Л36. Идентификация целевой машины с помощью Kali Linux. Развертывание Eve-ng.

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).

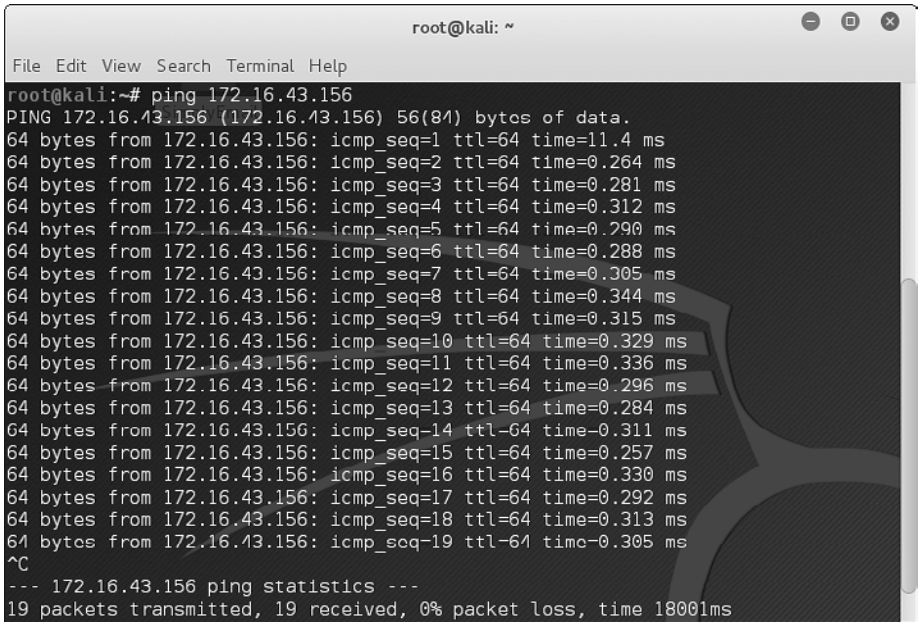

Рис. 5.1. Команда ping выполняется

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.

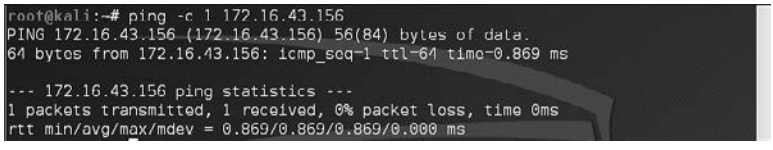

Рис. 5.2. Результат выполнения команды ping

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).

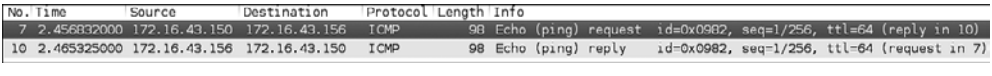

Рис. 5.3. Анализируем захваченные пакеты

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.

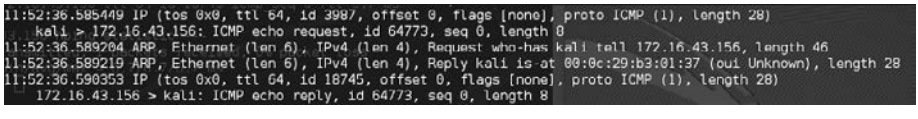

Рис. 5.5. Пакеты, захваченные с помощью tcpdump

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.

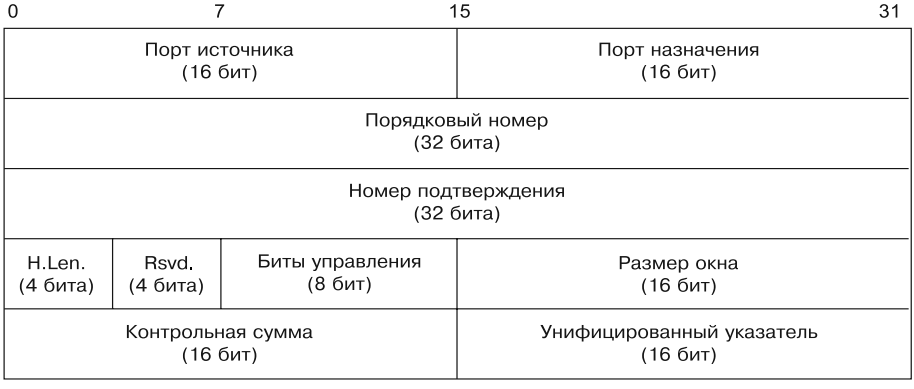

Рис. 5.11. Описание заголовка протокола TCP

- 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).

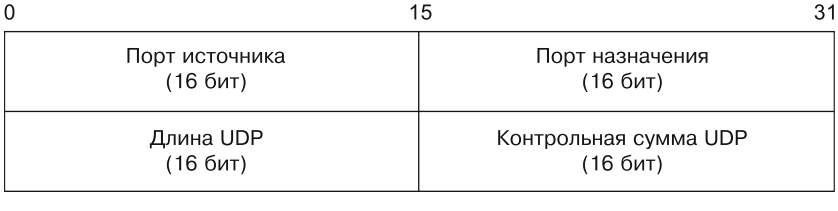

Рис. 5.12. Описание заголовка протокола UDP

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.