



# Unit-5 Network Forensics

DR. VIJETA KHARE



# Footprints:

refer to the impact that an investigator has on the systems under examination

- ▶ When conducting network forensics, investigators often work with live systems that cannot be taken offline.
- ▶ These may include routers, switches, and other types of network devices, as well as critical servers.
- ▶ network-based evidence is often highly volatile and must be collected through active means that inherently modify the system hosting the evidence.
- ▶ Even when investigators are able to sniff traffic using port monitoring or tapping a cable, there is always some impact on the environment, however small.
- ▶ This Every interaction that an investigator has with a live system modifies it in some way,

# Concepts in Digital Evidence

evidence (noun)

1. information or signs indicating whether a belief or proposition is true or valid.
2. information used to establish facts in a legal investigation or admissible as testimony in a law court.

- Real
- Best
- Direct
- Circumstantial
- Hearsay
- Business Records
- Digital
- Network-Based Digital

# Real Evidence

- ▶ Real evidence usually comprises the physicality of the event, and as such is often the most easily presented and understood element of a crime.

Examples of “real evidence” can include:

- ▶ • The murder weapon
- ▶ • The fingerprint or footprint
- ▶ • The signed paper contract
- ▶ • **The physical hard drive or USB device**
- ▶ • **The computer itself—chassis, keyboard, and all**

# Best Evidence

- ▶ “Best evidence” is roughly defined as the best evidence that can be produced in court.
- ▶ To prove the content of a writing, recording, or photograph, the original writing, recording, or photograph is required

Examples of “best evidence” include:

- ▶ • A photo of the crime scene
- ▶ • A copy of the signed contract
- ▶ • A file recovered from the hard drive
- ▶ • A bit-for-bit snapshot of a network transaction

# Direct Evidence

- ▶ “Direct evidence” is the testimony offered by a direct witness of the act or acts in question
- ▶ Direct evidence is usually admissible, so long as it's relevant. What other people witnessed can have a great impact on a case.

Examples of “direct evidence” can include:

- ▶ “I watched him crack passwords using John the Ripper and a password file he shouldn't have.”
- ▶ • “I saw him with that USB device.”

# Circumstantial Evidence

- ▶ In contrast to “direct evidence,” “circumstantial evidence” is evidence that does not directly support a specific conclusion. **Rather, circumstantial evidence may be linked together with other evidence and used to deduce a conclusion.**
- ▶ the primary mechanism used to link electronic evidence and its creator.”<sup>7</sup> Often, circumstantial evidence is used to establish the author of emails, chat logs, or other digital evidence.
- ▶ In turn, authorship verification is necessary to establish authenticity, which is required for evidence to be admissible in court

Examples of “circumstantial evidence” can include:

- ▶ • An email signature
- ▶ • A file containing password hashes on the defendant’s computer
- ▶ • The serial number of the USB device

# Hearsay

- ▶ “Hearsay” is the label given to testimony offered second-hand by someone who was not a direct witness of the act or acts in question
- ▶ Digital evidence can be classified as hearsay if it contains assertions created by people.

Examples of “hearsay” can include

- ▶ “The guy told me he did it.”
- ▶ • “He said he knew who did it, and could testify.”
- ▶ • “I saw a recording of the whole thing go down.”
- ▶ • A text file containing a personal letter



# Business Records

- ▶ Business records can include any documentation that an enterprise routinely generates and retains as a result of normal business processes, and that is deemed accurate enough to be used as a basis for managerial decisions

Examples of “business records” can include:

- ▶ Contracts and other employment agreements
- ▶ Invoices and records of payment received
- ▶ Routinely kept access logs
- ▶ /var/log/messages

# Digital Evidence

- ▶ “Digital evidence” is any documentation that satisfies the requirements of “evidence” in a proceeding, but that exists in electronic digital form.

Examples of “digital evidence” include:

- ▶ • Emails and IM sessions
- ▶ • Invoices and records of payment received
- ▶ • Routinely kept access logs
- ▶ • /var/log/messages

# Network-Based Digital Evidence

- ▶ “Network-based digital evidence” is digital evidence that is produced as a result of communications over a network.
- ▶ The primary and secondary storage media of computers (e.g., the RAM and hard drives) tend to be fruitful fodder for forensic analysis.

Examples of “network-based digital evidence” can include:

- ▶ • Emails and IM sessions
- ▶ • Browser activity, including web-based email
- ▶ • Routinely kept packet logs
- ▶ • /var/log/messages

# Challenges Relating to Network Evidence

- ▶ • **Acquisition:** It can be difficult to locate specific evidence in a network environment.
- ▶ • **Content :** metadata, network devices may or may not store evidence, due to limited storage capacity.
- ▶ • **Storage:** Network devices commonly do not employ secondary or persistent storage.
- ▶ • **Privacy:** Depending on jurisdiction, there may be legal issues involving personal privacy that are unique to network-based acquisition techniques.
- ▶ • **Seizure:** Seizing a network device can be much more disruptive. In the most extreme cases, an entire network segment may be brought down indefinitely.
- ▶ • **Admissibility:** Network forensics is a newer approach to digital investigations. There are sometimes conflicting or even nonexistent legal precedents for admission of various types of network-based digital evidence

# Network Forensics Investigative Methodology (OSCAR)

- ▶ • Obtain information
- ▶ • Strategize
- ▶ • Collect evidence
- ▶ • Analyze
- ▶ • Report

Dr. Vijeta Khare

# Obtain Information

## ▶ The Incident

- ▶ • Description of what happened, • Date, time, and method of incident discovery, • Persons involved, • Systems and data involved, • Actions taken since discovery, • Summary of internal discussions, • Incident manager and process, • Legal issues, • Time frame for investigation/recovery/resolution, • Goals

## ▶ The Environment

- ▶ Business model, • Legal issues, • Network topology (request a network map, etc. if you do not have one), • Available sources of network evidence, • Organizational structure, • Incident response management process/procedures • Communications systems • Resources available

# Strategize

Here are some tips for developing an investigative strategy:

- ▶ • Understand the goals and time frame of the investigation.
- ▶ • List your resources, including personnel, time, and equipment.
- ▶ • Identify likely sources of evidence.
- ▶ • For each source of evidence, estimate the value and cost of obtaining it.
- ▶ • Prioritize your evidence acquisition.
- ▶ • Plan the initial acquisition/analysis.
- ▶ • Decide upon method and times of regular communication/updates.
- ▶ • Keep in mind that after conducting your initial analysis, you may decide to go back and acquire more evidence. Forensics is an iterative process.

# Collect Evidence

- ▶ There are three components you must address every time you acquire evidence:
- ▶ **Document**—Make sure to keep a careful log of all systems accessed and all actions taken during evidence collection
- ▶ **Capture**—Capture the evidence itself. This may involve capturing packets and writing them to a hard drive, copying logs to hard drive or CD, or imaging hard drives of web proxies or logging servers.
- ▶ **Store/Transport**—Ensure that the evidence is stored securely and maintain the chain of custody. Keep an accurate, signed, verifiable log of the person who have accessed or possessed the evidence.





## **Best practices for evidence collection include:**

- ▶ • Acquire as soon as possible, and lawfully
- ▶ • Make cryptographically verifiable copies
- ▶ • Sequester the originals under restricted custody and access (or your earliest copy, when the originals are not available)
- ▶ • Analyze only the copies
- ▶ • Use tools that are reputable and reliable
- ▶ • Document everything you do!

# Analyze

- ▶ Correlation: between data, timestamps and source
- ▶ Timeline: timeline of whole activity
- ▶ Events of Interest: Isolate events of great interest
- ▶ Corroboration: False positives verification
- ▶ Recovery of additional evidence
- ▶ Interpretation

# Report

## **The report that you produce must be:**

- Understandable by nontechnical laypeople, such as:
  - ▶ – Legal teams
  - ▶ – Managers
  - ▶ – Human Resources personnel
  - ▶ – Judges
  - ▶ – Juries
- Defensible in detail
- Factual



# EVIDENCE ACQUISITION



# Active Vs. Passive Network Monitoring

## Active Network Monitoring

Works on specific aspects to analyze network performance.

Produces small amounts of data.

Used to find and report issues such as packet loss, jitter, HTTP response time, etc.

Can measure traffic inside & outside the network.

## Passive Network Monitoring

Provides a complete view of the network's performance.

Produces large amounts of data.

Used to identify elements that consume the highest amount of bandwidth.

Can measure the traffic inside the network only.

Passive evidence acquisition is the practice of gathering forensic-quality evidence from networks without emitting data at Layer 2 and above.

Active or interactive evidence acquisition is the practice of collecting evidence by interacting with stations on the network

# Physical Interception

- ▶ **Cables**

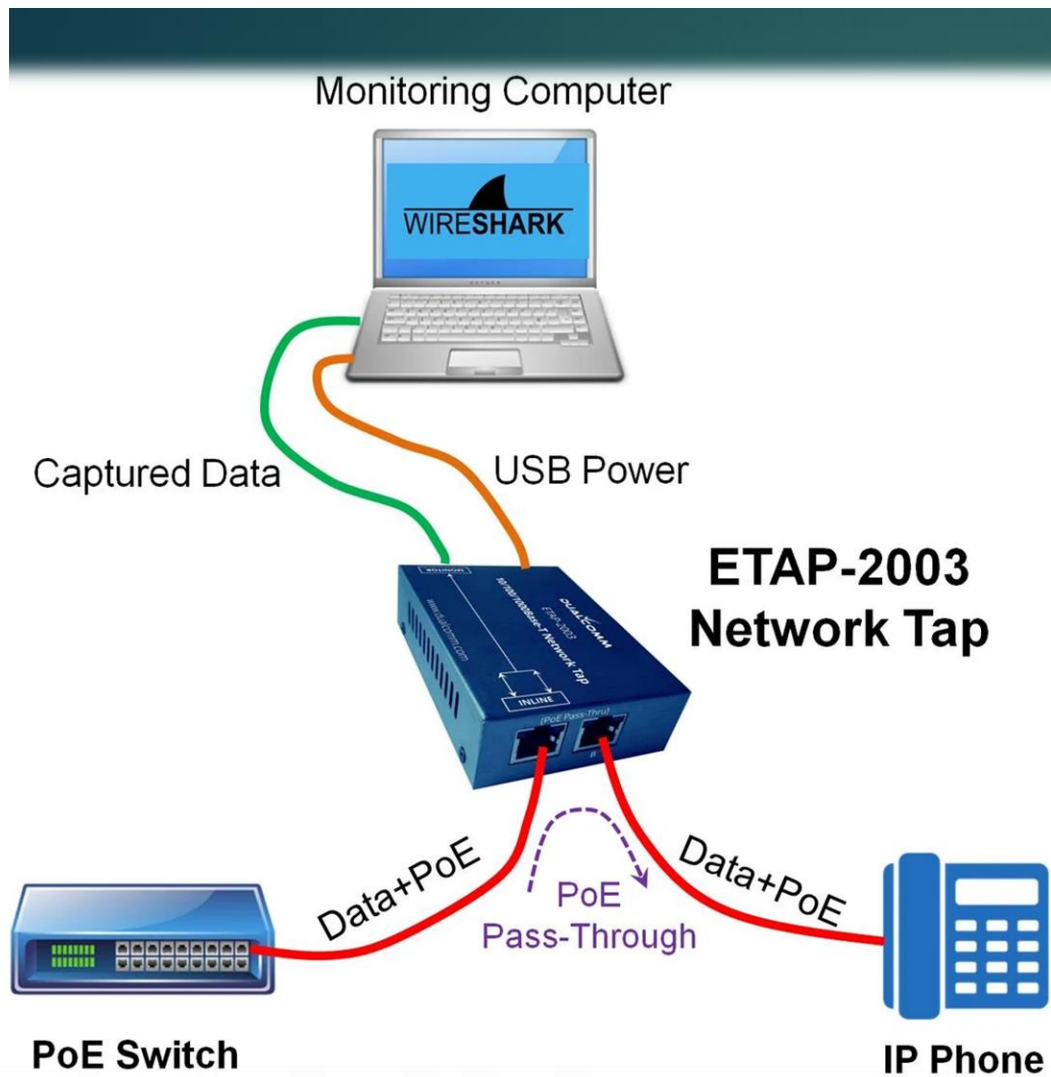
- ▶ **Copper**

- ▶ Coaxial: In most cases where coax is used, if you can tap the single copper core, you can access the traffic to and from all stations that share the physical medium.
    - ▶ Twisted Pair (TP): If you put a commercial TP network tap inline, it can capture all voltages for all twisted pairs in the cable.

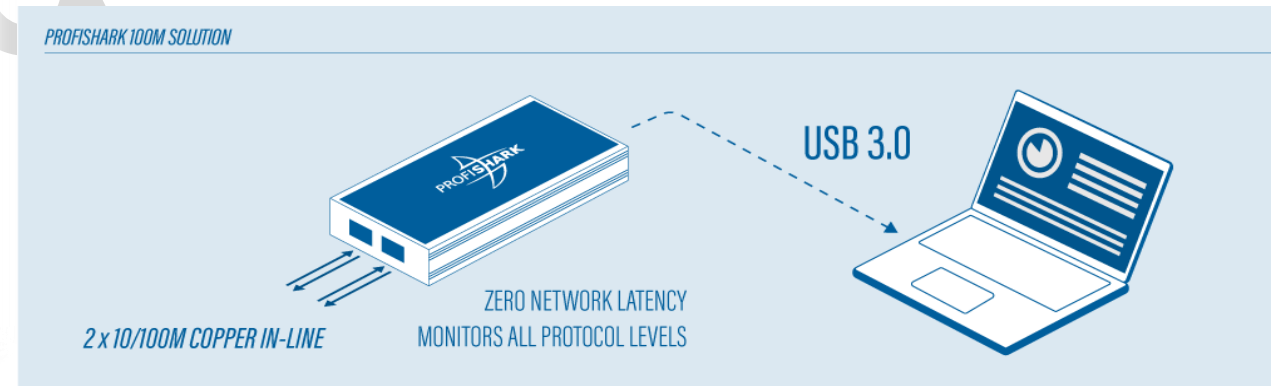
- ▶ **Optical**

# Inline network tap

- ▶ is a system that monitors events on a local network.
- ▶ hardware device, which provides a way to access the data flowing across a computer network
- ▶ Network TAPs are inserted between network devices, like a switch and router, where they copy the data without compromising network integrity.
- ▶ Network taps commonly have four ports: two connected inline to facilitate normal traffic, and two sniffing ports, which mirror that traffic (one for each direction).
- ▶ Insertion of an inline network tap typically causes a brief disruption, since the cable must be separated in order to connect the network tap inline.
- ▶ Network forensic analysts should keep in mind that every additional break in a cable is a potential point of failure; therefore inline insertion of network taps necessarily increase the risk of network disruption.



## Inline tap





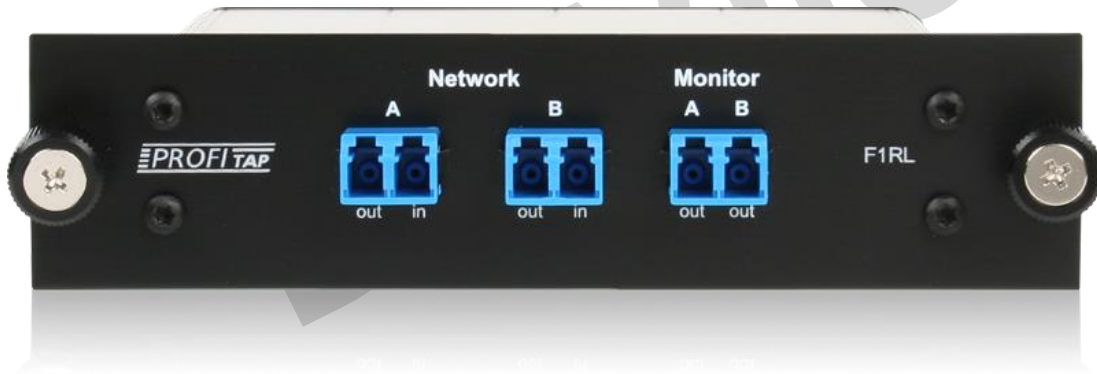
# Vampire Taps

- ▶ device for physically connecting a station, typically a computer, to a network that uses 10BASE5 cabling. This device clamps onto and "bites" into the cable




# Fiber Optic Taps

- ▶ To place a network tap inline on a fiber optic cable, network technicians splice the optic cable and connect it to each port of a tap. This causes a network disruption.
- ▶ Inline optical taps may cause noticable signal degradation.
- ▶ Network engineers often use tools called optical time-domain reflectometers (OTDR) to analyze and troubleshoot fiber optic cable signals



# Radio Frequency

- ▶ RF waves travel through the air, which is by nature a shared medium. As a result, WLAN traffic cannot be physically segmented in the way that switches segment traffic on a wired LAN.
- ▶ This attribute makes passive acquisition of WLAN traffic very easy—both for investigators and attackers.
- ▶ there are practical limitations on the distances over which stations can legally capture and receive data over 802.11 networks in the United States.
- ▶ Even when the Wi-Fi traffic is encrypted, there is commonly a single pre-shared key (PSK) for all stations.
- ▶ In this case, anyone who gains access to the encryption key can listen to all traffic relating to all stations (as with physical hubs)
- ▶ Furthermore, there are well-known flaws in common 802.11 encryption algorithms such as Wired Equivalent Privacy (WEP), which can allow investigators to circumvent or crack unknown encryption keys.

- 
- ▶ Regardless of whether or not Wi-Fi traffic is encrypted, investigators can gain a great deal of information by capturing and analyzing 802.11 management traffic. This information commonly includes:
    - ▶ • Broadcast SSIDs (and sometimes even nonbroadcast ones)
    - ▶ • WAP MAC addresses
    - ▶ • Supported encryption/authentication algorithms
    - ▶ • Associated client MAC addresses
    - ▶ • In many cases, the full Layer 3+ packet contents
  - ▶ **In order to capture wireless traffic, forensic investigators must first have the necessary hardware.**
  - ▶ **The network adapter must also support the specific 802.11 protocol in use (i.e., 802.11a/b/g cards do not necessarily support 802.11n).**

# Hubs

- ▶ A network hub is a dumb Layer 1 device that physically connects all stations on a local subnet to one circuit.
- ▶ A hub does not store enough state to track what is connected to it, or how. It maintains no knowledge of what devices are connected to what ports.
- ▶ When the hub receives a frame, it retransmits it on all other ports.
- ▶ Therefore, every device connected to the hub physically receives all traffic destined to every other device attached to the hub.
- ▶ **Investigators must be careful when using hubs as traffic capture devices. The investigator sees all traffic on the segment, but so can everyone else.**

# Switches

- ▶ Switches are the most prevalent Layer 2 device.
- ▶ Even the simplest switch maintains a CAM table, which stores MAC addresses with corresponding switch ports.
- ▶ A MAC address is an identifier assigned to each station's network card.
- ▶ The purpose of the CAM table is to allow the switch to isolate traffic on a port-byport basis so that each individual station only receives traffic that is destined for it, and not traffic destined for other computers.

# Obtaining Traffic from Switches

- ▶ Investigators can, and often do, capture network traffic using switches.
- ▶ Even though by default switches only send traffic to the destination port indicated in the frame, switches with sufficient software capabilities can be configured to replicate traffic from one or more ports to some other port for aggregation and analysis.
- ▶ The most common term is Cisco's Switched Port Analyzer (SPAN) and Remote Switched Port Analyzer (RSPAN).
- ▶ Switches have varying port mirroring capabilities.
- ▶ Port mirroring is inherently limited by the physical capacity of the switch itself.
- ▶ To sniff traffic from a switch, attackers use one of two common methods.
  - ▶ First, the attacker can flood the switch with bogus information for the CAM table by sending it many Ethernet packets with different MAC addresses.
  - ▶ Second, an attacker can conduct an "ARP spoofing" attack.



# Traffic Acquisition Software

- ▶ most common software libraries used for recording, parsing, and analyzing captured packet data: **libpcap** and **WinPcap**
- ▶ tools based on these libraries, including tcpdump, Wireshark, snort and others



# libpcap and WinPcap

- ▶ **Libpcap** is a UNIX C library that provides an API for capturing and filtering data link layer frames from arbitrary network interfaces.
- ▶ Different UNIX systems have different architectures for processing link-layer frames.
- ▶ The purpose of libpcap was to provide a layer of abstraction so that programmers could design portable packet capture and analysis tools.
- ▶ In 1999, the Computer Networks Group (NetGroup) in the Politecnico di Torino published **WinPcap**, a library based on libpcap that was designed for Windows systems.
- ▶ **The most popular packet sniffing and analysis tools today are based on the libpcap libraries. These include tcpdump, Wireshark, Snort, nmap, ngrep, and many others**
- ▶ A quintessential feature of libpcap-based utilities is that they can capture packets at Layer 2 from just about any network interface device and store them in a file for later analysis.

# The Berkeley Packet Filter (BPF) Language

- ▶ Libpcap includes an extremely powerful filtering language called the “Berkeley Packet Filter” (BPF) syntax.
- ▶ BPF allows you to filter traffic based on value comparisons in fields for Layer 2, 3, and 4 protocols.
- ▶ BPF invocations can be extremely simple, constructed from primitives such as “host” and “port” specifications, or very arcane constructions involving specific field values by offset (even down to individual bits).
- ▶ BPF filters can also consist of elaborate conditional chains, nesting logical ANDs and ORs.

# BPF Primitives

- ▶ By far, the easiest way to construct a BPF filter is to use BPF primitives to refer to specific protocols, protocol elements, or qualities of a packet capture
- ▶ The manual specifies three different kinds of qualifiers:
  - ▶ • **type** qualifiers say what kind of thing the id name or number refers to. Possible types are host, net, port and portrange.
  - ▶ • **dir** qualifiers specify a particular transfer direction to and/or from id. Possible directions are src, dst, src or dst, src and dst, addr1, addr2, addr3, and addr4.
  - ▶ • **proto** qualifiers restrict the match to a particular protocol. Possible protos are ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp.

## BPF Primitives

- ▶ Commonly used BPF primitives include:
- ▶ • host id, dst host id, src host id
- ▶ • net id, dst net id, src net id
- ▶ • ether host id, ether dst host id, ether src host id
- ▶ • port id, dst port id, src port id
- ▶ • gateway id, ip proto id, ether proto id
- ▶ • tcp, udp, icmp, arp
- ▶ • vlan id

There are many other BPF primitives.

**Example: 'host 192.168.0.1 and not host 10.1.1.1 and (port 138 or port 139 or port 445) '**

# Filtering Packets by Byte Value

- ▶ In addition to primitive comparisons, the BPF language can be used to compare the values of any byte-sized (or multibyte-sized) fields within a frame.
- ▶ **Important: Byte offsets are counted starting from 0!**
- ▶ Here are some examples: **`ip[9] != 1`**, **`icmp[0] = 3`** and **`icmp[1] = 0`**, **`tcp[0:2] = 31337`**, **`ip[12:4] = 0xC0A80101`**

## Filtering Packets by Bit Value

- ▶ we cite a specific byte or bytes (as explained above), and then compare them bit-by-bit to some value that we hope to find. This is called “bitmasking.”
- ▶ Let's suppose that we'd like to filter for packets where IP options are set (in other words, the IP header length is greater than 20 bytes). The low-order nibble of the IP header represents the IP header length, measured in 32-bit “words” (each word is four bytes long).
- ▶ To find all packets where the IP header is greater than 20 bytes in length, we need to match packets where the low-order nibble is greater than five (5 words \* 4 bytes per word = 20 bytes). To accomplish this, we create a BPF filter with a bitmask of “00001111” (0x0F), which is logically “AND-ed” with the targeted value. The resulting expression is: **`ip[0] & 0x0F > 0x05`**

# tcpdump

- ▶ Tcpdump was designed as a UNIX tool
- ▶ The basic purpose of tcpdump is to capture network traffic and then print or store the contents for analysis.
- ▶ Tcpdump captures traffic bit-by-bit as it traverses any physical media suitable for conducting link-layer traffic
- ▶ Beyond merely capturing packets, tcpdump can decode common Layer 2 through 4 Protocols
- ▶ The decoded packets can be displayed in hexadecimal or in the ASCII equivalents (where the data is textual), or both.

# Fidelity:

- ▶ One reason that tcpdump is such a powerful tool is that it is capable of capturing traffic with high fidelity, to the degree that the resulting packet capture can constitute evidence admissible in court.
- ▶ tcpdump's ability to capture packets may be limited by the clock speed of the processor in the capturing workstation
- ▶ on high-traffic networks, investigators may also be limited by disk space.
- ▶ One crucial configuration option for capturing packets using tcpdump is the snapshot length, known as "snaplen." Snaplen represents the number of bytes of each frame that tcpdump will record.
- ▶ If the chosen snaplen is too short, data will be missing from every frame and can never be recovered. If the snaplen is too long, it may cause performance degradation, limit the volume of traffic that can be stored
- ▶ Once upon a time, many people recommended using a snaplen of 1,514 bytes because the maximum transmission unit (MTU) of Ethernet is 1,500 bytes. Since the Ethernet header itself is 14 bytes long, this meant that the total frame length was 1,514 bytes.
- ▶ Later versions of tcpdump allowed the user to specify "0" for the snaplen, which would tell tcpdump to automatically capture the entire frame, no matter how long it was. (backward compatibility)

# Filtering Packets with tcpdump

- ▶ One good strategy for analysis of large volumes of traffic is to begin by filtering out any types of traffic that are not related to the investigation.
- ▶ Here's an example that shows how tcpdump is used to display traffic from the "eth0" network interface, excluding TCP port 80 traffic:

```
# tcpdump -nni eth0 'not (tcp and port 80) '
```

- ▶ • tcpdump -i eth0 -w great\_big\_packet\_dump.pcap
- ▶ This is the simplest case of listening on interface eth0 and writing all of the packets out to a single monolithic file.





tcpdump command-line usage:

- ▶ -i Listen on interface (eth0 , en1 , 2)
- ▶ -n Do not resolve addresses to names.
- ▶ -r Read packets from a pcap file
- ▶ -w Write packets to a pcap file
- ▶ -s Change the snapshot length from the default
- ▶ -C With -w, limit the capture file size , and begin a new file when it is exceeded
- ▶ -W With -C, limit the number of capture files created , and begin overwriting and rotating when necessary
- ▶ -D List available adapters (WinDump only)



```
tcpdump -i eth0 -s 0 -w targeted_full_packet_dump.pcap 'host 10.10.10.10'
```

- ▶ Here we introduce a simple BPF filter to grab and store in their entirety only those packets sent to or from the host at the address “10.10.10.10.”
- ▶ **tcpdump -i eth0 -s 0 -w RFC3514\_evil\_bits.pcap 'ip[6] & 0x80 != 0'**
- ▶ Finally, we introduce a more complicated BPF filter, in which we target the first byte of the IP fragmentation fields (byte offset 6). We employ a bitmask to narrow our inspection to the single highest order bit, most commonly known as the IP “reserved bit,” and we capture and store the packet only if the reserved bit is nonzero.
- ▶ **The Evil Bit (RFC 791):** In original IP the very first, or “high order,” bit of the sixth byte offset would be “reserved”
- ▶ Bellovin suggested that any packet that had been built for malicious or evil intent must set this bit to one.
- ▶ Later, Jason Mansfield has
- ▶ created an “evil bit changer.” This handy utility captures traffic or reads it from a file, sets the Evil Bit to “1,” recalculates the IP header checksum, and then forwards the frame along to its intended destination

# Wireshark

- ▶ Hands On Practice

Dr. Vijeta Khare



Dr. Vijeta Khare