12.6.7 Packet Capture Facts

This lesson covers the following topics:

- Packet capturing
- Switched network sniffing
- Wireshark
- TCPDump
- TCPReplay
- Additional sniffing tools

Packet Capturing

Packet capturing, also referred to as sniffing, is the process of collecting information as it crosses the network. Sniffing is similar to eavesdropping or wiretapping. It can be active or passive. Monitoring traffic is passive sniffing. Altering traffic in any way is active sniffing.

For sniffing to be effective, the network interface must be in promiscuous mode. Normally, an interface is set to grab frames that are directed only to its own MAC address. Turning on promiscuous mode gives the interface permission to grab every frame that comes its way, even if the frame is addressed to someone else. A lot of information can be gathered during this process. Attackers examine each packet closely to see which ones are useful.

There are several tools an attacker can use to make this job much easier, But's important to know what attackers focus on. One key area of focus is packets that are sent with less-secure protocols. Many protocols were designed with the concept that encryption happens at another layer. For example:

- SMTP was designed to deliver an email message without encrypting it.
- POP3 was designed to retrieve emails, therefore passwords and usernames are easy to intercept from it.
- FTP was designed to transmit files; all FTP traffic is sent in clear text.
- IMAP, HTTP, and Telnet send passwords and data using clear text.

Switched Network Sniffing

Networks that include switches can provide an initial challenge to an attacker because switches prevent sniffing an entire network. The following table lists methods an attacker can use to sniff out portions of a network.

Method	Description
MAC spoofing	A common low-level security measure is port security. Port security allows only specific MAC addresses to access a switch. The goal is to ensure that only authorized devices have access to the network. A MAC address for a network interface card (NIC) is assigned by the manufacturer. This address is hard-coded directly into the NIC and can't be changed. However, on some interfaces it is possible to change the MAC address of the interface driver. This allows an attacker's computer to connect to a switch using an authorized MAC address. This allows the attacker to capture packets from that network.
MAC flooding	When a switch is initially turned on, it doesn't know which devices it will support. A switch uses a content addressable memory (CAM) table to track MAC addresses. As it receives packets from

	various MAC addresses, it adds the addresses to its CAM table and associates each one with a physical port on the switch. This process allows data to be sent directly to the port where the intended recipient is located instead of sending all data across the entire network like a hub. Although one port can have multiple MAC addresses associated with it, the CAM table has a size limit. MAC flooding is the process of intentionally flooding the CAM table with Ethernet frames, each originating from a different MAC address. Once the table starts to overflow, the switch responds by broadcasting all incoming data to all ports, basically turning itself into a hub instead of a switch. When an attacker's MAC address is connected to one of the ports, the attacker can capture all traffic as it is broadcast across the network.
ARP poisoning	The Address Resolution Protocol (ARP) maps IP addresses to MAC addresses and provides the most efficient path for data transmission. ARP broadcasts are permitted to freely roam the network. An attacker can use the free flow of traffic for an advantage. By sending spoofed messages onto a network, the MAC address of the attacker can be associated with the IP address of another host, preferably the default gateway. As a result, the target machine will send frames to the attacker's system, thinking that it is the gateway. The attacker will then typically forward the frame to the original destination.
Port mirroring	Port mirroring can be challenging to set up, but is possible depending on the level of access an attacker has on a network. The concept behind port mirroring, also known as SPAN port, is simple. Port mirroring creates a duplicate of all network traffic on a port and sends it to another device. If all traffic from a target machine is directed through the switch to the server, an attacker can implement port mirroring. Port mirroring ensures that all traffic is sent to the attacker’s machine as well as the target machine.

Wireshark

Wireshark is one of the most well-known packet analyzers. It is available for Windows, Mac, and Linux operating systems. Wireshark has numerous tools that can be used to capture and analyze traffic. It includes search and filtering capabilities that make it a very powerful resource. These filtering commands can be typed into the filter window. The screen will display only the filtered data.

The following table lists commonly used filters:

Operator	Description
==	Equal (example: ip.addr == 192.168.1.3)
eq	Equal (example: tcp.port eq 161)
contains	Contains a specific value (example: http contains “http://www.stuff.com”
ne	Not equal (example: ip.src ne 192.168.1.3)
!=	Not equal (example: ip.addr != 192.168.1.3
&&	And (example ip.addr==192.168.1.3&&tcp.port=23)

or	Or (example ip.addr==192.168.1.3 or ip.addr ==192.168.1.4)

TCPDump

TCPDump is a command line sniffer designed for the Linux environment. This tool filters the contents of packets going through a network interface. TCPDump has several switches and options, a few of which you'll find in the following table.

Operator	Description
-i	Puts an interface into listening mode.
-w	Specifies the file the data should be saved in.
-a	Requests that ASCII strings are included in the output.
-x	Requests that ASCII and hexadecimal strings are included in the output.
-v	Turns on verbosity.
-n	Turns off DNS lookups.
dst	Requests that all traffic going to a specified destination is captured.
src	Requests that all information coming from a specified source is captured.
host	Requests that all traffic going to a specified destination and from a specified source is captured.
pcap	Requests that captured content be saved to a specified file.

TCPReplay

One of the most critical aspects of a cybersecurity plan is testing. Organizations may employ white-hat hackers to attempt attacks and report their findings so network managers can reconfigure security devices, address vulnerabilities, and mitigate potential risk. One tool that can be used to simulate attacks is TCPReplay. Once a packet capture is performed by TCPDump or Wireshark, it can be sent out again and again to test devices such as firewalls, IDS/IPS, and NetFlow, as well as infrastructure equipment such as switches and routers.

Using TCPDump or Wireshark, a simulated attack can be executed to capture packets. After manipulating the packet capture as needed, you can test defensive equipment replaying the attack with TCPReplay. After the test, you can analyze the results for alerts, detection, or prevention. The test will identify remediation points and areas that need reconfiguration. After remediation, the test can be replayed again, and the results are reevaluated. This can happen multiple times until network management deems the vulnerability closed.

TCPReplay has several switches and options, a few of which you'll find in the following table.

Operator	Description

-K	Takes information from the packet capture file and preloads it into RAM for faster access. Use this only if your system contains enough RAM.
-M	Replays packets at a given Mbps (megabits per second) rate. This parameter can be used to throttle the output.
-L *number*	Limits the number of packets sent. This can be used for limited tests to ensure that the system is set up correctly and to perform a quick end-to-end test.
-d *number*	Enables the verbosity for debug output. Used along with the enable-debug flag, the number can be in the range of 0 through 5. Higher numbers increase verbosity.

TCPReplay can be a helpful tool used to test the success or failure of security equipment and infrastructure. When used correctly, it provides network management with the information needed to keep an organization as safe as possible against attack. It provides vulnerability data that can be remediated and tested again until the risk is minimized.

Additional Sniffing Tools

The following table describes additional tools.

Tool	Description
Cain and Abel	*Cain and Abel* is a collection of tools including ARP poisoning. Cain and Abel redirects packets from a target by forging ARP replies.
Ufasoft Snif	*Ufasoft Snif* is a network sniffer used to capture, decrypt, and analyze packets as they travel across the network.
WinARPAttacker	*WinARPAttacker* can scan, detect, and even attack computers on a LAN.
Ettercap	*Ettercap* has multiple sniffing functions and can be used for ARP poisoning, passive sniffing, packet grabbing, and protocol decoding.
Etherflood	*Etherflood* is a tool that can flood a switched network with random MAC addresses.
SMAC	*SMAC* is a spoofing tool that allows an attacker to spoof a MAC address to any value.
WinDump	*WinDump* is the Windows version of TCPDump.