10.0.84      38:63:bb:06:c5:d6      1      60  Hewlett Packard
10.10.0.93      5c:b9:01:eb:35:1a      1      60  Hewlett Packard
10.10.0.110     00:9e:1e:5b:ef:c1      1      60  Cisco Systems, Inc
10.10.0.112     00:9e:1e:50:2b:41      1      60  Cisco Systems, Inc
10.10.0.115     00:9e:1e:5b:ee:41      1      60  Cisco Systems, Inc

root@kali:~#
```

Рис. 5.23. Сканирование портов с помощью Netdiscover

Автоматическое сканирование с помощью Striker

Striker — это встроенный в Python инструмент автоматического сканирования и сбора хорошо скрытой информации. Striker выполняет сканирование портов, привязанных к ним служб, а также уязвимостей, присущих этим службам. Как и рассмотренные в предыдущей главе инструменты (Red_Hawk и Devploit), Striker прост в установке и использовании.

Сначала Striker нужно скачать. Для этого откройте терминал и, введя следующую команду, перейдите на Рабочий стол (или в каталог по вашему выбору):

```
cd Desktop
```

Чтобы клонировать Striker на Рабочий стол или в выбранный вами каталог (рис. 5.24), введите следующую команду:

```
git clone https://github.com/s0md3v/Striker.git
```

```
root@kali:~# cd Desktop
root@kali:~/Desktop# git clone https://github.com/s0md3v/Striker.git
Cloning into 'Striker'...
remote: Counting objects: 237, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 237 (delta 2), reused 0 (delta 0), pack-reused 230
Receiving objects: 100% (237/237), 123.63 KiB | 201.00 KiB/s, done.
Resolving deltas: 100% (123/123), done.
root@kali:~/Desktop#
```

Рис. 5.24. Клонирование Striker на Рабочий стол

После успешного завершения загрузки перейдите в каталог `Striker` (рис. 5.25). Для этого введите команду `cd Striker`, а затем с помощью команды `ls` просмотрите сохраненные в папке файлы. Вы должны увидеть список из пяти файлов, включая `requirements.txt` и `striker.py`.

```
root@kali:~/Desktop# cd Striker
root@kali:~/Desktop/Striker# ls
LICENSE  plugins  README.md  requirements.txt  striker.py
root@kali:~/Desktop/Striker#
```

Рис. 5.25. Список файлов, сохраненных в папке Striker

Чтобы Striker работал без ошибок, сначала мы должны использовать установщик управления пакетами (pip). Он обеспечит выполнение всех требований, необходимых для запуска Striker и модуля Whois (который предназначен для сбора информации).

Для запуска установщика введем две команды: сначала `pip install -r requirements.txt`, после — `pip install whois` (рис. 5.26).

```
root@kali:~/Desktop/Striker# pip install -r requirements.txt
Collecting requests[socks]==2.18.1 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/5a/58/671011e3ff4a06e29693
22267d78dcfda1bf4d1576551df1cce93cd7239d/requests-2.18.1-py2.py3-none-any.whl (8
8kB)
    100% |████████████████████████████████| 92kB 81kB/s
Requirement already satisfied: mechanize==0.2.5 in /usr/lib/python2.7/dist-packa
ges (from -r requirements.txt (line 2))
Collecting bs4==0.0.1 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f45617413
9ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting python-whois (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/63/8a/8ed58b8b28b6200ce1cd
fe4e4f3bbc8b85a79eef2aa615ec2fef511b3d68/python-whois-0.7.0.tar.gz (82kB)
    100% |████████████████████████████████| 92kB 244kB/s
Collecting whois (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/13/e8/656817674977bb7dd1dc
ee5e779daa10df65eca3dad65a018b0614bf2ac9/whois-0.7.tar.gz
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python2.7/dist-pac
kages (from requests[socks]==2.18.1->-r requirements.txt (line 1))
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/lib/python2.7/dist-
packages (from requests[socks]==2.18.1->-r requirements.txt (line 1))
Collecting urllib3<1.22,>=1.21.1 (from requests[socks]==2.18.1->-r requirements.
```

Рис. 5.26. Запуск установщика

После успешной установки всех компонентов введите команду `pip install who is` (даже если компонент уже был установлен) (рис. 5.27).

```
root@kali:~/Desktop/Striker# pip install whois
Requirement already satisfied: whois in /usr/local/lib/python2.7/dist-packages
```

Рис. 5.27. Установка компонента who is

Когда все компоненты будут установлены, для запуска Striker введите команду `python striker.py` (рис. 5.28).

Рис. 5.28. Запуск приложения Striker

Итак, Striker запущен. Теперь для начала сканирования следует ввести IP-адрес или URL целевой машины.

Для этого примера мы использовали сайт http://scanme.nmap.org/, упомянутый в разделе «Сканирование Nmap». Сравните результаты сканирования с результатами, полученными Nmap ранее (рис. 5.29).

Рис. 5.29. Результаты, полученные с помощью Striker

Обратите внимание, что атакующая машина нашла сведения о записи DNS, а также два адреса электронной почты (рис. 5.30).

```
[+] Host Records (A)
scanme.nmap.orgHTTP: (scanme.nmap.org) (45.33.32.156) AS63949 Linode, LLC United States

[+] TXT Records

[+] DNS Map: https://dnsdumpster.com/static/map/scanme.nmap.org.png

[>] Initiating 3 intel modules
[>] Loading Alpha module (1/3)
[>] Beta module deployed (2/3)
[>] Gamma module initiated (3/3)


[+] Emails found:
------------------
pixel-1532702357215843-web-@scanme.nmap.org
pixel-1532702359779164-web-@scanme.nmap.org
```

Рис. 5.30. Найденные адреса электронной почты и данные о записях DNS

Анонимность с помощью Nipe

Nipe — это инструмент, который в качестве шлюза пользователя по умолчанию задействует Tor-сеть, направляя через нее весь трафик. Обычно Tor используется для обеспечения некоторого уровня конфиденциальности и анонимности. Следует отметить, что при использовании данного инструмента для обеспечения анонимности маскировать один IP-адрес недостаточно, так как может быть доступна информация DNS. Для полной конфиденциальности и анонимности следует замаскировать как IP, так и DNS.

Сначала необходимо установить Nipe, клонировав его на Рабочий стол или в каталог по вашему выбору. Откройте терминал, перейдите в каталог на Рабочем столе или в выбранный вами каталог:

```
cd Desktop
```

Клонируйте Nipe на свой компьютер (рис. 5.31). Для этого введите следующую команду:

```
git clone https://github.com/GouveaHeitor/nipe.git
```

```
root@kali:~# cd Desktop
root@kali:~/Desktop# git clone https://github.com/GouveaHeitor/nipe.git
Cloning into 'nipe'...
remote: Counting objects: 744, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 744 (delta 0), reused 0 (delta 0), pack-reused 741
Receiving objects: 100% (744/744), 100.74 KiB | 488.00 KiB/s, done.
Resolving deltas: 100% (382/382), done.
root@kali:~/Desktop#
```

Рис. 5.31. Клонирование Nipe в выбранный каталог

Введите команду `cd Nipe`, чтобы перейти в каталог `Nipe`. Просмотрите содержимое этого каталога (рис. 5.32). Для этого введите команду `ls`.

```
root@kali:~/Desktop# cd nipe
root@kali:~/Desktop/nipe# ls
lib  LICENSE.md  nipe.pl  README.md
```

Рис. 5.32. Содержимое каталога Nipe

Чтобы установить Nipe, введите команду `cpan install Switch JSON LWP::UserAgent`. При появлении запроса на выполнение автоматической установки нажмите клавишу Enter (рис. 5.33).

```
root@kali:~/Desktop/nipe# cpan install Switch JSON LWP::UserAgent
Loading internal null logger. Install Log::Log4perl for logging messages

CPAN.pm requires configuration, but most of it can be done automatically.
If you answer 'no' below, you will enter an interactive dialog for each
configuration option instead.

Would you like to configure as much as possible automatically? [yes]
```

Рис. 5.33. Установка Nipe

Для установки зависимостей Nipe выполните следующую команду: `perl nipe.pl install` (рис. 5.34).

```
root@kali:~/Desktop/nipe# perl nipe.pl install
Reading package lists... Done
Building dependency tree
Reading state information... Done
iptables is already the newest version (1.6.2-1).
tor is already the newest version (0.3.3.9-1).
The following packages were automatically installed and are no longer required:
  dh-python libbabeltrace-ctf1 libcamel-1.2-60 libcdio17 libcue1
  libedataserver-1.2-22 libedataserverui-1.2-1 libfile-copy-recursive-perl
  libhttp-parser2.7.1 libisl15 libllvm5.0 libnfs8 libpoppler73
  libqgis-core2.18.17 libqgis-gui2.18.17 libqgis-networkanalysis2.18.17
  libqgispython2.18.17 libsynctex1 libtcl8.5 libtk8.5 libx265-146
  openjdk-9-jdk openjdk-9-jdk-headless openjdk-9-jre python-subprocess32
  python-unicodecsv python3-configargparse python3-editorconfig python3-flask
  python3-itsdangerous python3-jsbeautifier python3-pyinotify
  python3-simplejson python3-werkzeug tk8.5
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 539 not upgraded.
root@kali:~/Desktop/nipe#
```

Рис. 5.34. Установка зависимостей Nipe

Перед запуском Nipe проверьте общедоступный IP-адрес и DNS IP и сравните их с IP-адресами после запуска Nipe. Вот адреса двух сайтов, которые можно использовать для просмотра общедоступного IP: www.whatsmyipaddress.com и www.dnsleak.com.

Для запуска сервисов Nipe введите команду `perl nipe.pl start` (рис. 5.35).

```
root@kali:~/Desktop/nipe#
root@kali:~/Desktop/nipe# perl nipe.pl start
root@kali:~/Desktop/nipe#
```

Рис. 5.35. Запуск сервисов Nipe

Чтобы замаскировать свой IP, вы можете перезапустить службу. Для этого введите команду `perl nipe.pl restart`. Описание всех команд, предназначенных для установки и применения инструмента Nipe, вы найдете на странице GitHub, расположенной по адресу https://github.com/GouveaHeitor/nipe.

Для проверки IP и DNS используйте сайты, перечисленные ранее. С их помощью вы сможете убедиться, что ваши настройки действительно изменились.

Резюме

В этой главе мы обсудили, как обнаружить цель. Глава начиналась с обсуждения методов обнаружения цели: идентификации целевой машины и определения используемой на ней операционной системы. Затем мы продолжили работу с инструментами, входящими в состав Kali Linux и GitHub. Эти инструменты предназначены для обнаружения и идентификации целевых машин.

Далее мы рассмотрели несколько инструментов для обнаружения и сканирования целевых машин: ping, Nmap, p0f и Striker. Кроме того, вы узнали, как с помощью Nipe замаскировать ваш IP и DNS.

В следующей главе мы поговорим о сканировании уязвимостей и инструментах Kali Linux, которые для этого можно использовать.

Вопросы

1. Какой инструмент можно применять для отправки эхо-запросов ICMP одновременно нескольким хостам?
2. Сколько сценариев доступно в Nmap версии 7.7?
3. Каково назначение флага FIN?
4. Что означает понятие «отфильтрованный порт»?
5. Какой параметр Nmap позволяет затруднить обнаружение пакетов при обходе брандмауэров и идентификаторов?
6. Какая команда используется для сканирования с помощью средства Netdiscover диапазона IP-адресов?
7. Какой параметр Netdiscover применяется для запуска пассивного сканирования?
8. Какой сайт можно использовать для предотвращения утечки информации DNS?

Дополнительные материалы

- Сетевые инструменты Linux: https://gist.github.com/miglen/70765e663c48ae0544da08c07006791f.
- Обработчик сценариев Nmap: https://nmap.org/book/nse.html.
- Методы сканирования портов: https://nmap.org/book/man-port-scanning-techniques.html.