10.1.2 Sniffer Facts

This lesson covers the following topics:

- Sniffing
- Switched network sniffing
- Wireshark
- TCPDump
- Additional sniffing tools

Sniffing

Sniffing is the process of collecting information as it crosses the network. Sniffing is similar to eavesdropping or wiretapping and can be active or passive. If you're simply monitoring traffic, that is passive sniffing. If you alter traffic in any way, that is active sniffing. For sniffing to be effective, you want to put your network interface into promiscuous mode. Normally, an interface is set to only grab onto frames that are directed to its MAC address. Turning on promiscuous mode gives the interface permission to grab onto every frame that comes its way, even if it's addressed to someone else. A lot of information can be gathered during this process, so you will need to examine each packet closely to see which ones are useful.

Thankfully, there are tools that can help make this job much easier, but it's still important to know what to keep an eye out for. First, you'll want to focus on packets that are being sent with less-secure protocols. Luckily for you, the hacker, some protocols weren't designed to be overly secure. SMTP, for example, was designed to deliver an email message with the hopes that encryption happened at another layer. Similarly, POP3 was simply designed to retrieve emails; passwords and usernames are easy to intercept from it. FTP was designed to transmit files, all of which are sent in clear form. Other vulnerable protocols include IMAP, HTTP, and Telnet. Passwords and data are sent over clear text, once again in hopes that encryption is happening at a different layer. Second, when examining packets, you'll want to keep an eye on the source and destination IP addresses. The IP addresses will, most likely, be listed in hexadecimal format, so you'll want to refresh your hex to standard conversion skills if you haven't done so in a while.

Switched Network Sniffing

Networks that include switches can provide an initial challenge. You won't be able to sniff an entire network, but in the table below, you will find a few methods that can help you sniff out portions of the network.

Method	Description
MAC spoofing	A common low-level security measure is port security. Port security allows only specific MAC addresses access to 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, it is possible to change the MAC address of the interface driver. Let's say you want to access a network, but the administrator has implemented port security measures. Thanks to your previous reconnaissance and scanning, you know that your target computer has access to the network, and you even know the MAC address. Using one of several software tools, you can spoof your computer's MAC address to look like the target's MAC address, and you can connect directly to the network with minimal effort.
MAC flooding	When a switch is initially turned on, it doesn't know which devices it's going to be supporting. A switch tracks MAC addresses in a content addressable memory (CAM) table. 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 is only so big. As a hacker, you can use a method called MAC flooding to intentionally flood the CAM table with Ethernet frames, each originating from different MAC addresses. 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. Since your MAC address is now connected to one of the ports, you are able to capture all traffic as it is broadcast across the network.
ARP poisoning	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 around the network. You can use this free flow of traffic to your advantage. By sending spoofed messages onto a network, you can associate your MAC address with the IP address of another host, preferably the default gateway. As a result, the target machine will send frames to your system, thinking that you are their gateway, before you forward them on to the original destination.
Port mirroring	Port mirroring can be challenging to set up, but is possible depending on the level of access you've been able to obtain to a network. The concept behind port mirroring, also known as SPAN port, is actually pretty 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, you can implement port mirroring. Port mirroring ensures that any time the data comes through, it is duplicated and sent out to the attacker's machine as well.

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, and the screen will only display what you have selected. The following table lists the filters you are most likely to use:

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 provides information on the contents of packets on a network interface that match a given filter. TCPDump has several switches and options, a few of which you’ll find in the table below:

Operator	Description
-i	Puts an interface into listening mode.
-w	Specifies which file the data should be saved in.
-a	Requests that asci strings are included in the output.
-x	Requests that asci and hexadecimal strings are included in the output.
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.

Additional Sniffing 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* is a sniffing tool. It has multiple 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.