

7.

Nmap, Netdiscover, Striker, Nipe. Eve-ng.

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 172.16.43/24. 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          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.

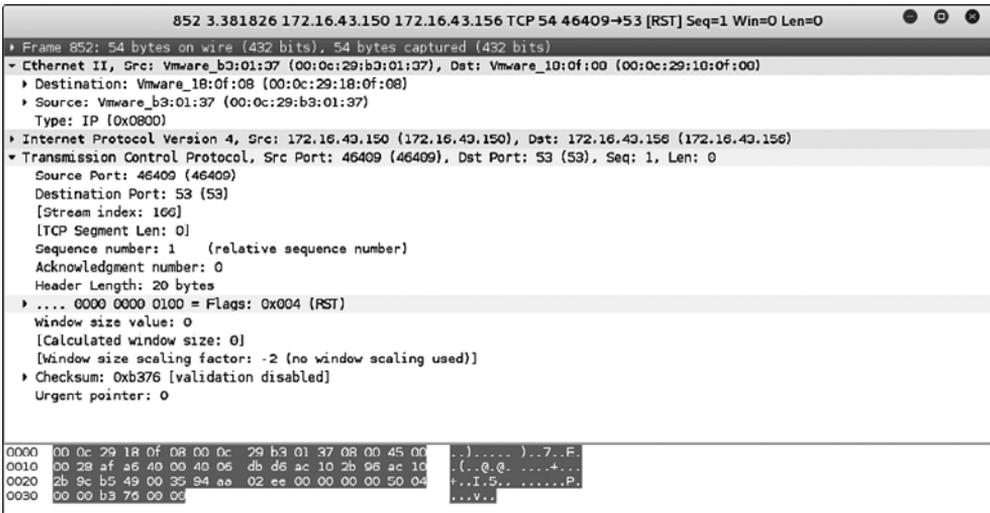


Рис. 5.13. Анализ пакета

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)*. Хотя этот формат устарел, он довольно популярен. Вывод Greppable 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 greppable) 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							
<ul style="list-style-type: none"> 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							
<ul style="list-style-type: none"> 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.

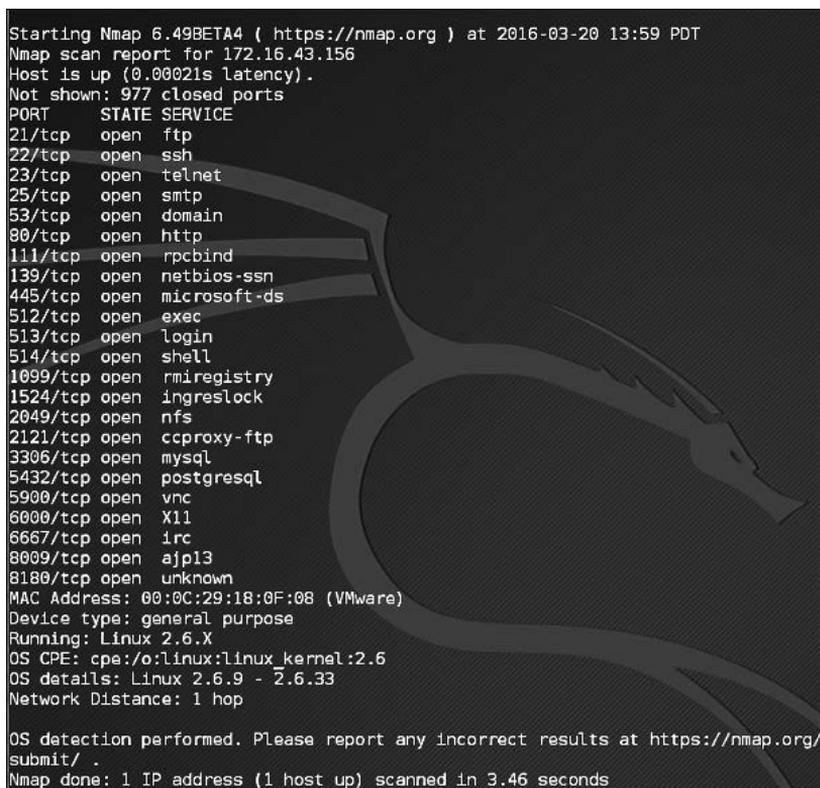
Обнаружение операционной системы

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).

The image shows a terminal window with a dark background and a dragon logo. The text is a Nmap scan report for the IP address 172.16.43.156. It lists various open ports and their corresponding services. At the bottom, it reports the OS detection results: Linux 2.6.X kernel, specifically Linux 2.6.9 - 2.6.33. The scan was completed in 3.46 seconds.

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-20 13:59 PDT
Nmap scan report for 172.16.43.156
Host is up (0.00021s 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)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.46 seconds
```

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.

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-20 14:01 PDT
Nmap scan report for 172.16.43.156
Host is up (0.00021s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ssh-hostkey:
|_ 1024 60:0f:c5:a1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_ 2048 56:56:24:0f:21:1d:da:a7:2b:ae:61:bl:24:3d:e8:f3 (RSA)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
|_smtp_commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_ssl-cert: Subject: commonName=ubuntu894-base.localdomain/organization=OCUSA/stateOrProvinceName=There is no such thing outsi
ide US/countryName=XX
|_ Not valid before: 2010-03-17T14:07:45
|_ Not valid after: 2010-04-16T14:07:45
|_ ssl-date: 2016-02-14T13:18:17+00:00; -35d07h43m11s from scanner time.
53/tcp    open  domain      ISC BIND 9.4.2
|_ dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_ http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_ http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_ http title: Metasploitable2 Linux
```

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).

```

MAC Address: 00:0C:29:18:0F:08 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ rblst: 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: 2016-02-14T08:18:16-05:00

TRACEROUTE
HOP RTT ADDRESS
1 0.21 ms 172.16.43.156

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 78.16 seconds

```

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.

```

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-20 14:16 PDT
Nmap scan report for fe80::20c:29ff:fe18:f08
Host is up (0.00011s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
2121/tcp  open  ccproxy-ftp
5432/tcp  open  postgresql
MAC Address: 00:0C:29:18:0F:08 (VMware)

```

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

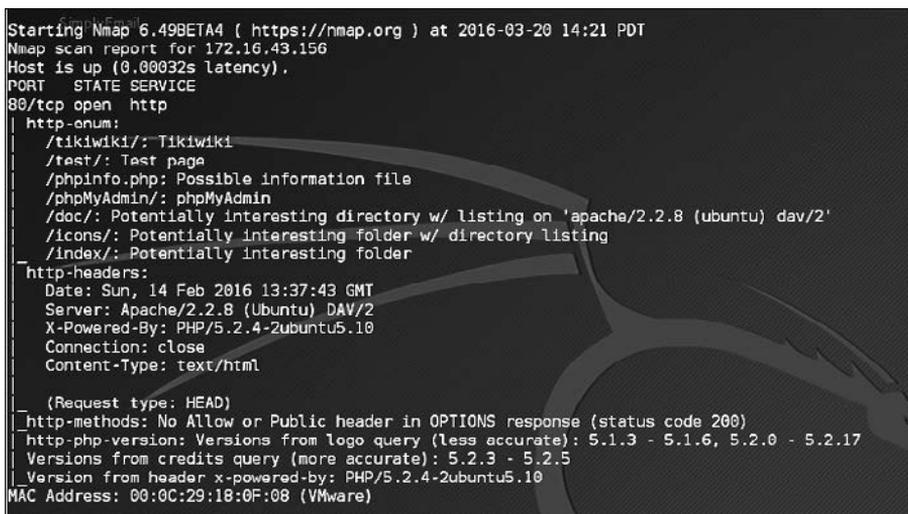
```

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.



```

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-20 14:21 PDT
Nmap scan report for 172.16.43.156
Host is up (0.00032s latency).
PORT      STATE SERVICE
80/tcp    open  http
|_http-enum:
| /tikiki/: Tikiwiki
| /test/: Test page
| /phpinfo.php: Possible information file
| /phpMyAdmin/: phpMyAdmin
| /doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2'
| /icons/: Potentially interesting folder w/ directory listing
| /index/: Potentially interesting folder
|_http-headers:
| Date: Sun, 14 Feb 2016 13:37:43 GMT
| Server: Apache/2.2.8 (Ubuntu) DAV/2
| X-Powered-By: PHP/5.2.4-2ubuntu5.10
| Connection: close
| Content-Type: text/html
|_ (Request type: HEAD)
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-php-version: Versions from logo query (less accurate): 5.1.3 - 5.1.6, 5.2.0 - 5.2.17
| Versions from credits query (more accurate): 5.2.3 - 5.2.5
| Version from header x-powered-by: PHP/5.2.4-2ubuntu5.10
MAC Address: 00:0C:29:18:0F:08 (Vmware)

```

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 lubuntu3
(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 отправляемых 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     100  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/58/671011e3ff4a06e2969322267d78dcfda1bf4d1576551df1cce93cd7239d/requests-2.18.1-py3-none-any.whl (88kB)
    100% |████████████████████████████████████████| 92kB 81kB/s
Requirement already satisfied: mechanize==0.2.5 in /usr/lib/python2.7/dist-packages (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/7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting python-whois (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/63/8a/8ed58b8b28b6200celcdfe4e4f3bbc8b85a79eef2aa615ec2fef511b3d68/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/e8/656817674977bb7dd1dcee5e779daa10df65eca3dad65a018b0614bf2ac9/whois-0.7.tar.gz
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python2.7/dist-packages (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 whois` (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).

```
root@kali:~/Desktop/Striker# python striker.py
[?] Enter the target: █
```

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).

```
[?] Enter the target: scanme.nmap.org
[!] IP Address : 45.33.32.156
[!] Server: Apache/2.4.7 (Ubuntu)
[+] Clickjacking protection is not in place.
[+] Operating System : Ubuntu
[!] scanme.nmap.org doesn't seem to use a CMS
[+] Honeypot Probability: 0%
-----
[~] Trying to gather whois information for scanme.nmap.org
[+] Whois information found
[-] Unable to build response, visit https://who.is/whois/scanme.nmap.org
-----
PORT      STATE SERVICE      VERSION
21/tcp    closed ftp
22/tcp    open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux;
protocol 2.0)
23/tcp    closed telnet
80/tcp    open  http        Apache httpd 2.4.7 ((Ubuntu))
110/tcp   closed pop3
143/tcp   closed imap
443/tcp   closed https
3389/tcp  closed ms-wbt-server
-----
```

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 Тог-сеть, направляя через нее весь трафик. Обычно Тог 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 libcuel
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 libsyntax1 libtcl8.5 libtk8.5 libx265-146
openjdk-9-jdk openjdk-9-jdk-headless openjdk-9-jre python-subprocess32
python-unicodcsv 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.

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?