# Cyber Security Tools and Technologies

Dr. Arshad Ali
Associate Professor
NUCES Lahore

# Acknowledgment

- These slides are based on Chapter 9 - Sniffers module of Certified Ethical Hacker (CEH) Study Guide v12

# Module Objective

- Comprehending Active and Passive Sniffing
- Overview of Sniffers
- Understanding Sniffers from a cracker perspective
- Sniffing Tools
- ARP Spoofing and Redirection
- DNS and IP Sniffing and Spoofing
- HTTPS Sniffing
- Illustration of various tools used in the above context

# Sniffers – An Introduction

- Sniffers monitor and capture network data.
- Form of tapping phone wires and know about the conversation
- It was an expensive proposition.
  - Because a special network interface required to be able to capture the packets.
  - On top of that was the software
- So, A sniffer can be a self-contained software program or a hardware device with the appropriate software or firmware programming.
  - Not many companies could sell that.

# Sniffers – An Introduction

- Then came consumer network interfaces that could forward all packets up through the interface into the operating system.
- Finally, there was a piece of software called Ethereal. Suddenly, getting your packets captured was free.
- Sniffers usually act as network probes or "snoops" -- examining network traffic but not intercepting or altering it.
- Some sniffers work only with TCP/IP packets,
  - but the more sophisticated tools can work with many other protocols and at lower levels such as the Ethernet frame.

**Snooping** is unauthorized access to another person's or company's data.

# Sniffers – An Introduction

- Possibly if a set of enterprise switch ports is open, then one of their employees can sniff the whole traffic of the network.
- Anyone in the same physical location can plug into the network using Ethernet cable or connect wirelessly to that network and sniff the total traffic.
- Sniffing allows to see all sorts of traffic, both protected and unprotected.
- In the right conditions and with the right protocols in place,
  - an attacking party may be able to gather information that can be used for further attacks or
  - to cause other issues for the network or system owner.

# Security Concern

- ❑ Users of computer networks unwittingly disclose
- ❑ sensitive information about themselves through the use of insecure software, and protocols.

- ❑ Standard implementations of widely adopted protocols such as Windows file sharing (CIFS/SMB), telnet, POP3, HTTP and FTP transmit login passwords in clear text, exposing an extremely large segment of the internet population to sniffing-related attacks.

CIFS: Common Internet File System
SMB: Server Message Block – protocol

# Sniffing Software

- The most common software now to capture packets is freely available.
- Some of the software is command-line oriented, which is really helpful
  - if you are connected to a system over SSH or other mechanisms that provide only a text-oriented interface.
- This sort of software is not great for analysis
  - if you need to dig into the packet content and see entire streams,
  - if you want to go deeper with your analysis and get statistics and graphs.
- Fortunately, once called Ethereal, currently known as Wireshark), has a lot of capabilities to enable deep analysis of the packets.

# Sniffing Challenges

- Most networks are switched which are essentially segmented).
- Only packets addressed to you in some way will get to your network interface.
- Since we may have the ability to see the data anyway, through the applications transmitting the messages
  - just look at messages to and from the device you are on.
- We need ways to get the rest of the traffic on the network to a system under our control.
- Fortunately, we can even take a targeted approach at it,
  - determining exactly which systems we want to get traffic from.

# Sniffing Challenges

- We also need to deal with encryption.

- Web traffic today generally defaults to using Transport Layer Security (TLS),
  - which encrypts all messages.
- Challenge: one of the predominant protocols becomes obscured to us.
- Fortunately, there are some ways around that.

# Packet Capture

- If you can get to the right place in the network to capture the data,
  - you can potentially grab usernames and passwords or other authentication/authorization traffic.
- An attacker could potentially grab credit card information or other personally identifiable information (PII) that is marketable.
- Depending on the organization, there could also be personal health information (PHI).
- There may be other useful information available on the network as you maneuver through your client's assets.

# Packet Capture

- Packet capturing is the process of acquiring network traffic addressed to systems other than your own.
- You can certainly capture packets that are only addressed to your system,
  - Its not interesting as you're already getting them.
- NICs are programmed to only forward those frames to the operating system
  - whose destination MAC address is either the MAC address of the NIC or
  - the broadcast MAC address (ff:ff:ff:ff:ff:ff).
- To force the NIC to forward all messages up to OS,
  - the card has to be put into promiscuous mode.
  - This mode just gets the NIC to forward all messages up,

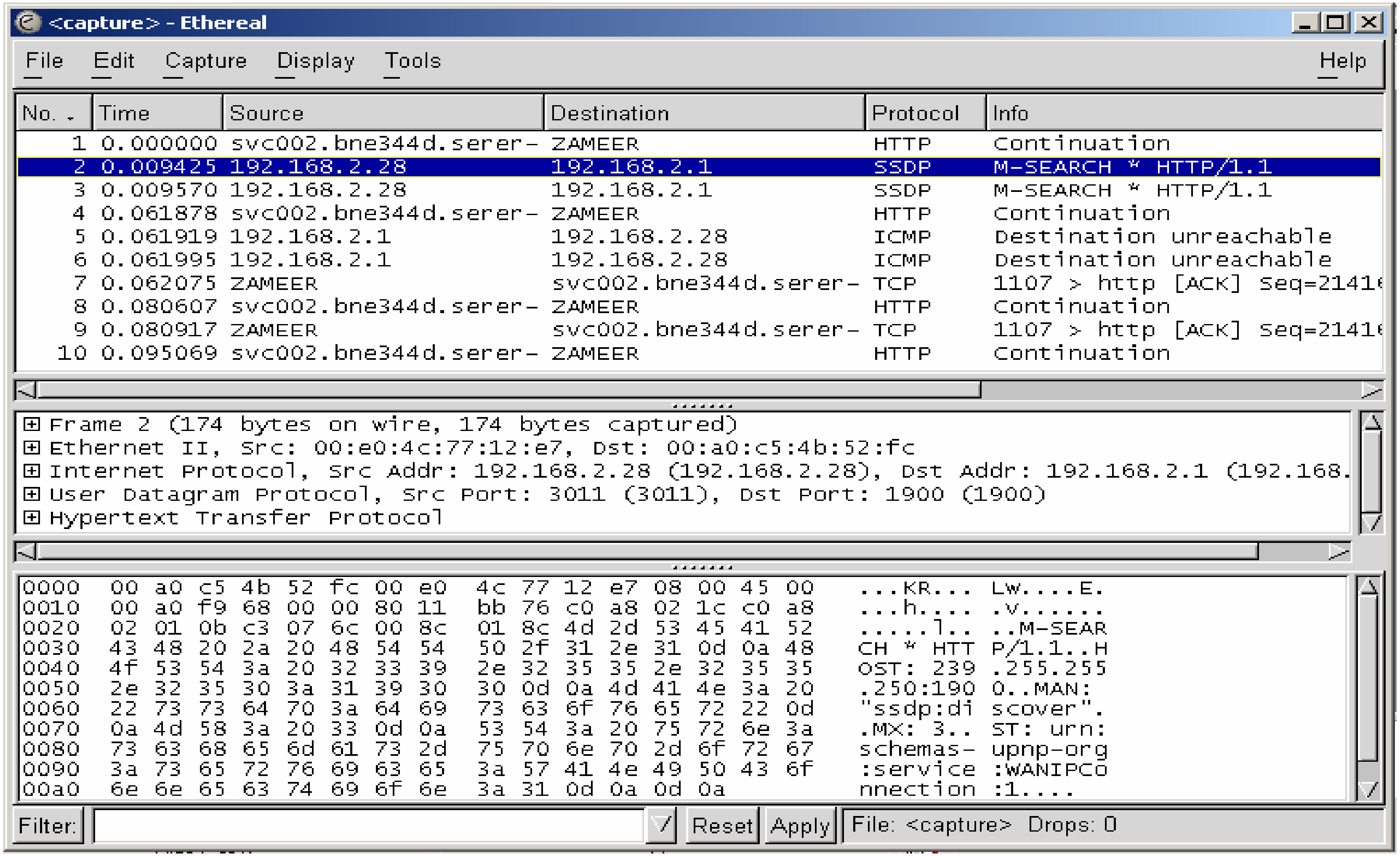
# Packet Capture

- Once the networking stack in the OS, has the frames, they can be intercepted by a piece of software.
- packet capturing software will parse the message, extracting information out of each protocol header.
- A command line program, like tcpdump may display information out of different headers.
- Tools with graphical user interface (GUI) like Wireshark have more options regarding
  - how to display the header data, as well as the payload data.

# Packet Capture

- Headers provide instructions to the protocol on how to behave.
- The data being carried from one endpoint to another is called the payload.
  - Payload may be broken up between multiple packets and certainly multiple frames.
  - Fragmentation may be forced by MTU at link layer.
- This is one reason to use a GUI based program to analyze packets, even if we are capturing them with another tool.
- A program like Wireshark helps to make packet analysis much easier than if we were trying to review all the details on the command line.

# Tool: Ethereal



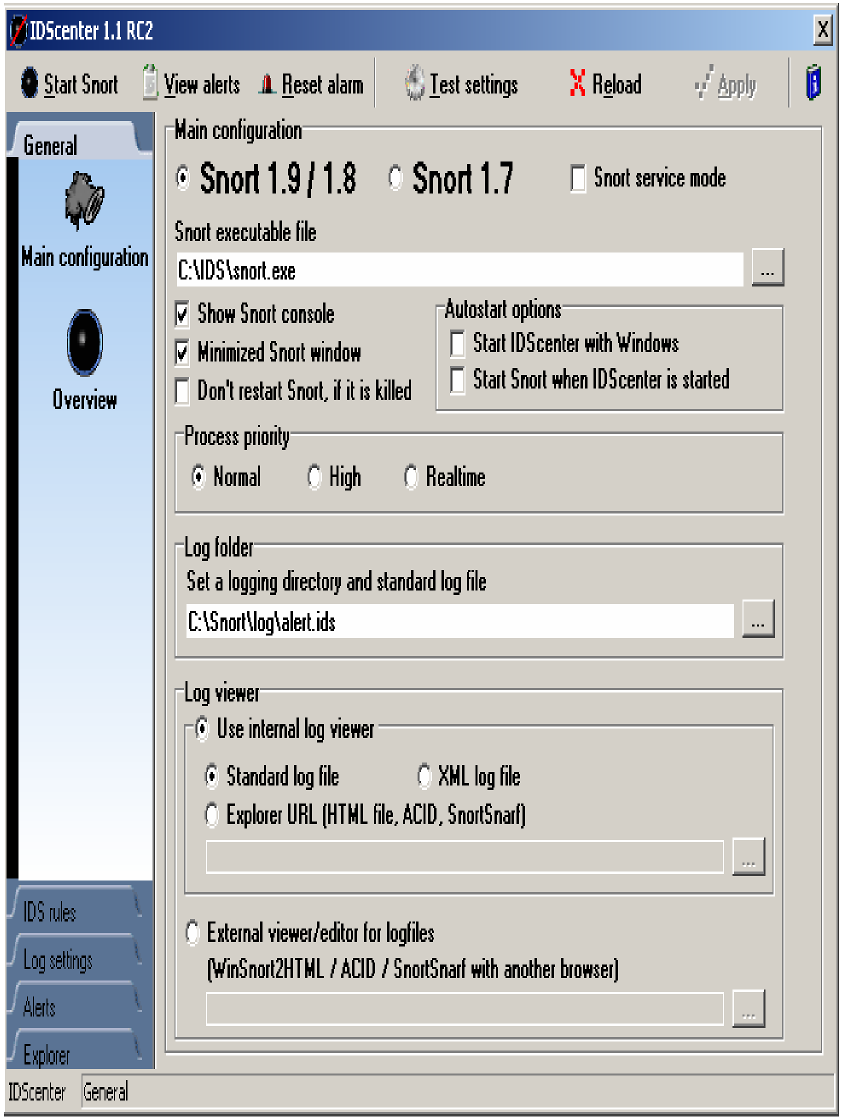
```
No. .  Time      Source                  Destination             Protocol  Info
    1 0.000000 svc002.bne344d.serer- ZAMEER                  HTTP      Continuation
    2 0.009425 192.168.2.28          192.168.2.1             SSDP      M-SEARCH * HTTP/1.1
    3 0.009570 192.168.2.28          192.168.2.1             SSDP      M-SEARCH * HTTP/1.1
    4 0.061878 svc002.bne344d.serer- ZAMEER                  HTTP      Continuation
    5 0.061919 192.168.2.1           192.168.2.28            ICMP      Destination unreachable
    6 0.061995 192.168.2.1           192.168.2.28            ICMP      Destination unreachable
    7 0.062075 ZAMEER                svc002.bne344d.serer-   TCP       1107 > http [ACK] Seq=2141
    8 0.080607 svc002.bne344d.serer- ZAMEER                  HTTP      Continuation
    9 0.080917 ZAMEER                svc002.bne344d.serer-   TCP       1107 > http [ACK] Seq=2141
   10 0.095069 svc002.bne344d.serer- ZAMEER                  HTTP      Continuation
```

```
⊞ Frame 2 (174 bytes on wire, 174 bytes captured)
⊞ Ethernet II, Src: 00:e0:4c:77:12:e7, Dst: 00:a0:c5:4b:52:fc
⊞ Internet Protocol, Src Addr: 192.168.2.28 (192.168.2.28), Dst Addr: 192.168.2.1 (192.168.
⊞ User Datagram Protocol, Src Port: 3011 (3011), Dst Port: 1900 (1900)
⊞ Hypertext Transfer Protocol
```

```
0000  00 a0 c5 4b 52 fc 00 e0  4c 77 12 e7 08 00 45 00   ...KR... Lw....E.
0010  00 a0 f9 68 00 00 80 11  bb 76 c0 a8 02 1c c0 a8   ...h.... .v......
0020  02 01 0b c3 07 6c 00 8c  01 8c 4d 2d 53 45 41 52   .....l.. ..M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1..H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0..MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover".
0070  0a 4d 58 3a 20 33 0d 0a  53 54 3a 20 75 72 6e 3a   .MX: 3.. ST: urn:
0080  73 63 68 65 6d 61 73 2d  75 70 6e 70 2d 6f 72 67   schemas- upnp-org
0090  3a 73 65 72 76 69 63 65  3a 57 41 4e 49 50 43 6f   :service :WANIPCo
00a0  6e 6e 65 63 74 69 6f 6e  3a 31 0d 0a 0d 0a         nnection :1....
```

SSDP: Simple Service Discovery Protocol

# Tool: Snort



Main configuration

● Snort 1.9 / 1.8 ○ Snort 1.7 □ Snort service mode

Snort executable file

C:\IDS\snort.exe

☑ Show Snort console

☑ Minimized Snort window

□ Don't restart Snort, if it is killed

Autostart options

□ Start IDScenter with Windows

□ Start Snort when IDScenter is started

Process priority

● Normal ○ High ○ Realtime

Log folder

Set a logging directory and standard log file

C:\Snort\log\alert.ids

Log viewer

● Use internal log viewer

● Standard log file ○ XML log file

○ Explorer URL (HTML file, ACID, SnortSnarf)

○ External viewer/editor for logfiles

(WinSnort2HTML / ACID / SnortSnarf with another browser)

- There are three main modes in which Snort can be configured: sniffer, packet logger, and network intrusion detection system.
- Sniffer mode simply reads the packets of the network and displays them for you in a continuous stream on the console.
- Packet logger mode logs the packets to the disk.
- Network intrusion detection mode is the most complex and configurable configuration, allowing Snort to analyze network traffic for matches against a user defined rule set

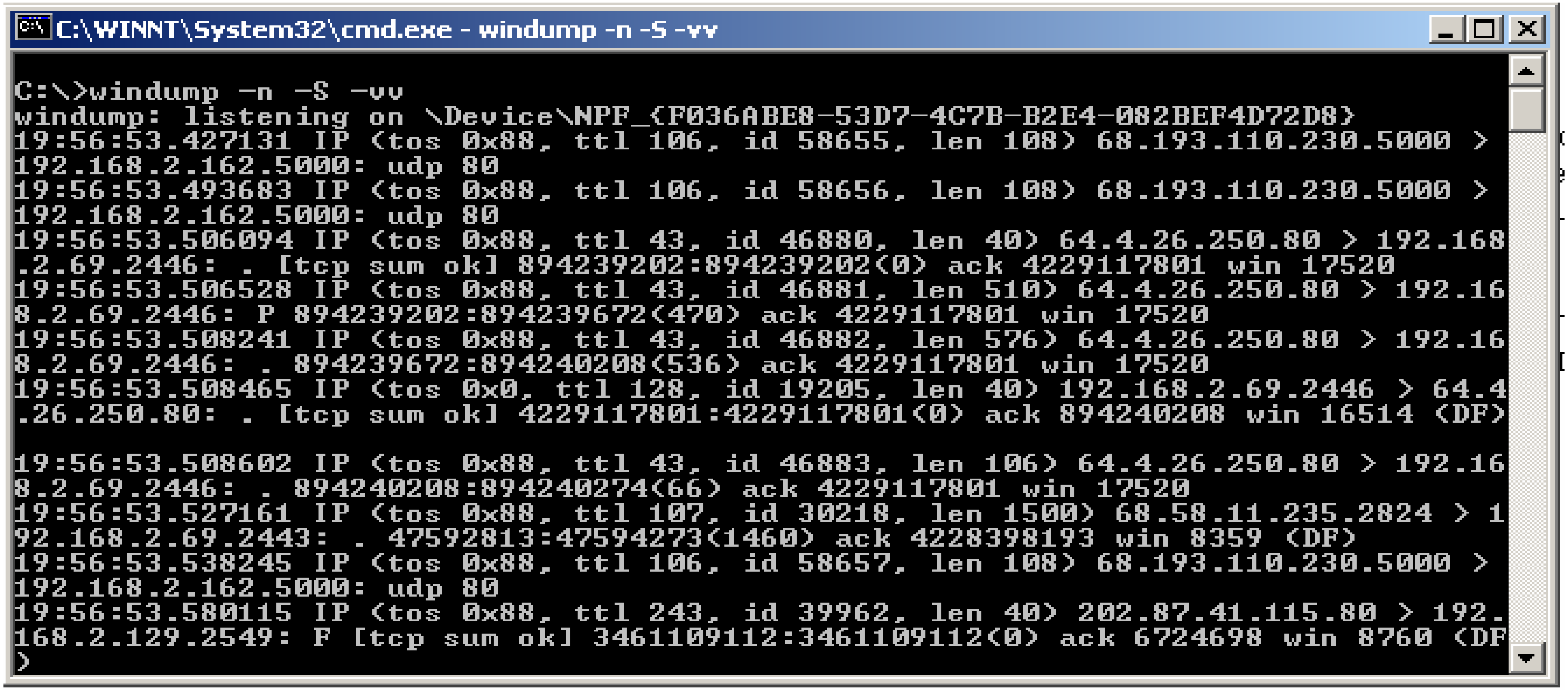
# Tool: tcpdump

- First written in the late 1980s and was ported to Unix implementations.
- Standardized in the late 1990s, pulling all the divergent implementations together.
- Available on most Linux distributions as well as different Berkeley Software Distribution (BSD) distributions.
- A command-line program used to get an idea of what is happening on the network,
  - Can be used to capture traffic and store that traffic in a file to be opened later on.

Explore various tcpdump commands

# Tool: Windump

- WinDump is the porting to the Windows platform of tcpdump, the most used network sniffer/analyzer for UNIX.
- Command line tool perfect for displaying header information



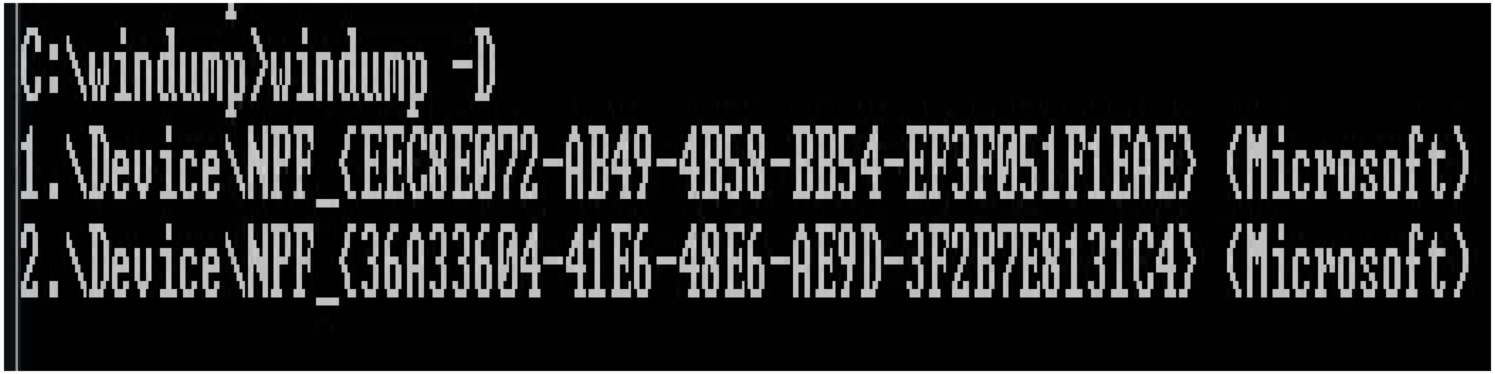
```
C:\>windump -n -S -vv
windump: listening on \Device\NPF_{F036ABE8-53D7-4C7B-B2E4-082BEF4D72D8}
19:56:53.427131 IP (tos 0x88, ttl 106, id 58655, len 108) 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.493683 IP (tos 0x88, ttl 106, id 58656, len 108) 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.506094 IP (tos 0x88, ttl 43, id 46880, len 40) 64.4.26.250.80 > 192.168
.2.69.2446: . [tcp sum ok] 894239202:894239202(0) ack 4229117801 win 17520
19:56:53.506528 IP (tos 0x88, ttl 43, id 46881, len 510) 64.4.26.250.80 > 192.16
8.2.69.2446: P 894239202:894239672(470) ack 4229117801 win 17520
19:56:53.508241 IP (tos 0x88, ttl 43, id 46882, len 576) 64.4.26.250.80 > 192.16
8.2.69.2446: . 894239672:894240208(536) ack 4229117801 win 17520
19:56:53.508465 IP (tos 0x0, ttl 128, id 19205, len 40) 192.168.2.69.2446 > 64.4
.26.250.80: . [tcp sum ok] 4229117801:4229117801(0) ack 894240208 win 16514 (DF)

19:56:53.508602 IP (tos 0x88, ttl 43, id 46883, len 106) 64.4.26.250.80 > 192.16
8.2.69.2446: . 894240208:894240274(66) ack 4229117801 win 17520
19:56:53.527161 IP (tos 0x88, ttl 107, id 30218, len 1500) 68.58.11.235.2824 > 1
92.168.2.69.2443: . 47592813:47594273(1460) ack 4228398193 win 8359 (DF)
19:56:53.538245 IP (tos 0x88, ttl 106, id 58657, len 108) 68.193.110.230.5000 >
192.168.2.162.5000: udp 80
19:56:53.580115 IP (tos 0x88, ttl 243, id 39962, len 40) 202.87.41.115.80 > 192.
168.2.129.2549: F [tcp sum ok] 3461109112:3461109112(0) ack 6724698 win 8760 (DF
)
```

# Tool: Windump

- In order to run 'windump' you need to have library called 'WinPcap'.

- Execute following command to print the available network interfaces on the system and on which tcpdump can capture packets.

```
windump -D
```



```
C:\windump>windump -D
1.\Device\NPF_{EEC8E072-AB49-4B58-BB54-EF3F051F1EAE} (Microsoft)
2.\Device\NPF_{36A33604-41E6-48E6-AE9D-3F2B7E8131C4} (Microsoft)
```

# Tool: Windump

- Export a tcpdump to a file while following the scenario where we can observe an excpetion. To do that, execute the following command first,

```
windump -i {network_Interface_id} -w {filename}.pcap
```

```
windump -i 1 -w tcpdump.pcap
```

- **network_Interface_id:** the number of the network interface
  - if it listed only 1 interface, the number would be ‘1’.
  - If multiple interfaces were returned, select a number from the list of interfaces and use here
- ***filename*** —give any name to the file and save with .pcap (packet capturing extension)

# Tool: Windump

- Run a scenario which you need to capture tcpdump. The captured packets will save in file (tcpdump.pcap)

Show only the first 2 packets

```
windump -n -r flename.pcap -c 2
```

Tracking host by source MAC address

```
windump -n -r filename.pcap -e "ether src 00:a0:cc:3b:bf:fa"
```

Tracking host by IP, whether that IP is source or destination

```
windump -n -r filename.pcap "host 192.168.0.1"
```

# Hands on Activity

- Capture packets using tcpdump or windump on your local network.
- Be sure to save the file.
- Also be sure to engage in some network activity, like browsing the web, to have something to capture.

# tshark

- Wireshark includes the program tshark, which can also be used to capture packets.
- Program running without any parameters for capturing traffic.

```
root@quiche:~# tshark
```

- tshark helpfully tells us that running the program as root could be dangerous.

# Wireshark

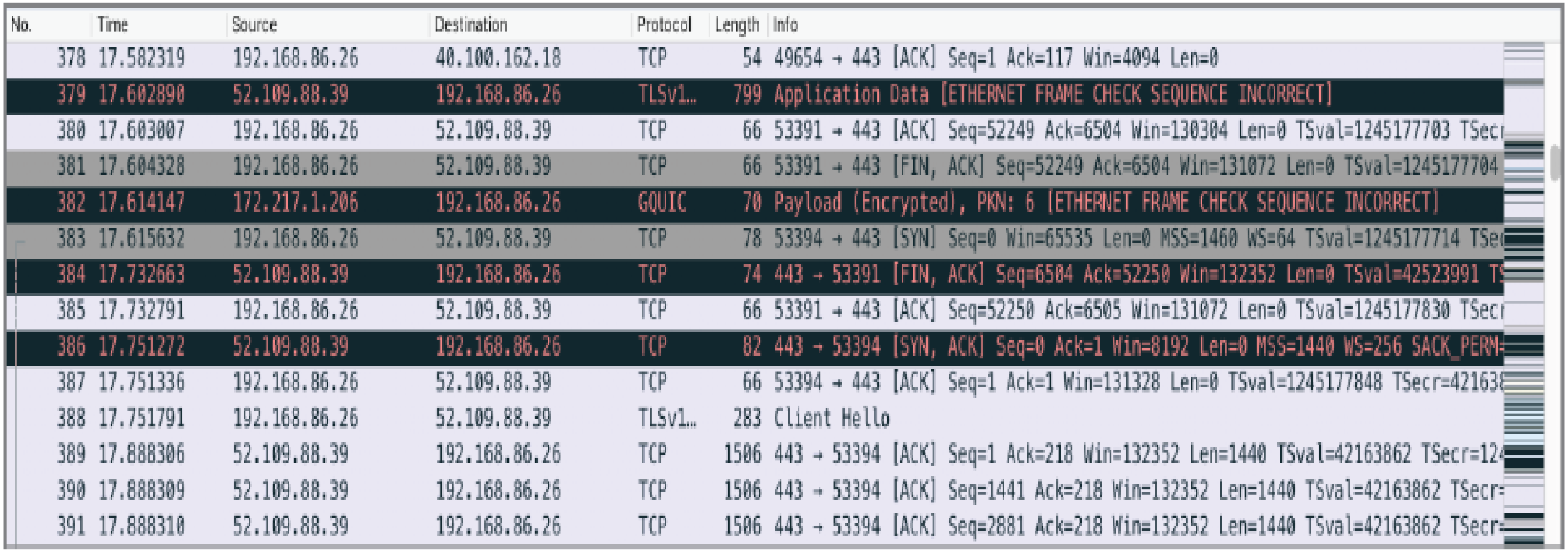
- A GUI-based packet capture program which comes with some command-line programs.

**Advantages:**

- gives us a way to view the packets easily, moving around the complete capture.
- Unlike with tcpdump and tshark, we see the entire network stack in Wireshark,
  - which technically makes what we have captured frames rather than packets
- Gives us the ability to easily scroll through the list of all frames captured as well.
- we get what is essentially a summary of each frame like tcpdump

# Wireshark

- The configurable columns are  frame number, relative time from the start of the capture, addresses, protocol, frame length, and then an info column.
- Gives summary of each frame.
- provides full protocol decodes



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 378 | 17.582319 | 192.168.86.26 | 40.100.162.18 | TCP | 54 | 49654 → 443 [ACK] Seq=1 Ack=117 Win=4094 Len=0 |
| 379 | 17.602890 | 52.109.88.39 | 192.168.86.26 | TLSv1… | 799 | Application Data [ETHERNET FRAME CHECK SEQUENCE INCORRECT] |
| 380 | 17.603007 | 192.168.86.26 | 52.109.88.39 | TCP | 66 | 53391 → 443 [ACK] Seq=52249 Ack=6504 Win=130304 Len=0 TSval=1245177703 TSecr |
| 381 | 17.604328 | 192.168.86.26 | 52.109.88.39 | TCP | 66 | 53391 → 443 [FIN, ACK] Seq=52249 Ack=6504 Win=131072 Len=0 TSval=1245177704 |
| 382 | 17.614147 | 172.217.1.206 | 192.168.86.26 | GQUIC | 70 | Payload (Encrypted), PKN: 6 [ETHERNET FRAME CHECK SEQUENCE INCORRECT] |
| 383 | 17.615632 | 192.168.86.26 | 52.109.88.39 | TCP | 78 | 53394 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1245177714 TSec |
| 384 | 17.732663 | 52.109.88.39 | 192.168.86.26 | TCP | 74 | 443 → 53391 [FIN, ACK] Seq=6504 Ack=52250 Win=132352 Len=0 TSval=42523991 TS |
| 385 | 17.732791 | 192.168.86.26 | 52.109.88.39 | TCP | 66 | 53391 → 443 [ACK] Seq=52250 Ack=6505 Win=131072 Len=0 TSval=1245177830 TSecr |
| 386 | 17.751272 | 52.109.88.39 | 192.168.86.26 | TCP | 82 | 443 → 53394 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM= |
| 387 | 17.751336 | 192.168.86.26 | 52.109.88.39 | TCP | 66 | 53394 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=1245177848 TSecr=42163 |
| 388 | 17.751791 | 192.168.86.26 | 52.109.88.39 | TLSv1… | 283 | Client Hello |
| 389 | 17.888306 | 52.109.88.39 | 192.168.86.26 | TCP | 1506 | 443 → 53394 [ACK] Seq=1 Ack=218 Win=132352 Len=1440 TSval=42163862 TSecr=124 |
| 390 | 17.888309 | 52.109.88.39 | 192.168.86.26 | TCP | 1506 | 443 → 53394 [ACK] Seq=1441 Ack=218 Win=132352 Len=1440 TSval=42163862 TSecr= |
| 391 | 17.888310 | 52.109.88.39 | 192.168.86.26 | TCP | 1506 | 443 → 53394 [ACK] Seq=2881 Ack=218 Win=132352 Len=1440 TSval=42163862 TSecr= |

# Hands on Activity

- Open the file in Wireshark you saved using using tcpdump
- Try to filter the web traffic so that all you see is the web browsing you were doing.

# Port Mirroring/Spanning

- Simple hubs were just electrical repeaters with no intelligence, getting traffic from everywhere on the network was easy.
- Switches improve the performance and security of a network by doing filtering at layer 2 of the network device.
- A switch knows which systems are connected to it at which port.

# Port Mirroring/Spanning

- When a frame comes in with a destination MAC address
  - the switch can look up the port where the MAC address is, and
  - send the frame out that port to the destination system.
- Thus, other systems on the network never see that frame pass their network interface.
- This makes capturing more difficult if you are looking for traffic that isn't passing your network interface.

# Port Mirroring/Spanning

- To capture other traffic, we need to have access to the switch.
- This would allow you to configure the switch to mirror ports.
  - i.e., any traffic that passes through one port would be mirrored to another port.
- Multiple ports may be mirrored to a single port,
- this lets monitoring traffic to and from multiple systems.

# Port Mirroring/Spanning

- **Switched Port Analyzer** (SPAN): feature used for configuring port mirroring On Cisco devices
- **Port spanning process**: when you set up port mirroring, you configure a SPAN port
- Other switch vendors may use other terminology for this process.

# Port Mirroring/Spanning

**Important: Idea of Oversubscription**.

- If you have five 1-Gbps switch ports and you are mirroring them out to a single 1-Gbps port,
  - You are possibly oversubscribing the receiving port.
- So, you could easily drop packets (randomly) you were trying to capture
  - depending on if too much data is coming in to be able to send out.

# Working of Sniffing

- A sniffer normally turns the NIC of the system to the **promiscuous mode**
  - This mode allows a network device (NIC) to intercept and read each network packet transmitted on its segment (even not addressed to it)
- A sniffer can continuously monitor all the traffic to a computer through the NIC by decoding the information encapsulated in the data packets.

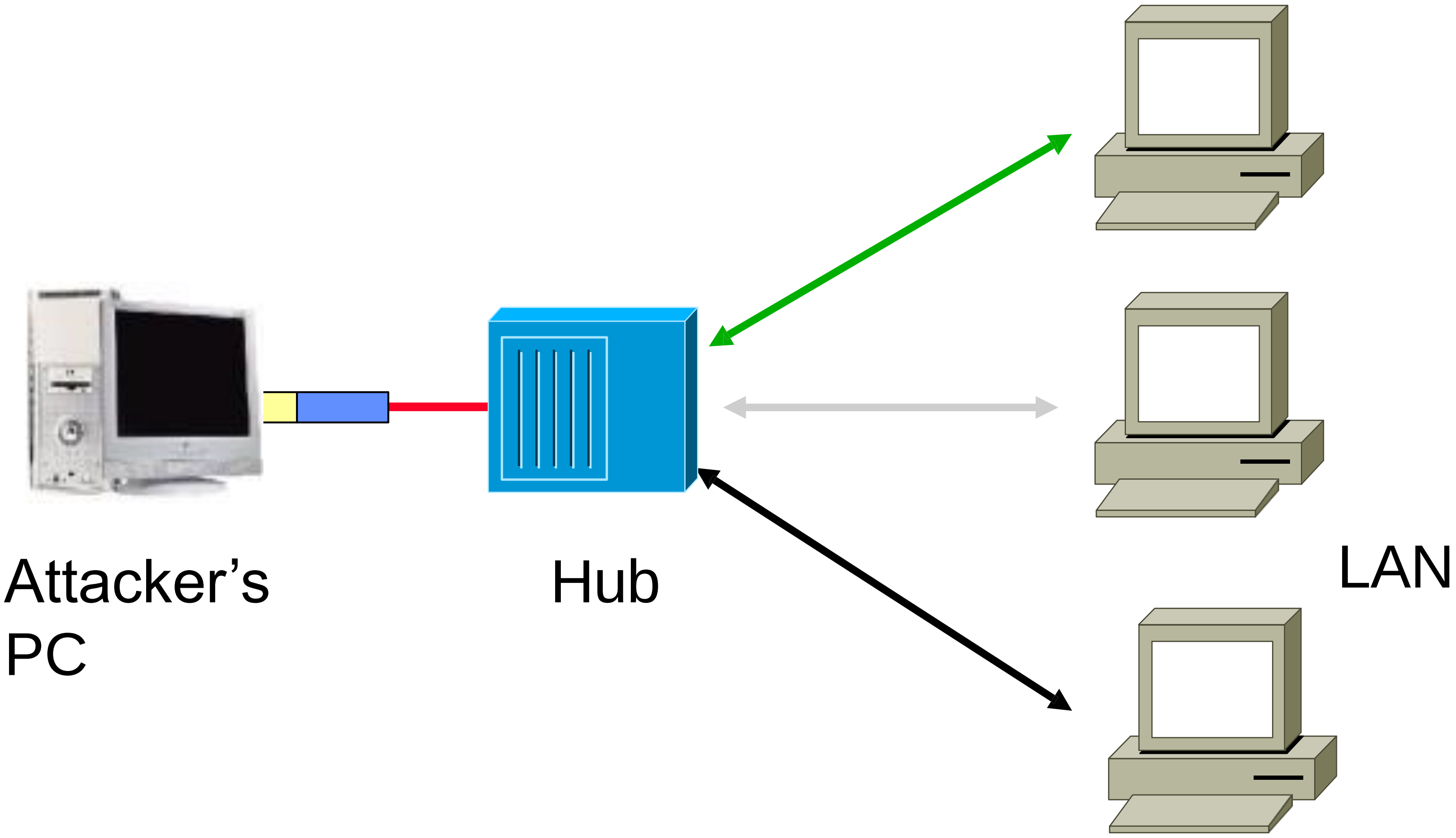
- **Types of Sniffing**
- Passive Sniffing
- Active Sniffing

# Passive Sniffing

- ❑ When several devices are connected to your LAN or wireless network,
  - ❑ a hacker could connect too and passively monitor traffic going through the hub.
- ❑ The traffic is locked but it is not altered in any way
- ❑ It allows listening only

- ❑ This type of packet sniffing can be very difficult to detect
  - ❑ Fortunately, hubs are almost obsolete nowadays.
  - ❑ Most modern networks use switches.
  - ❑ So, passive sniffing is no more effective.

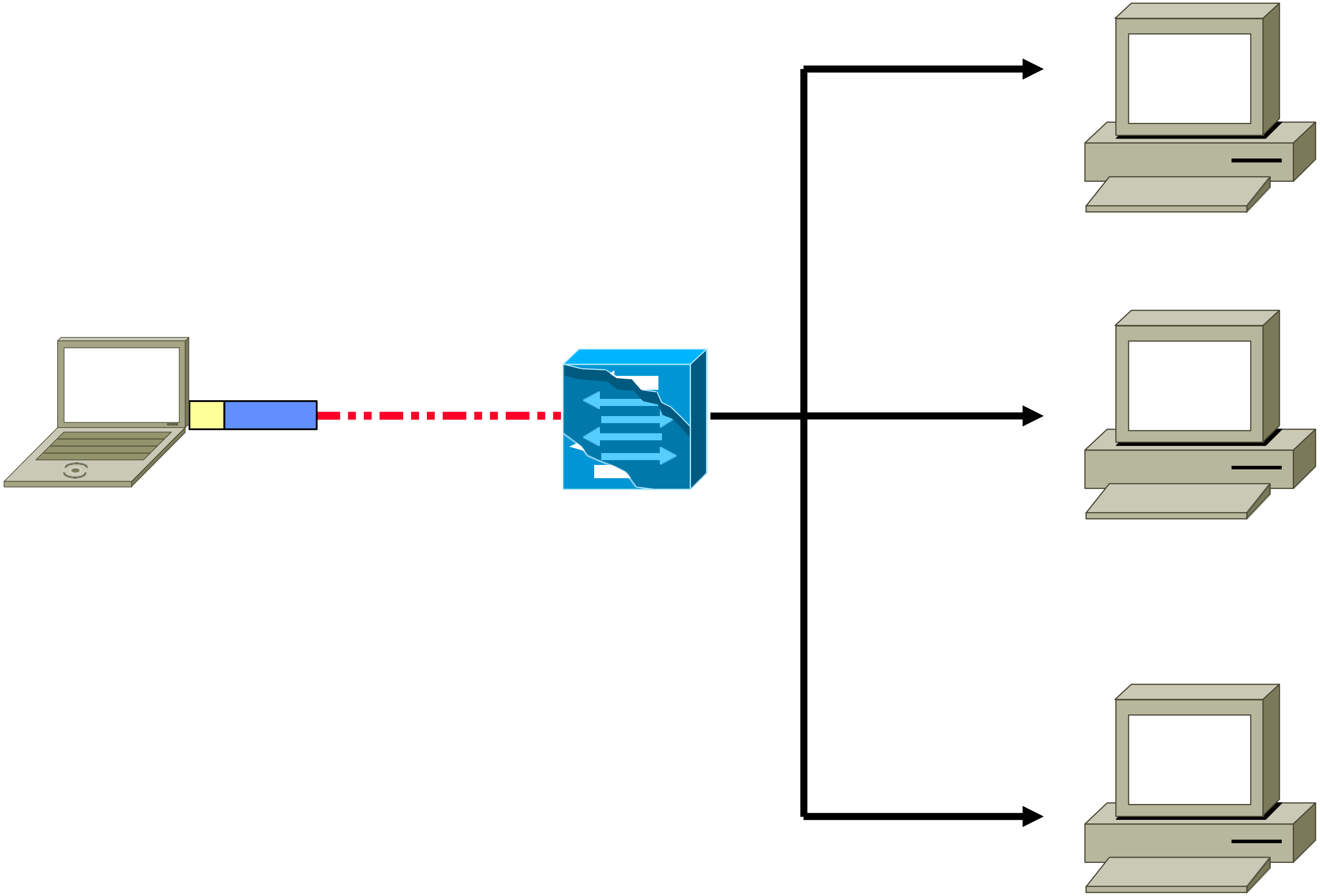
# Passive Sniffing



# Active Sniffing

- The switch regulates the flow of data between its ports by actively monitoring the MAC address on each port,
  - which helps it pass data only to its intended target.
- Sniffers need to actively inject traffic into the LAN to enable sniffing of the traffic.
  - The traffic is not only locked and monitored, but it may also be altered in some way as determined by the attack.

# Active Sniffing



# Detecting Sniffers

- While the sniffer is running on a single system and is collecting information (may be thought as a passive activity)
- Ways of detecting when an interface is in promiscuous mode.
  - Easiest: look on the system having the interface in promiscuous mode.
  - For example: On a mac OS with tcpdump running, **ifconfig** produces the following output.

```
en0:
flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST
> mtu 1500

options=50b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV,CHANNEL_IO>
                ether 14:98:77:31:b2:33
```

# Detecting Sniffers

- ipconfig on Windows just provides the IP configuration, rather than the interface configuration,
  - so not possible to see if the interface is in promiscuous mode.
- To identify whether a device is in promiscuous mode,
  - You don't have to be on the machine
  - though you do have to have access to the local network to be able to look at packets yourself in some way.
- A network/system administrator responsible for the DNS server may be able to identify a device that is sniffing packets.

# Detecting Sniffers

- Tools like Wireshark and tcpdump will do hostname lookups by default,
  - so for every new IP address that comes across, there will be a corresponding DNS lookup by the operating system where the packet capture is happening.
- When a device is in promiscuous mode,
  - all traffic arriving at the interface is forwarded to the operating system.
  - This may allow unusual or unexpected behavior.

# Detecting Sniffers

- For example, if you were to craft a packet using packETH or another tool,
  - you could send an ICMP echo request message to a device you suspected of capturing packets.
  - You create the packet by setting the correct IP address but an incorrect MAC address.
  - If the device's interface isn't in promiscuous mode, it won't receive the message.
  - If it gets the message, it will notice the IP address is correct and respond.
- However, in a switched environment,
- the message simply won't get to the device, which would rely on an attack technique like ARP spoofing to get the message to the device.
- Resultantly, this technique may not be reliable.

# Detecting Sniffers

- Finally, you can observe spoofing attacks like ARP spoofing.
- If you see a large number of ARP messages on the network, especially ARP replies without corresponding requests,
    - it's certain there is an ARP spoofing attack happening.
    - This may indicate someone is capturing packets on the network.

# Packet Analysis

- See Packet Analysis section of Chapter 9 of CEH V12 (page 26 to page 32)
- It is based on Wireshark

# ARP Spoofing

- ARP has two stages.
- First: request, where a system knows an IP address but doesn't know the corresponding MAC address.
- It sends an ARP request asking for the system with the IP address to respond with its MAC address.
- The response is the system replying, indicating its MAC address to the requestor.
- There is nothing to authenticate that request, though.
- In theory, anyone could respond to that request with their MAC address to get the requesting system to send the message to the attacker's/spoofer's address. We could make it even easier by simply not waiting for the request to begin with and just sending the reply.

# ARP Attacks

Two types of ARP attacks exist.

- **ARP spoofing:** Attacker sends fake ARP packets that link its MAC address with an IP of a computer already on the LAN.
- ARP packets can be forged to send data to the attacker’s machine.
- **ARP poisoning:** After a successful ARP spoofing, attacker changes the company's ARP table, so it contains falsified MAC maps. The contagion (poison) spreads.

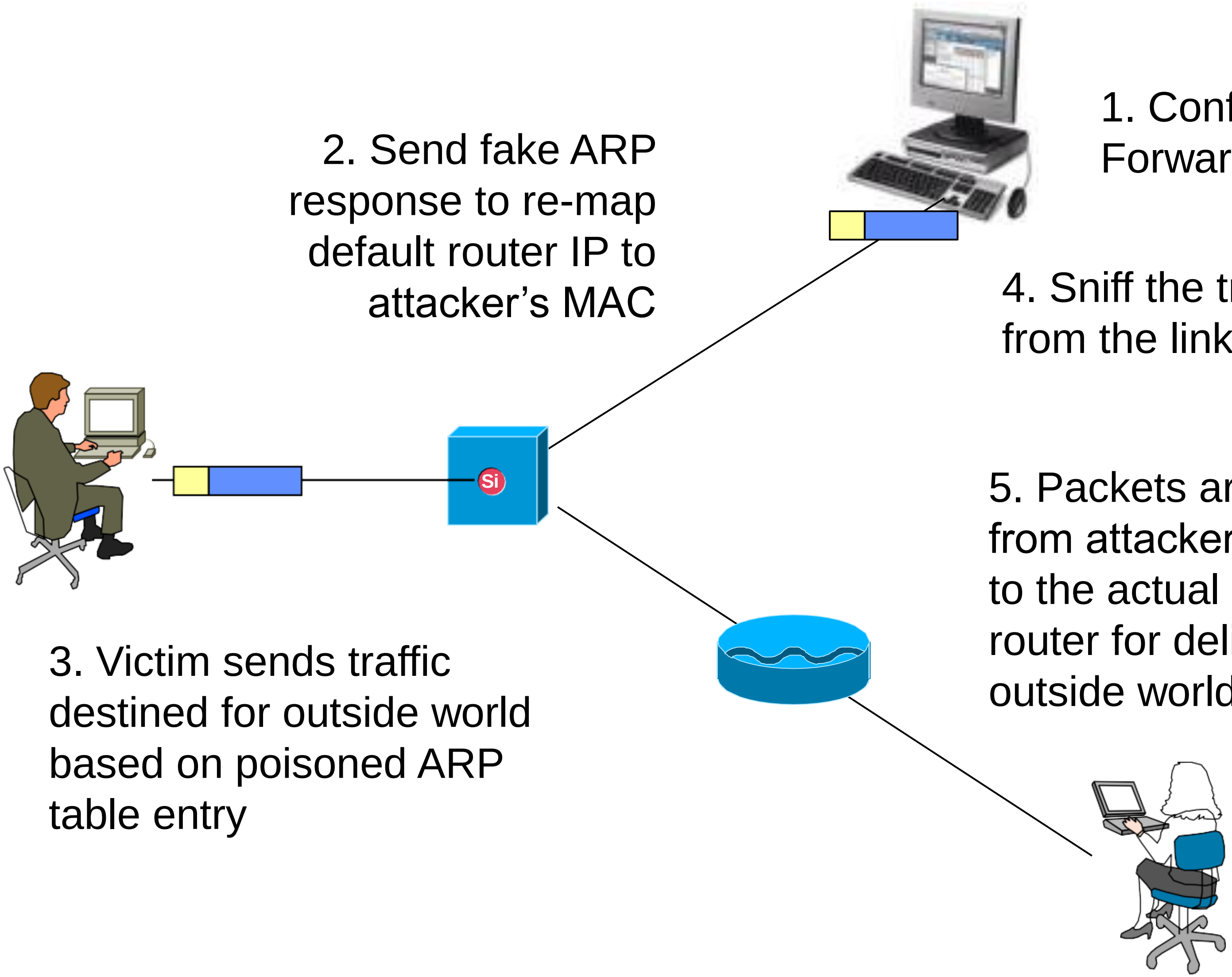
- The goal is to link a hacker's MAC with the LAN.
- The result means any traffic sent to the compromised LAN will head to the attacker instead.

# ARP Attacks

A successful ARP attacked/hacker can:

- **Hijack.** Someone may look over everything that heads to the LAN before releasing it.
- **Deny service.** Someone may refuse to release anything from the infected LAN unless some kind of ransom is paid.
- **Sit in the middle.** Someone conducting a man-in-the-middle attack can do almost anything, including altering documents before sending them out.
- These attacks both threaten confidentiality and reduce user confidence.
- They are among the most dangerous attacks anyone can perpetrate.

# ARP Spoofing



1. Configure IP Forwarding

2. Send fake ARP response to re-map default router IP to attacker’s MAC

3. Victim sends traffic destined for outside world based on poisoned ARP table entry

4. Sniff the traffic from the link

5. Packets are forwarded from attacker’s machine to the actual default router for delivery to the outside world

# Activity

Tools required

- ❑ VMware workstation
- ❑ Kali Linux or Linux Operating system
- ❑ Ettercap Tool
- ❑ LAN connection

- ❑ Perform all the steps provided at the following link:

https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_arp_poisoning.htm

# DNS Sniffing and Spoofing

- DNS Spoofing is said to have occurred when a DNS entry points to another IP instead of the legitimate IP address.
- When an attacker wants to poison a DNS cache, he will use a faulty DNS – which can be his own domain running a hacked DNS server. The DNS server is termed as hacked because the IP address records are manipulated to suit the attacker's needs.

# WinDNSSpoof

- This tool is a simple DNS ID Spoofer for Windows 9x/2K.
- In order to use it you must be able to sniff traffic of the computer being attacked.
- Usage : wds -h
  - Example : wds -n www.microsoft.com -i 216.239.39.101
  - -g 00-00-39-5c-45-3b

# DHCP Starvation Attack

- DHCP starvation is used to acquire information from end users
- works on IPv4 addresses that are dynamically provided to endpoints
- can end up having two outcomes,
  - both malicious

In this attack,

- attacker sends (broadcasts) many DHCPDISCOVER messages to the DHCP server
- DHCP server responds with an offer to the client.
- Here, the server will keep sending offers,
  - reserving the IP address for that client, expecting an acknowledgment that never comes.

# DHCP Starvation Attack

- At some point, the DHCP server runs out of IP addresses to provide to any legitimate endpoint.
- May result in a denial-of-service attack
- No endpoints would be able to get IP addresses from the legitimate DHCP server.

At this point

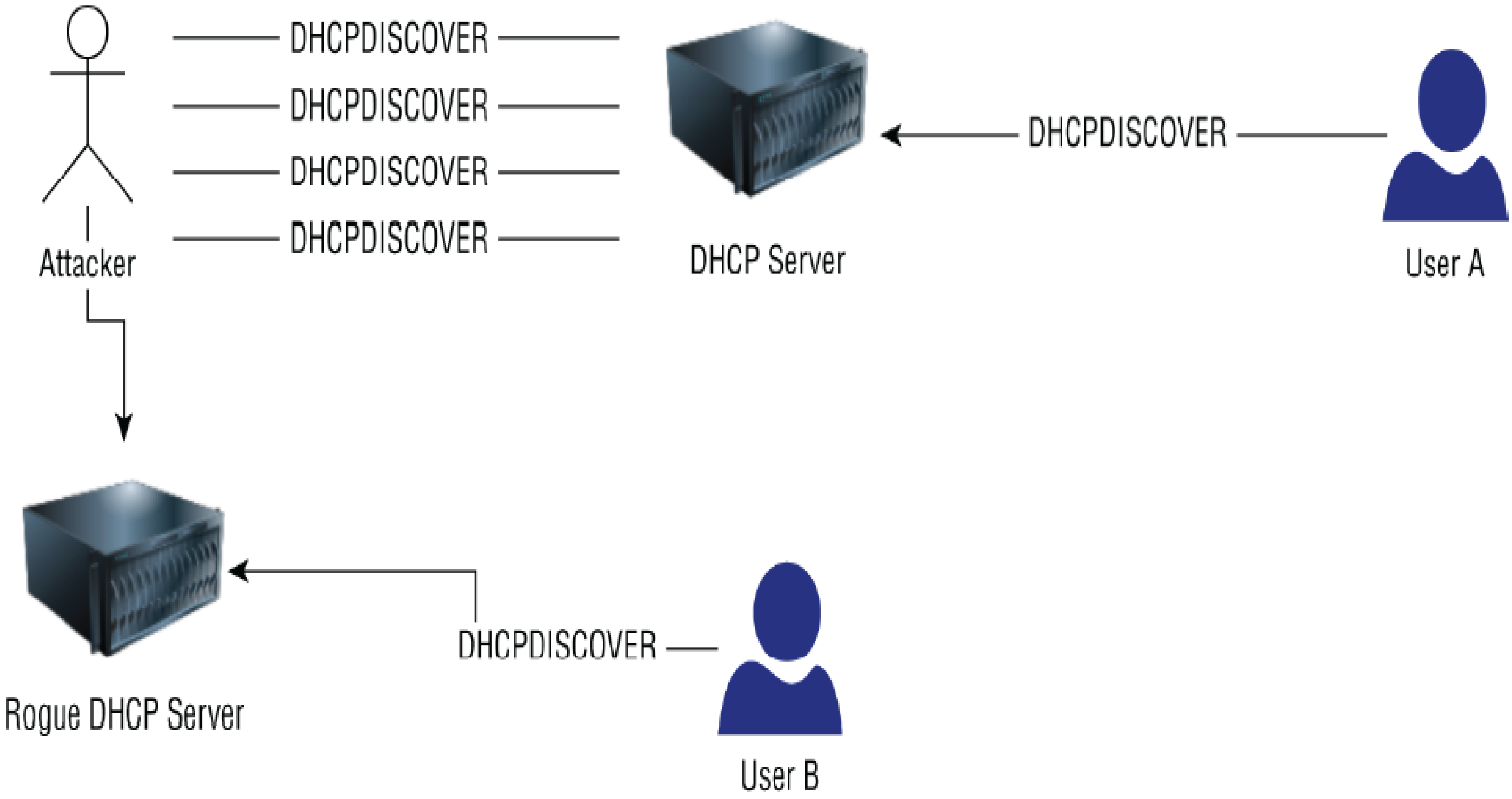
- the attacker can create a new DHCP server
- As DHCP works with broadcast addresses for new requests,
  - so there is no need to pretend to be any specific IP address

# DHCP Starvation Attack

In addition to IP addresses,

- a DHCP server can provide other IP configuration details,
  - which may include a bogus DNS server,
  - which could provide malicious addresses to requests to ensure users go to websites controlled by the attacker.
- the attacker could provide a default gateway of a system on the local network they controlled.
- This would guarantee all traffic would be sent to this system so it could be observed before being sent on to the legitimate gateway.
- This requires some extra work to turn the device being controlled into what is effectively a router,
  - but it is possible and would work much like the ARP

# DHCP Starvation Attack



# Spoofing Detection

- ❑ See Spoofing Detection section of Chapter 9 of CEH V12 (page 45 to page 47)