

Covers all Exam Objectives for CEHv6



Includes Real-World Scenarios, Hands-On Exercises, and Leading-Edge Exam Prep Software Featuring:

- Custom Test Engine
- Hundreds of Sample Questions
- Electronic Flashcards
- Entire Book in PDF

CEH™

Certified Ethical Hacker

STUDY GUIDE


Exam 312-50
Exam EC0-350

Kimberly Graves



SERIOUS SKILLS.

Chapter 6. Gathering Data from Networks: Sniffers.....	1
Section 6.1. Understanding Host-to-Host Communication.....	2
Section 6.2. How a Sniffer Works.....	6
Section 6.3. Sniffing Countermeasures.....	6
Section 6.4. Bypassing the Limitations of Switches.....	7
Section 6.5. Wireshark Filters.....	9
Section 6.6. Understanding MAC Flooding and DNS Spoofing.....	12
Section 6.7. Summary.....	14
Section 6.8. Exam Essentials.....	15
Section 6.9. Review Questions.....	16
Section 6.10. Answers to Review Questions.....	19



Chapter 6

Gathering Data from Networks: Sniffers

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ Understand the protocol susceptible to sniffing
- ✓ Understand active and passive sniffing
- ✓ Understand ARP poisoning
- ✓ Understand ethereal capture and display filters
- ✓ Understand MAC flooding
- ✓ Understand DNS spoofing techniques
- ✓ Describe sniffing countermeasures



A *sniffer* is a packet-capturing or frame-capturing tool. It basically captures and displays the data as it is being transmitted from host to host on the network. Generally a sniffer intercepts traffic on the network and displays it in either a command-line or GUI format for a hacker to view. Most sniffers display both the Layer 2 (frame) or Layer 3 (packet) headers and the data payload. Some sophisticated sniffers interpret the packets and can reassemble the packet stream into the original data, such as an email or a document.

Sniffers are used to capture traffic sent between two systems, but they can also provide a lot of other information. Depending on how the sniffer is used and the security measures in place, a hacker can use a sniffer to discover usernames, passwords, and other confidential information transmitted on the network. Several hacking attacks and various hacking tools require the use of a sniffer to obtain important information sent from the target system. This chapter will describe how sniffers work and identify the most common sniffer hacking tools.



The term *packet* refers to the data at Layer 3, or the Network layer, of the OSI model, whereas *frame* refers to data at Layer 2, or the Data Link layer. Frames contain MAC addresses, and packets contain IP addresses.

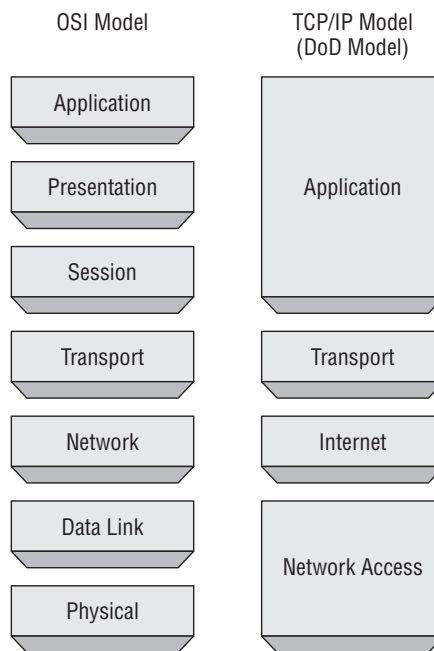
Understanding Host-to-Host Communication

All Host-to-Host network communications is based upon the TCP/IP Data Communications Model. The TCP/IP Model is a 4 layer model. The TCP/IP Model maps to the older OSI model with 7 layers of data communication. Most applications use the TCP/IP suite for host-to-host data communications. See Figure 6-1.

In normal network operations, the application layer data is encapsulated and a header containing address information is added to the beginning of the data. An IP header containing source and destination IP address are added to the data as well as a MAC header

containing source and destination MAC addresses. IP addresses are used to route traffic to the appropriate IP network, and the MAC addresses ensure the data is sent to the correct host on the destination IP network. In this manner, traffic is sent from source host to destination host across the Internet and delivery to the correct host is ensured. The postal system works much the same way. Mail is routed to the appropriate area using the zip code, and then the mail is delivered within the zip code to the street and house number. The IP address is similar to the zip code to deliver mail to the regional area, and the street and house numbers are like the MAC address of that exact station on the network.

FIGURE 6.1 TCP/IP Model



The address system ensures accurate delivery to the receiver. In normal network operations, a host should not receive data intended for another host as the data packet should only be received by the intended receiver. Simply said, the data should only be received by the station with the correct IP and MAC address. However, we know that sniffers do receive data not intended for them.



Real World Scenario

What Does Mail Delivery Have to Do with Hacking?

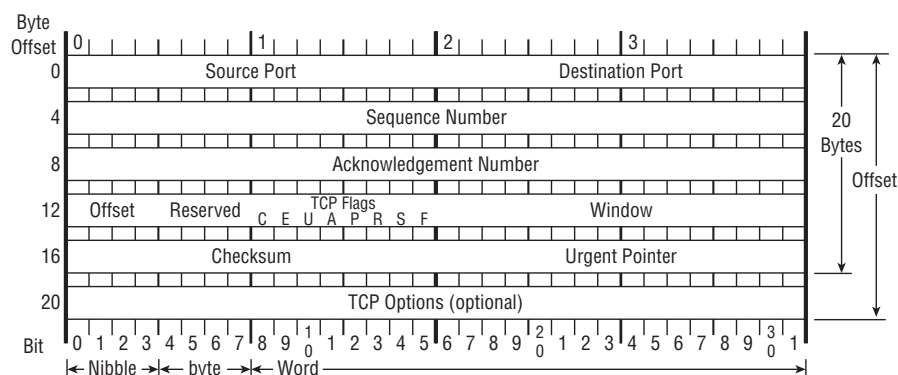
In the real world, sometimes mail is not delivered to the intended receiver. I'm sure you have all opened your mailbox to discover an envelope addressed to your neighbor or someone who used to live at your address. This happens on a fairly regular basis at my house. Most people will just leave the mail in the box for the postal carrier to redeliver or physically take the envelope to a neighbor. This same type of situation can occur in computer networking, where application layer data does not reach its intended recipient because of a delivery error or other network fault.

Another cause of mail not being received by the intended recipient is someone is performing reconnaissance and watching your mailbox. Let's assume you are not home and the postal carrier delivers your mail to the mailbox. Someone watching the mailbox from down the street or a nearby building could wait for the mail to be delivered to the mailbox, and they go take the mail or just a particular envelope out of the box. This would be especially effective if the hacker performed some reconnaissance and knew what time each day the mail was delivered. The hacker could then examine and read the information in the envelope, and if they were trying to cover their tracks simply reseal the envelope and put it back in the mailbox.

Sniffing data on a network occurs in much the same way. Data is intercepted, read, and either sent on to the intended recipient or just discarded.

In addition to understanding network addresses, it is also important to understand the format of the TCP Header. Figure 6.2 shows the TCP Header format.

FIGURE 6.2 TCP Header Format



The TCP Header is comprised of the following fields:

Source Port: 16 bits The source port number.

Destination Port: 16 bits The destination port number.

Sequence Number: 32 bits The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number: 32 bits If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive.

Data Offset: 4 bits The number of 32 bit words in the TCP Header. This indicates where the data begins.

Reserved: 6 bits Reserved for future use. Must be zero.

Control Bits: 6 bits

- URG: Urgent Pointer field significant
- ACK: Acknowledgment field significant
- PSH: Push Function
- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender

Window: 16 bits The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

Checksum: 16 bits The checksum field is a computation of all fields to ensure all data was received and the data was not modified in transit.

Urgent Pointer: 16 bits This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Options: variable Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length.

When referring to the length of the fields in the TCP Header, 8 bits comprises a single byte. A Nibble is less than a byte and a Word is more than a byte.

In the next section we will explore how a hacking tool manipulates normal network operations in order to capture traffic on a host that is not the intended receiver.

How a Sniffer Works

Sniffer software works by capturing packets not destined for the sniffer system's MAC address but rather for a target's destination MAC address. This is known as *promiscuous mode*. Normally, a system on the network reads and responds only to traffic sent directly to its MAC address. However, many hacking tools change the system's NIC to promiscuous mode. In promiscuous mode, a NIC reads all traffic and sends it to the sniffer for processing. Promiscuous mode is enabled on a network card with the installation of special driver software. Many of the hacking tools for sniffing include a promiscuous-mode driver to facilitate this process. Not all Windows drivers support promiscuous mode, so when using hacking tools ensure that the driver will support the necessary mode.

Any protocols that don't encrypt data are susceptible to sniffing. Protocols such as HTTP, POP3, Simple Network Management Protocol (SNMP), and FTP are most commonly captured using a sniffer and viewed by a hacker to gather valuable information such as usernames and passwords.

There are two different types of sniffing: passive and active. *Passive sniffing* involves listening and capturing traffic, and is useful in a network connected by hubs; *active sniffing* involves launching an Address Resolution Protocol (ARP) spoofing or traffic-flooding attack against a switch in order to capture traffic. As the names indicate, active sniffing is detectable but passive sniffing is not detectable.

In networks that use hubs or wireless media to connect systems, all hosts on the network can see all traffic; therefore, a passive packet sniffer can capture traffic going to and from all hosts connected via the hub. A switched network operates differently. The switch looks at the data sent to it and tries to forward packets to their intended recipients based on MAC address. The switch maintains a MAC table of all the systems and the port numbers to which they're connected. This enables the switch to segment the network traffic and send traffic only to the correct destination MAC addresses. A switch network has greatly improved throughput and is more secure than a shared network connected via hubs.

Another way to sniff data through a switch is to use a span port or port mirroring to enable all data sent to a physical switch port to be duplicated to another port. In many cases, span ports are used by network administrators to monitor traffic for legitimate purposes.

Sniffing Countermeasures

The best security defense against a sniffer on the network is encryption. Although encryption won't prevent sniffing, it renders any data captured during the sniffing attack useless because hackers can't interpret the information. Encryption such as AES and RC4 or RC5 can be utilized in VPN technologies and is commonly used to prevent sniffing on a network.

Countermeasure Tools

NetIntercept is a spam and virus firewall. It has advanced filtering options and can learn and adapt as it identifies new spam. It also intercepts and quarantines the latest email viruses and Trojans, preventing a Trojan from being installed and possibly installing a sniffer.

Sniffdet is a set of tests for remote sniffer detection in TCP/IP network environments. Sniffdet implements various tests for the detection of machines running in promiscuous mode or with a sniffer.

WinTCPKill is a TCP connection termination tool for Windows. The tool requires the ability to use a sniffer to sniff incoming and outgoing traffic of the target. In a switched network, WinTCPKill can use an ARP cache-poisoning tool that performs ARP spoofing.

Bypassing the Limitations of Switches

Because of the way Ethernet switches operate, it is more difficult to gather useful information when sniffing on a switched network. Since most modern networks have been upgraded from hub to switches, it takes a little more effort to sniff on a switched network. One of the ways to do that is to trick the switch into sending the data to the hackers' computer using ARP poisoning.

How ARP Works

ARP allows the network to translate IP addresses into MAC addresses. When one host using TCP/IP on a LAN tries to contact another, it needs the MAC address or hardware address of the host it's trying to reach. It first looks in its ARP cache to see if it already has the MAC address; if it doesn't, it broadcasts an ARP request asking, "Who has the IP address I'm looking for?" If the host that has that IP address hears the ARP query, it responds with its own MAC address, and a conversation can begin using TCP/IP.

ARP poisoning is a technique that's used to attack an Ethernet network and that may let an attacker sniff data frames on a switched LAN or stop the traffic altogether. ARP poisoning utilizes ARP spoofing, where the purpose is to send fake, or spoofed, ARP messages to an Ethernet LAN. These frames contain false MAC addresses that confuse network devices such as network switches. As a result, frames intended for one machine can be mistakenly sent to another (allowing the packets to be sniffed) or to an unreachable host (a denial-of-service, or DoS, attack). ARP spoofing can also be used in a man-in-the-middle attack, in which all traffic is forwarded through a host by means of ARP spoofing and analyzed for passwords and other information.

ARP Spoofing and Poisoning Countermeasures

To prevent ARP spoofing, permanently add the MAC address of the gateway to the ARP cache on a system. You can do this on a Windows system by using the `ARP -s` command at the command line and appending the gateway's IP and MAC addresses. Doing so prevents a hacker from overwriting the ARP cache to perform ARP spoofing on the system but can be difficult to manage in a large environment because of the number of systems. In an enterprise environment, port-based security can be enabled on a switch to allow only one MAC address per switch port.

In Exercise 6.1 you will use Wireshark to sniff traffic.

EXERCISE 6.1

Use Wireshark to Sniff Traffic

1. Download and install the latest stable version of Wireshark from www.wireshark.org.
2. Click on the Capture menu and then select interfaces.



3. Click the Start button next to the interface that shows packets being sent and received. If you have multiple interfaces with packet activity, choose one of them—preferably the interface with the most activity.
4. Click on a packet to analyze that single packet. The detailed headers will be displayed beneath the packet capture screen.
5. Expand each header (IP, TCP) of a packet and identify the address information.

This exercise will provide much more network traffic if performed on a hub rather than a switch. A wireless network can be used, as a wireless LAN is a shared network segment similar to how a hub operates.

Hacking Tools

Wireshark is a freeware sniffer that can capture packets from a wired or wireless LAN connection. The software was previously called Ethereal. Wireshark is a common and popular program because it is free, but it has some drawbacks. An untrained user may find it difficult to write filters in Wireshark to capture only certain types of traffic.

Snort is an intrusion detection system (IDS) that also has sniffer capabilities. It can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, Common Gateway Interface (CGI) attacks, Server Message Block (SMB) probes, and OS fingerprinting attempts.

WinDump is the Windows version of tcpdump, the command-line network analyzer for Unix. WinDump is fully compatible with tcpdump and can be used to watch, diagnose, and save to disk network traffic according to various rules.

EtherPeek is a great sniffer for wired networks with extensive filtering and TCP/IP conversation tracking capabilities. The latest version of EtherPeek has been renamed OmniPeek.

WinSniffer is an efficient password sniffer. It monitors incoming and outgoing network traffic and decodes FTP, POP3, HTTP, ICQ, Simple Mail Transfer Protocol (SMTP), telnet, Internet Message Access Protocol (IMAP), and Network News Transfer Protocol (NNTP) usernames and passwords.

Iris is an advanced data- and network-traffic analyzer that collects, stores, organizes, and reports all data traffic on a network. Unlike other network sniffers, Iris is able to reconstruct network traffic, such as graphics, documents, and emails including attachments.

Wireshark Filters

Wireshark is a freeware sniffer that can capture packets from a wired or wireless LAN connection. It is a very powerful tool which can provide network and upper layer protocol data captured on a network. Like a lot of other network programs, Wireshark uses the pcap network library to capture packets.

Wireshark was called Ethereal until 2006 when the main developer decided to change its name because of copyright reasons with the Ethereal name, which was registered by the company he decided to leave in 2006.

In Exercise 6.1 you installed and began capturing packets using Wireshark. To narrow down the amount of information gathered by Wireshark, you can use filters. These filters limit the amount of information captured or displayed.

Here are some examples of Wireshark filters:

ip.dst eq www.eccouncil.org This sets the filter to capture only packets destined for the web server `www.eccouncil.org`.

ip.src == 192.168.1.1 This sets the filter to capture only packets coming from the host `192.168.1.1`.

eth.dst eq ff:ff:ff:ff:ff:ff This sets the filter to capture only Layer 2 broadcast packets.

host 172.18.5.4 This sets the filter to capture only traffic to or from IP address `172.18.5.4`.

net 192.168.0.0/24 This sets the filter to capture traffic to or from a range of IP addresses.

port 80 This sets the filter to capture traffic to destination port 80 (HTTP).

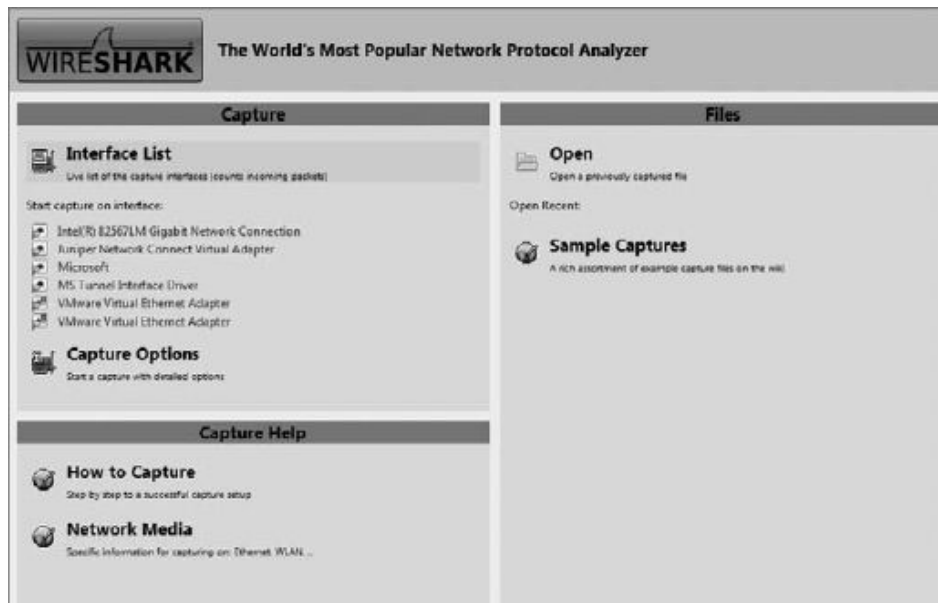
port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420 This sets the filter to capture HTTP GET requests. The filter looks for the bytes “G”, “E”, “T”, and “ ” (hex values 47, 45, 54, and 20) just after the TCP header. “`tcp[12:1] & 0xf0 >> 2`” figures out the TCP header length.

Exercise 6.2 shows you how to write filters in Wireshark.

EXERCISE 6.2

Create a Wireshark filter to capture only traffic to or from an IP address

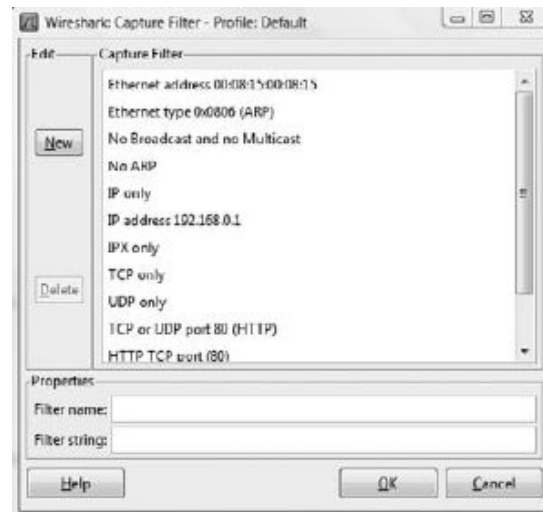
1. Open Wireshark.



2. Click the active Network Interface to capture traffic.

EXERCISE 6.2 (continued)

3. Click Capture, then select filters.



4. Click the new button to create a new filter.
5. Name the new filter in the filter name field.
6. Type **host IPaddress** in the filter string field.
7. Click OK.
8. Select the capture menu and click start to begin the capture.

Repeat the above steps to create filters using the following strings:

net 192.168.0.0/24 To capture traffic to or from a range of IP addresses.

src net 192.168.0.0/24 To capture traffic from a range of IP addresses.

dst net 192.168.0.0/24 To capture traffic to a range of IP addresses.

port 53 To capture only DNS (port 53) traffic.

host www.example.com and not (port 80 or port 25) To capture non-HTTP and non-SMTP traffic on your server.

port not 53 and not arp To capture all except ARP and DNS traffic.

tcp portrange 1501-1549 To capture traffic within a range of ports.

not broadcast and not multicast Capture only unicast traffic. Useful to get rid of noise on the network if you only want to see traffic to and from your machine.

EXERCISE 6.2 (continued)

Practice writing filters in Wireshark that capture only one type of protocol traffic or traffic from a specific source IP or MAC address. Use your PC's IP or MAC address to test that the filter is working.

It's important to understand how to create these filters before you attempt the CEH exam.

Understanding MAC Flooding and DNS Spoofing

A packet sniffer on a switched network can't capture all traffic as it can on a hub network; instead, it captures traffic either coming from or going to the system. It's necessary to use an additional tool to capture all traffic on a switched network. There are essentially two ways to perform active sniffing and make the switch send traffic to the system running the sniffer:

ARP Spoofing This method involves using the MAC address of the network gateway and consequently receiving all traffic intended for the gateway on the sniffer system. A hacker can also *flood* a switch with so much traffic that it stops operating as a switch and instead reverts to acting as a hub, sending all traffic to all ports. This active sniffing attack allows the system with the sniffer to capture all traffic on the network.



Many switches have been patched or redesigned to not be susceptible to the flooding vulnerability.

DNS Spoofing (or DNS Poisoning) This is a technique that tricks a DNS server into believing it has received authentic information when in reality it hasn't. Once the DNS server has been poisoned, the information is generally cached for a while, spreading the effect of the attack to the users of the server. When a user requests a certain website URL, the address is looked up on a DNS server to find the corresponding IP address. If the DNS server has been compromised, the user is redirected to a website other than the one that was requested, such as a fake website.

To perform a DNS attack, the attacker exploits a flaw in the DNS server software that can make it accept incorrect information. If the server doesn't correctly validate DNS responses to ensure that they come from an authoritative source, the server ends up caching the incorrect entries locally and serving them to users that make subsequent requests.

This technique can be used to replace arbitrary content for a set of victims with content of an attacker's choosing. For example, an attacker poisons the IP address's DNS entries

for a target website on a given DNS server, replacing them with the IP address of a server the hacker controls. The hacker then creates fake entries for files on this server with names matching those on the target server. These files may contain malicious content, such as a worm or a virus. A user whose computer has referenced the poisoned DNS server is tricked into thinking the content comes from the target server and unknowingly downloads malicious content.

The types of DNS spoofing techniques are as follows:

Intranet Spoofing Acting as a device on the same internal network

Internet Spoofing Acting as a device on the Internet

Proxy Server DNS Poisoning Modifying the DNS entries on a proxy server so the user is redirected to a different host system

DNS Cache Poisoning Modifying the DNS entries on any system so the user is redirected to a different host

Hacking Tools

EtherFlood is used to flood an Ethernet switch with traffic to make it revert to a hub. By doing this, a hacker is able to capture all traffic on the network rather than just traffic going to and from their system, as would be the case with a switch.

Dsniff is a collection of Unix-executable tools designed to perform network auditing as well as network penetration. The following tools are contained in dsniff: filesnarf, mailsnarf, msgsnarf, urlsnarf, and websp. These tools passively monitor a vulnerable shared network (such as a LAN where the sniffer sits behind any exterior firewall) for interesting data (passwords, email, files, and so on).

Sshmitm and webmitm implement active man-in-the-middle attacks against redirected Secure Shell (SSH) and HTTPS sessions.

Arpspoof, dnsspoof, and macof work on the interception of switched network traffic that is usually unavailable to a sniffer program because of switching. To get around the Layer 2 packet-switching issue, dsniff spoofs the network into thinking that it's a gateway that data must pass through to get outside the network.

IP Restrictions Scanner (IRS) is used to find the IP restrictions that have been set for a particular service on a host. It combines ARP poisoning with a TCP stealth or half-scan technique and exhaustively tests all possible spoofed TCP connections to the selected port of the target. IRS can find servers and network devices like routers and switches and identify access-control features like access control lists (ACLs), IP filters, and firewall rules.

sTerm is a telnet client with a unique feature: it can establish a bidirectional telnet session to a target host, without ever sending the real IP and MAC addresses in any packet. Using ARP poisoning, MAC spoofing, and IP spoofing techniques, sTerm can effectively bypass ACLs, firewall rules, and IP restrictions on servers and network devices.

Cain & Abel is a multipurpose hacking tool for Windows. It allows easy recovery of various kinds of passwords by sniffing the network; cracking encrypted passwords using dictionary or brute-force attacks; recording Voice over IP, or VoIP, conversations; decoding scrambled passwords; revealing password boxes; uncovering cached passwords; and analyzing routing protocols. The latest version contains a lot of new features like ARP Poison Routing (APR), which enables sniffing on switched LANs and man-in-the-middle attacks. The sniffer in this version can also analyze encrypted protocols such as SSH-1 and HTTPS, and it contains filters to capture credentials from a wide range of authentication mechanisms.

Packet Crafter is a tool used to create custom TCP/IP/UDP packets. The tool can change the source address of a packet to do IP spoofing and can control IP flags (such as checksums) and TCP flags (such as the state flags, sequence numbers, and ack numbers).

SMAC is a tool used to change the MAC address of a system. It lets a hacker spoof a MAC address when performing an attack.

MAC Changer is a tool used to spoof a MAC address on Unix. It can be used to set the network interface to a specific MAC address, set the MAC randomly, set a MAC of another vendor, set another MAC of the same vendor, set a MAC of the same kind, or display a vendor MAC list to choose from.

WinDNSSpoof is a simple DNS ID spoofing tool for Windows. To use it on a switched network, you must be able to sniff traffic of the computer being attacked. Therefore, it may need to be used in conjunction with an ARP spoofing or flooding tool.

Distributed DNS Flooder sends a large number of queries to create a DoS attack, disabling DNS. If DNS daemon software logs incorrect queries, the impact of this attack is amplified.

Summary

Sniffing is an invaluable tool in the CEH's toolkit. Sniffing can be used to gather information passively and capture valuable data such as passwords. The advantage of sniffing is that it can be performed passively and is virtually undetectable when used in a passive mode. More aggressive methods of sniffing, such as ARP poisoning and DNS spoofing, can be used if passive sniffing does not yield the information the CEH is looking to gather. Just be forewarned that these active methods can be detected and alert security personnel to an attack on the network.

Exam Essentials

Understand how a sniffer works. A sniffer operates in promiscuous mode, meaning it captures all traffic regardless of the destination MAC specified in the frame.

Understand the differences between sniffing in a shared network connected via hubs and a switched network. All traffic is broadcast by a hub, but it's segmented by a switch. To sniff on a switched network, either flooding or ARP spoofing tools must be used.

Know the difference between packets and frames. Packets are created at Layer 3 of the OSI model, and frames are created at Layer 2.

Understand how the Address Resolution Protocol (ARP) works. ARP is used to find a MAC address from a known IP address by broadcasting the request on the network.

Know the difference between active and passive sniffing. Active sniffing is used to trick the switch into acting like a hub so that it forwards traffic to the attacker. Passive sniffing captures packets that are already being broadcast on a shared network.

Review Questions

1. What is sniffing?
 - A. Sending corrupted data on the network to trick a system
 - B. Capturing and deciphering traffic on a network
 - C. Corrupting the ARP cache on a target system
 - D. Performing a password-cracking attack
2. What is a countermeasure to passive sniffing?
 - A. Implementing a switched network
 - B. Implementing a shared network
 - C. ARP spoofing
 - D. Port-based security
3. What type of device connects systems on a shared network?
 - A. Routers
 - B. Gateways
 - C. Hubs
 - D. Switches
4. Which of the following is a countermeasure to ARP spoofing?
 - A. Port-based security
 - B. WinTCPkill
 - C. Wireshark
 - D. MAC-based security
5. What is dsniff?
 - A. A MAC spoofing tool
 - B. An IP address spoofing tool
 - C. A collection of hacking tools
 - D. A sniffer
6. At what layer of the OSI model is data formatted into packets?
 - A. Layer 1
 - B. Layer 2
 - C. Layer 3
 - D. Layer 4

7. What is snort?
 - A. An IDS and packet sniffer
 - B. Only an IDS
 - C. Only a packet sniffer
 - D. Only a frame sniffer
8. What mode must a network card operate in to perform sniffing?
 - A. Shared
 - B. Unencrypted
 - C. Open
 - D. Promiscuous
9. The best defense against any type of sniffing is _____.
 - A. Encryption
 - B. A switched network
 - C. Port-based security
 - D. A good security training program
10. For what type of traffic can WinSniffer capture passwords? (Choose all that apply.)
 - A. POP3
 - B. SMTP
 - C. HTTP
 - D. HTTPS
11. Which of the following software tools can perform sniffing? (Choose all that apply.)
 - A. Dsniff
 - B. Wireshark
 - C. NetBSD
 - D. Netcraft
12. At what layer of the OSI model is data formatted into frames?
 - A. Layer 1
 - B. Layer 2
 - C. Layer 3
 - D. Layer 4
13. In which type of header are MAC addresses located?
 - A. Layer 1
 - B. Layer 2
 - C. Layer 3
 - D. Layer 7

14. In which type of header are IP addresses located?
- A. Layer 1
 - B. Layer 2
 - C. Layer 3
 - D. Layer 7
15. In which header do port numbers appear?
- A. IP
 - B. MAC
 - C. Data Link
 - D. Transport
16. What is the proper Wireshark filter to capture traffic only sent from IP address 131.1.4.7?
- A. `ip.src == 131.1.4.7`
 - B. `ip.address.src == 131.1.4.7`
 - C. `ip.source.address == 131.1.4.7`
 - D. `src.ip == 131.1.4.7`
17. Which Wireshark filter will only capture traffic to `www.google.com`?
- A. `ip.dst = www.google.com`
 - B. `ip.dst eq www.google.com`
 - C. `ip.dst == www.google.com`
 - D. `http.dst == www.google.com`
18. Passwords are found in which layer of the OSI model?
- A. Application
 - B. IP
 - C. Data Link
 - D. Physical
19. Wireshark was previously known as _____.
- A. Packet Sniffer
 - B. Ethereal
 - C. EtherPeek
 - D. SniffIT
20. Cain & Abel can perform which of the following functions? (Choose all that apply.)
- A. Sniffing
 - B. Packet generation
 - C. Password cracking
 - D. ARP poisoning

Answers to Review Questions

1. B. Sniffing is the process of capturing and analyzing data on a network.
2. A. By implementing a switched network, passive sniffing attacks are prevented.
3. C. A network connected via hubs is called a shared network.
4. A. Port-based security implemented on a switch prevents ARP spoofing.
5. C. Dsniff is a group of hacking tools.
6. C. Packets are created and used to carry data at Layer 3.
7. A. Snort is both an intrusion detection system (IDS) and a sniffer.
8. D. A network card must operate in promiscuous mode in order to capture traffic destined for a different MAC address than its own.
9. A. Encryption renders the information captured in a sniffer useless to a hacker.
10. A, B, C. WinSniffer can capture passwords for POP3, SMTP, and HTTP traffic.
11. A, B. Dsniff and Wireshark are sniffer software tools.
12. B. Data is formatted into frames at Layer 2.
13. B. MAC addresses are added in the Layer 2 header.
14. C. IP addresses are added in the Layer 3 header.
15. D. Port numbers are in the Transport layer.
16. A. `ip.src == 131.1.4.7` will capture traffic sent from IP address 131.1.4.7.
17. B. `ip.dst eq www.google.com` is the filter that will capture traffic with the destination `www.google.com`.
18. A. Most passwords such as HTTP, FTP, and telnet passwords are found at the Application layer of the OSI model.
19. B. Wireshark was previously called Ethereal.
20. A, C, D. Cain & Abel can perform sniffing, password cracking, and ARP poisoning.

