Developing Procedures for Network Forensics

As you have seen, log files are often examined along with forensic image files collected from devices. To get these files, you need to establish a good working relationship with network administrators and technicians. Network forensics can be a long, tedious process, and unfortunately, the trail can go cold quickly. A standard procedure often used in network forensics is as follows:

1. Always use a standard installation image for systems on a network. This image isn't a bit-stream image but an image containing all the standard applications used. You should also have MD5 and SHA-1 hash values of all application and OS files.
2. When an intrusion incident happens, make sure the vulnerability has been fixed to prevent other attacks from taking advantage of the opening.
3. Attempt to retrieve all volatile data, such as RAM and running processes, by doing a live acquisition before turning the system off.
4. Acquire the compromised drive and make a forensic image of it.
5. Compare files on the forensic image with the original installation image. Compare hash values of common files, such as `Win.exe` and standard dynamic link libraries (DLLs), and ascertain whether they have changed.

In digital forensics, you can work from the image to find most of the deleted or hidden files and partitions. Sometimes you restore the image to a physical drive so that you can run programs on the drive. In network forensics, you have to restore the drive to see how malware that attackers have installed on the system works. For example, intruders might have transmitted a Trojan program that gives them access to the system and then installed a rootkit, which is a collection of tools that can perform network reconnaissance tasks (using the `ls` or `netstat` command to collect information, for instance), keylogging, and other actions.

Reviewing Network Logs

Network logs record traffic in and out of a network. Network servers, routers, firewalls, and other devices record the activities and events that move through them. A common way of examining network traffic is running the `tcpdump` command-line program (*www.tcpdump.org*), which can produce hundreds or thousands of lines of records. A sample output is shown here:

```
TCP log from 2017-12-16:15:06:33 to 2017-12-16:15:06:34.
Fri Dec 15 15:06:33 2017; TCP; eth0; 1296 bytes; from
  204.146.114.10:1916 to 156.26.62.201:126
Fri Dec 15 15:06:33 2017; TCP; eth0; 625 bytes; from
  192.168.114.30:289 to 188.226.173.122:13
Fri Dec 15 15:06:33 2017; TCP; eth0; 2401 bytes; from
  192.168.5.41:529 to 188.226.173.122:31
Fri Dec 15 15:06:33 2017; TCP; eth0; 1296 bytes; from
  206.199.79.28:1280 to 10.253.170.210:168;first packet
```

The first line of the output is simply the header. The rest of the lines follow the format *time; protocol; interface; size; source and destination addresses*. Take another look at the second line from the previous output:

```
Fri Dec 15 15:06:33 2017; TCP; eth0; 1296 bytes; from
  204.146.114.10:1916 to 156.26.62.201:126
```

This line shows that data was transmitted on Saturday, December 16, 2017, at 15:06:33. It was a TCP packet sent via the Ethernet 0 interface of 1296 bytes. The packet was sent from 204.146.114.10:1916 to 156.26.62.201:126. In these IP addresses, the numbers after the colon represent the port number.

When viewing network logs, port information can give you clues to investigate. For example, you might notice that a particular IP address is coming in frequently on an unusual port. A receiving port above 1024, for example, should also raise a flag. You can check the Internet Assigned Numbers Authority Web site (*www.iana.org/assignments/port-numbers*) for a list of assigned port numbers.

Using a network analysis tool such as Wireshark (which you use later in this chapter), you could generate a list of the top 10 Web sites users in your network are visiting. As shown in the following output, the number of bytes being transferred is listed first, followed by the IP address of the site:

```
Top 10 External Sites Visited:
      4897 188.226.173.122
      2592 156.26.62.201
      4897 110.150.70.190
      4897 132.130.65.172
      4897 192.22.192.204
      4897 83.141.167.38
      1296 167.253.170.210
      1296 183.74.83.174
      625 6.234.186.83
      789 89.40.199.255
```

You could also generate a list of the top 10 internal users, as shown:

```
Top 10 Internal Users:
      4897 192.168.5.119
      4897 192.168.5.41
      4897 192.168.5.44
      4897 192.168.5.5
      2401 204.146.114.50
      1296 192.168.5.95
      1296 204.146.114.10
      1296 204.146.114.14
      1296 206.199.79.28
      625 192.168.5.72
```

These network logs can show you patterns, such as an employee transmitting data to or from a particular IP address frequently. Further investigation of the IP address could show that this employee is accessing an online shopping site during company time, for example.

Note

Automated software packages, such as Tripwire (*www.tripwire.com*), can tell you when suspicious network activity has occurred. Tripwire is an audit control program that detects anomalies in traffic and sends alerts automatically.

As with all investigations, keep preservation of evidence in mind. Your investigation might turn up other companies that have been compromised. In much the same way you wouldn't turn over proprietary company information to become public record, you shouldn't reveal information discovered about other companies. In these situations, the best course of action is to contact the companies and enlist their aid in tracking down network intruders. Depending on the situation, at some point you might have to report the incident to federal authorities.

Using Network Tools

A variety of tools are available for network administrators to perform remote shutdowns, monitor device use, and more. The tools covered in this chapter are a combination of freeware and enterprise software; some tools have free demo versions.

Tools such as Splunk (*www.splunk.com*), Spiceworks (*www.spiceworks.com*), Nagios (*www.nagios.org*), and Cacti (*www.cacti.net*) help you monitor your network efficiently and thoroughly. For example, you can consult records that the tool generates to prove an employee ran a program without permission. You can also monitor your network and shut down machines or processes that could be harmful.

Although these tools are helpful for network administrators, imagine what would happen if an attacker (or even an internal user) could get administrative rights to the network and start using these tools. For example, in a networking class, students had to install their own servers and then harden their systems. One student was able to log on to another student's server and shut it down remotely because no password for the default user account had been created.

Using Packet Analyzers

Packet analyzers are devices or software placed on a network to monitor traffic. Most network administrators use them for increasing security and tracking bottlenecks. However, attackers can use them to get information covertly. Most packet analyzers work at Layer 2 or 3 of the OSI model. To understand what's happening on a network, often you have to look at the higher layers by using custom software that comes with switches and routers, however.

Some analyzers perform packet captures, some are used for analysis, and some handle both tasks. Your organization needs to have policies about using these tools to comply with new federal laws on digital evidence. Windows has many tools capable of capturing and analyzing packets, but you can't feed the data they collect directly into other tools. Most tools can read anything captured in Pcap (packet capture) format. (Libpcap is the version for Linux, and Winpcap is the version for Windows.) Programs such as `tcpdump` and Wireshark (*www.wireshark.org*) use the Pcap format, for example. To take advantage of the strengths in different tools, many investigators do a capture with `tcpdump` and then analyze the capture in Wireshark.

As a forensics expert, you must choose the tool that best suits your purposes. For example, if your network is being hit with SYN (the synchronize portion of the TCP handshake) flood attacks, you want to find packets with the SYN flag set. In a SYN flood attack, the attacker keeps asking your server to establish a connection. Although your server can handle thousands of connections, it can handle only a limited number of establishing connections. To find these packets, `tcpdump` and `tethereal` (a network protocol analyzer) can be programmed to examine TCP headers to find the SYN flag. Figure 10-15 shows a TCP header; the Flags area contains several flags, including the SYN flag (denoted as S in the figure).

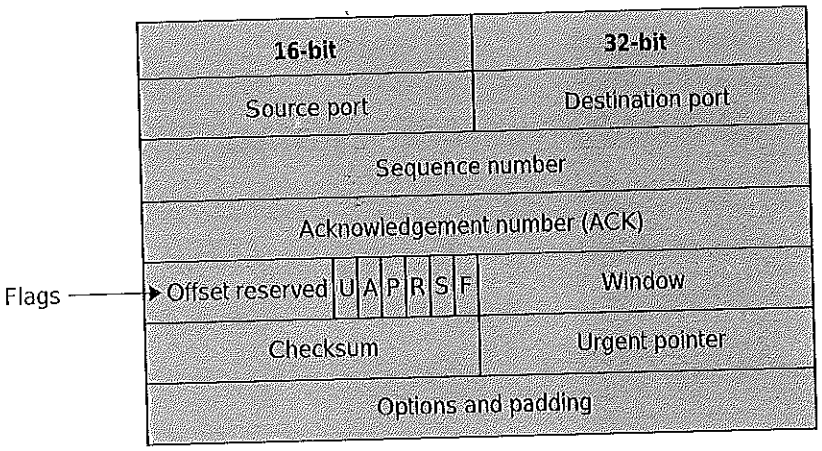

Figure 10-15 A TCP header

`Tcpslice` (*http://sourceforge.net/projects/tcpslice/*) is a good tool for extracting information from large Libpcap files; you simply specify the time frame you want to examine. It's also capable of combining files. A suite of tools called `Tcpreplay` (*http://tcpreplay.appneta.com*) can be used to replay network traffic recorded in Libpcap format; you use this information to test network devices, such as IDSs, switches, and routers.

Etherape (*http://etherape.sourceforge.net*) is a tool for viewing network traffic graphically. Another GUI tool, Netdude (*http://netdude.sourceforge.net/*), was designed as an easy-to-use interface for inspecting and analyzing large `tcpdump` files (sometimes several gigabytes). Argus (*www.qosient.com/argus*) is a session data

probe, collector, and analysis tool. This real-time flow monitor can be used for security, accounting, and network management.

Wireshark can be used in a real-time environment to open saved trace files from packet captures. An important feature is its capability to rebuild sessions. To use this feature, right-click a frame in the upper pane and click Follow TCP Stream. Wireshark then traces the packets associated with an exploit. To see how this tool works, download the most recent version of Wireshark for Windows (*www.wireshark.org/download.html*) and install it on your workstation. Then follow these steps:

1. Start Wireshark, Notice the list of networks with traffic (see Figure 10-16).

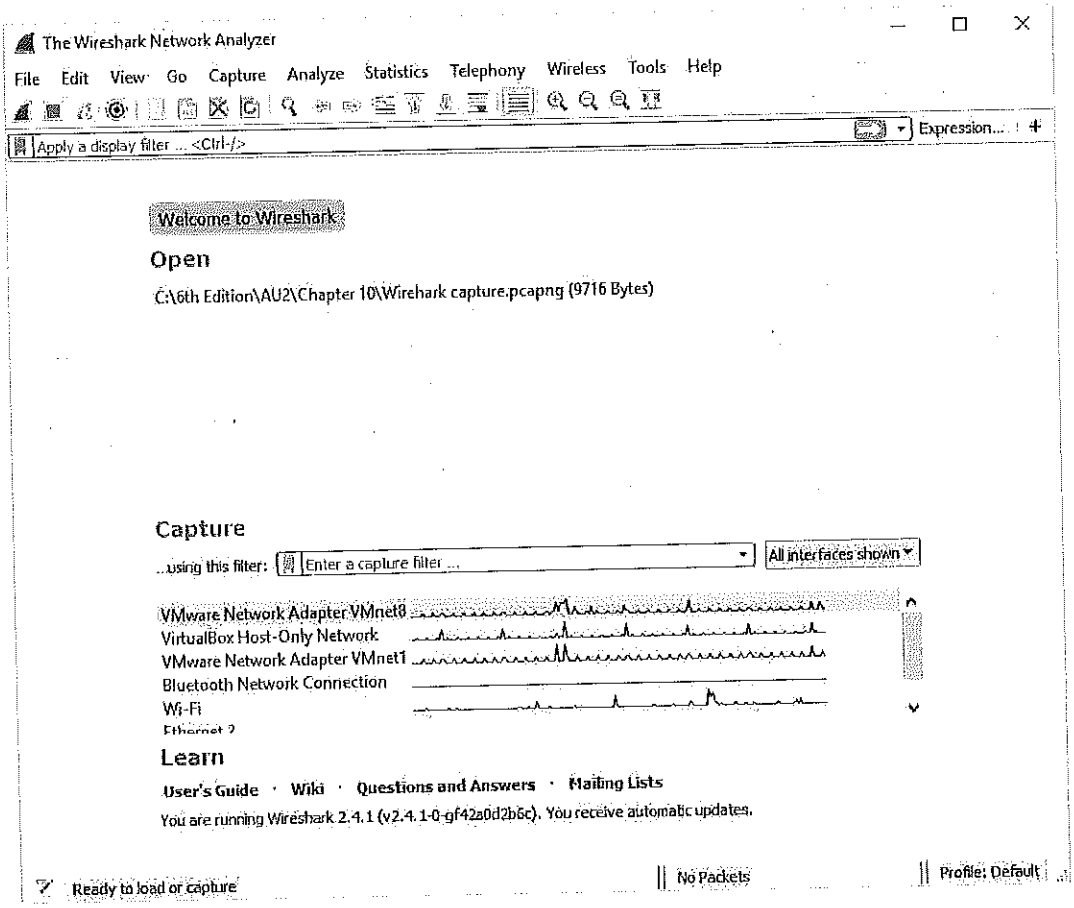

Figure 10-16 The opening window in Wireshark

Source: Wireshark Foundation, *www.wireshark.org*

2. Double-click a network that's showing activity. (If you're not on a live network, ping another student or yourself and visit some Web sites and download a file to generate traffic. Then start this activity again.)
3. After several frames have been captured, click **Stop**.

4. After the trace has been loaded, scroll through the upper pane until you see a UDP frame or an SSOP frame. Right-click the frame, point to **Follow**, and click **UDP Stream**. You should see a window similar to Figure 10-17.

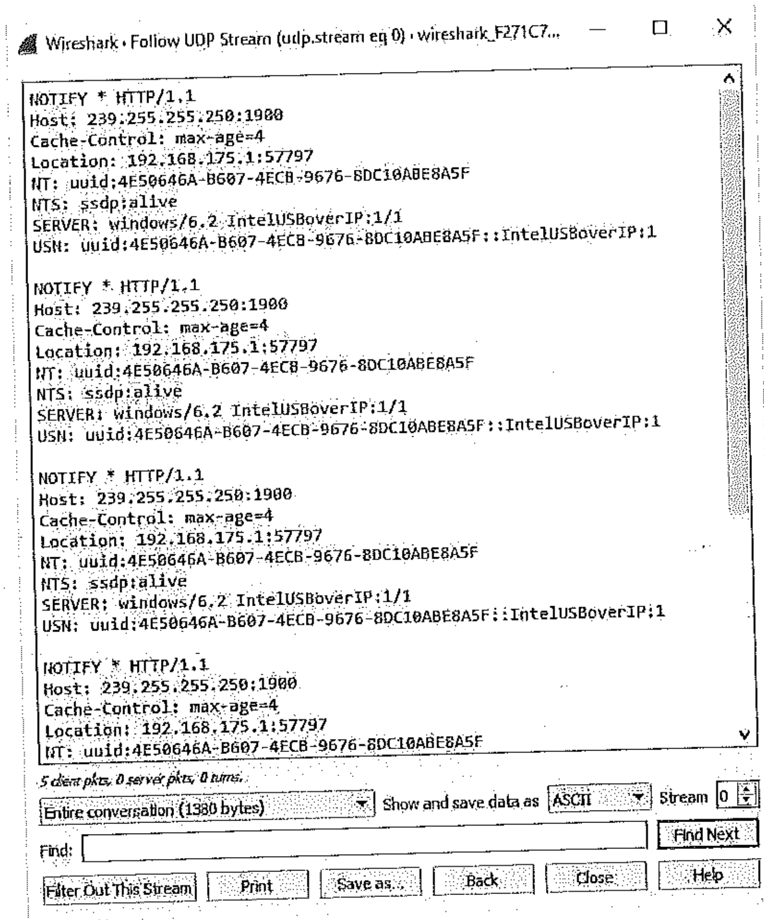

Figure 10-17 Following a UDP stream

Source: Wireshark Foundation, *www.wireshark.org*

5. Review the information in this window, and then exit Wireshark.

You can find information on network forensics tools at many of the Web sites mentioned in this chapter. If you're interested in learning even more about network forensics, the next section covers the Honeynet Project.

Investigating Virtual Networks

An article in the *Journal of Cybersecurity* explores how to modify the investigation approach that's used in physical networks so that it applies to virtual or logical networks (*www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-1439_522.pdf*). A virtual switch is a little different from a physical switch, in that there's no spanning tree between virtual switches. For example, say that 24 students each create a virtual

network, using a 192.168.x.x network on the type 1 hypervisor, such as vSphere. The numbering isn't unique to their network, but the networks don't overlap because each has its own virtual switch. There's no way to connect the virtual switches, and they don't share the same physical adapters. If there were a physical adapter, the students would have to worry about subnets, which could cause problems, such as a server on Student A's network having the same IP address as one on Student B's network.

This feature of virtual switches grew out of people needing VMs on isolated networks but residing on the same physical rack server as other virtual networks. An additional complication is that hypervisors, including Xen Server, can assign MAC addresses to virtual devices, *and* devices can have the same MAC address on different virtual networks. So Student A's VM can have a virtual Ethernet adapter or virtual network interface card with the same IP address and same MAC address as an adapter on Student B's network and even Student C's network. Each student network can't detect other student networks because of the virtual switches, which enables each student to create large networks independent of other students and is limited only by the allotted RAM and storage the student's login provides. In this setup, you could have 24 NICs with the same IP address and MAC address that are on different networks.

To take this to the next level, say you're dealing with a cloud service provider (CSP) that hosts networks for several to hundreds of companies. As stated in the *Journal of Cybersecurity* article cited at the beginning of this section, network forensics investigations in the cloud are hampered by the very qualities that make the cloud appealing—elasticity and flexibility. If needed (and it's allowed in the service level agreement), a new server can come online to deal with load balancing. In addition, automatic failovers are in place, which may or may not be in the same physical location as the server. Add to that hundreds or even thousands of NICs with the same IP address and MAC address, and you can see that traditional physical network forensics can't handle these arrangements.

Wireshark and Network Miner are two tools that are capable of analyzing virtual networks. As these networks become more complex, newer or updated tools will be needed.

Examining the Honeynet Project

The Honeynet Project (*www.honeynet.org*) was developed to make information widely available in an attempt to thwart Internet and network attackers. Many people participate in this worldwide project, which is now a nonprofit organization. The objectives are awareness, information, and tools. The first step is to make people and organizations aware that threats exist and they might be targets. The second is to provide information on how to protect against these threats, including how attackers operate, how they communicate, and what tactics they use. Finally, for people who want to do their own research, the Honeynet Project offers tools and methods.

A major threat is **distributed denial-of-service (DDoS)** attacks. A trace of a DDoS attack might go through other organizations' networks, not just yours or your ISP's.

In DDoS attacks, hundreds or even thousands of machines can be used. These machines are known as **zombies** because they have unwittingly become part of the attack. When the first DDoS attacks began, the main concerns were the high monetary impact and the amount of time it took to track down these attacks.

Another major threat is **zero day attacks**. Attackers look for holes in networks and OSs and exploit these weaknesses before patches are available. Vendors usually aren't aware that these vulnerabilities exist, so they haven't developed and released patches for them. Penetration testers attempt to break into networks to find undiscovered vulnerabilities and then predict where the next onslaught of network attacks will come from.

In any organization, you have to determine the value of the data you're protecting and weigh it against the price of the defense system you plan to install. When an attack strikes, your first response is to stop it and prevent it from going further. Then you need to see what defense procedures worked and what additional procedures might be needed. Training and informing IT staff are critical.

The Honeynet Project was set up as a resource to help network administrators deal with DDoS and other attacks. It involves installing honeypots and honeywalls at different locations in the world. A **honeypot** is a computer set up to look like any other machine on your network; its purpose is to lure attackers to your network, but it contains no information of real value. You can take the honeypot offline to analyze it and not affect the running of your network. **Honeywalls** are computers set up to monitor what's happening to honeypots on your network and record what attackers are doing (see *www.honeynet.org/papers/cdrom/*). Honeypots and honeywalls are commonly used to attract intruders and see what they're attempting to do on a network.

Chapter Summary

- Virtual machines are used extensively in organizations and are a common part of forensic investigations. Investigators must be familiar with file extensions that indicate the existence of VMs.
- There are two types of hypervisors for running virtual machines. Type 1 hypervisors contain their own OSs and are loaded directly on physical hardware. Type 2 hypervisors are applications installed on top of an OS.
- Virtualization Technology is Intel's CPU design for security and performance enhancements that enable the BIOS to support virtualization. Virtual Machine Extensions (VMX) are instruction sets created for Intel processors to handle virtualization. Intel now has virtualization technology for memory, I/O, graphics, and security.