6.3 捕获和保存流量

首先要学习的是如何把当前流量捕获下来并把它们打印到屏幕上。要在 TShark 里捕获，仅需执行命令 tshark。这条命令会从网卡开始抓取当前流量，并会在你的终端窗口上实时显示抓取的结果，如下所示：

```
   C:\Program Files\Wireshark>tshark
  1   0.000000 172.16.16.128 -> 74.125.95.104 TCP 66 1606      80 [
Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
  2   0.030107 74.125.95.104 -> 172.16.16.128 TCP 66 80      1606 [
Seq=0 Ack=1 Win=5720 Len=0 MSS=1406 SACK_PERM=1 WS=64
  3   0.030182 172.16.16.128 -> 74.125.95.104 TCP 54 1606      80 [
Seq=1 Ack=1 Win=16872 Len=0
  4   0.030248 172.16.16.128 -> 74.125.95.104 HTTP 681 GET / HTTP/1
  5   0.079026 74.125.95.104 -> 172.16.16.128 TCP 60 80      1606 [
Seq=1 Ack=628 Win=6976 Len=0
```

要在 Tcpdump 里抓取流量，可执行 tcpdump 命令。一旦执行这条命令，你的终端窗口就会出现如下所示的内容：

```
   sanders@ppa:~$ tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol
listening on eth0, link-type EN10MB (Ethernet), capture size 65535
21:18:39.618072 IP 172.16.16.128.slm-api > 74.125.95.104.http: Flag
seq 2082691767, win 8192, options [mss 1460,nop,wscale 2,nop,nop,sa
length 0
21:18:39.648179 IP 74.125.95.104.http > 172.16.16.128.slm-api:
Flags [S.], seq 2775577373, ack 2082691768, win 5720, options [mss
1406,nop,nop,sackOK,nop,wscale 6], length 0
21:18:39.648254 IP 172.16.16.128.slm-api > 74.125.95.104.http: Flag
ack 1, win 4218, length 0
21:18:39.648320 IP 172.16.16.128.slm-api > 74.125.95.104.http: Flag
seq 1:628, ack 1, win 4218, length 627: HTTP: GET / HTTP/1.1
21:18:39.697098 IP 74.125.95.104.http > 172.16.16.128.slm-api: Flag
ack 628, win 109, length 0
```

> 注意
>
> 因为在 UNIX 系统里抓包需要管理员权限，所以你要以 root 账户运行 Tcpdump，或者在命令前加上 sudo。但在很多情况下，你在类 UNIX 系统上只有受限的普通用户权限。如果你遇到权限方面的问题，那么这可能就是原因所在。

根据你的系统配置，TShark 和 Tcpdump 可能不会默认从你设想的网卡抓取流量。如果这种情况发生了，你就需要手动去明确它。你可以使用 TShark 的-D 参数来列出当前所有可用的网卡，系统会以数字列表的形式打印出网卡信息，如下所示：

```
C:\Program Files\Wireshark>tshark -D
1. \Device\NPF_{1DE095C2-346D-47E6-B855-11917B74603A} (Local Area C
2)
2. \Device\NPF_{1A494418-97D3-42E8-8C0B-78D79A1F7545} (Ethernet 2)
```

要使用其中一个网卡，可以在命令后面添加-i 参数和上网卡的标号，如下所示：

```
C:\Program Files\Wireshark>tshark –i 1
```

这个命令会让 TShark 只抓取针对 Local Area Connection 2 网卡的流量，该网卡在列表里被标注为 1 号。我建议始终明确要从哪个网卡抓取流量，因为虚拟软件和 VPN 软件会在系统中添加自己的网卡，而且你也需要知道你捕获的网络流量来自哪里。

在 Linux 或者 OS X 系统运行 Tcpdump 的话，请使用 ifconfig 命令来列出可用的网卡。

```
sanders@ppa:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:1f:a7:55
          inet addr:172.16.16.139 Bcast:172.16.16.255 Mask:255.255.
          inet6 addr: fe80::20c:29ff:fe1f:a755/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:5119 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3088 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:876746 (876.7 KB) TX bytes:538083 (538.0 KB)
```

指明网卡也是用-i 参数实现：

```
sanders@ppa:~$ tcpdump –i eth0
```

这个命令会让 Tcpdump 只从 eth0 网卡中捕获流量。

一旦设置完成，你就可以开始捕获流量了。如果你监听的网卡非常繁忙，那么在你屏幕上所打出的信息可能会滚动得飞快，以至于你来不及去查看它们。这时候我们可以把抓取的包存成文件，然后只从文件中读取我们想要的数据包。

要把抓到的包存为文件，可使用-w 参数加上要保存的文件名。抓包进程会持续进行，除非你按下 Ctrl-C 组合键。流量文件会直接保存到当前执行命令的目录下，除非另指明路径。

下面就是使用 TShark 命令的一个例子：

```
C:\Program Files\Wireshark>tshark –i 1 –w packets.pcap
```

这个命令会把从 1 号网卡捕获的流量全部写到以 packets.pcap 命名的文件中。

使用 Tcpdump 时，类似的命令如下：

```
sanders@ppa:~$ tcpdump –i eth0 –w packets.pcap
```

要想从保存的文件中回读数据包，可使用-r 参数加上文件名：

```
C:\Program Files\Wireshark>tshark –r packets.pcap
```

这个命令会读取 packets.pcap 中的所有数据并把它们打印到屏幕上。

使用 Tcpdump 差不多是一样的命令。

```
sanders@ppa:~$ tcpdump –r packets.pcap
```

你也许会注意到，如果你要读取的文件包含了太多的数据包，那么你会遇到之前讲过的情况，一大堆的信息在屏幕飞快滚动以至于什么都看不清。这时你可以使用-c 参数来限制在屏幕上显示的数据包数量。

比如，使用 TShark 下面的命令只会显示在捕获文件中最开始的 10 个包。

```
C:\Program Files\Wireshark>tshark –r packets.pcap –c10
```

在 Tcpdump 里用的是一样的参数：

```
sanders@ppa:~$ tcpdump –r packets.pcap –c10
```

抓包的时候也可以使用-c 参数，这表明只会抓取前 10 个包。当和-w 参数一起使用时，可以把结果存成文件。

下面是在 TShark 中此命令的示例：

```
C:\Program Files\Wireshark>tshark –i 1 –w packets.pcap –c10
```

还有 Tcpdump 下的类似命令：

```
sanders@ppa:~$ tcpdump –i eth0 –w packets.pcap –c10
```