Evidence Acquisition

Network Forensics

Jae Woong Joo
Table of Contents

3.1 Physical Interception
3.2 Traffic Acquisition Software
3.3 Active Acquisition
3.4 Conclusion
3.1 Physical Interception

- It is possible to obtain network traffic without sending or modifying any data frames on the network.

- There are many ways to transmit data over physical media, and just as many ways to intercept it.
  - The simplest case is a station connected to another station over a physical conduit, such as a copper cable.

- Forensic investigators can passively acquire network traffic by intercepting it as it is transmitted across cables, through the air, or through network equipment such as hubs and switches.
Pidgeon Sniffing?

- IP networks can be built upon a wide variety of physical media

- For example, RFC 1149, “Standard for the transmission of IP datagrams on avian carriers”
  - Avian carriers can provide high delay, low throughput, and low altitude service
  - The connection topology is limited to a single point-to-point path for each carrier
  - Avian networks are fairly resistant to passive interception
3.1.1 Cables

- Cables allow for point-to-point connections between stations.
- The most common materials for cables are copper and fiber.
- Each of these can be sniffed, although the equipment and side effects vary based on the physical media.
  - Copper
  - Optical
  - Intercepting Traffic in Cables
3.1.1 Cables (Copper)

- **Coaxial**
  - Coaxial cable, or “coax,” consists of a single copper wire core wrapped in insulation and covered with a copper shield.
  - Since the transmission media is the single copper core, all stations on the network must negotiate the transmission and reception of signals.
  - 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)**
  - TP cables contain multiple pairs of copper wires.
  - TP wires are typically deployed in a star topology.
  - This means that by tapping one pair of TP wires on a switched network, you may receive traffic relating to only one end station.
  - If you put a commercial TP network tap inline, it can capture all voltages for all twisted pairs in the cable.
3.1.1 Cables (Optical)

- Fiber optic cables consist of thin strands of glass which are bundled together in order to transmit signals across a distance.

- Light is transmitted into the fiber at one end and travels along an optic fiber, reflecting constantly against the walls until it reaches an optical receiver at the other end.
  - The light naturally degrades during travel, and depending on the length of the fiber optic cable
  - an optical regenerator may be used to amplify the light signal in transit.
3.1.1 Cables (Intercepting Traffic in Cables)

- There are a variety of tools available for intercepting traffic in cables
  - Inline Network Taps
  - Vampire Taps
  - Induction coils
  - Fiber Optic Taps
3.1.1 Cables (Intercepting Traffic in Cables)

- **Inline Network Taps**
  - An inline network tap is a Layer 1 device, which can be inserted inline between two physically connected network devices.
  - Four ports: two connected inline to facilitate normal traffic, and two sniffing ports, which mirror that traffic.
  - 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.
    - Allows for extremely high-fidelity packet captures.
3.1.1 Cables (Intercepting Traffic in Cables)

- **Vampire Taps**
  - “Vampire taps” are devices that pierce the shielding of copper wires in order to provide access to the signal within.
  - The cable does not need to be severed in order for a vampire tap to be installed.
  - Inserting a vampire tap, even if done correctly, can bring down the link on a TP cable.
3.1.1 Cables (Intercepting Traffic in Cables)

- **Induction Coils**
  - All wires conducting voltages emit various electromagnetic signals outside of the intended channel.

  - Such electromagnetic radiation is more pronounced in unshielded wires, such as UTP, due to the lack of shielding that plastic sheathing affords.

  - it is theoretically possible to introduce what is called an “induction coil” alongside such wiring in order to translate the laterally emitted signals into their original digital form.

  - Induction coils are devices that essentially transform the magnetism of weak signals to induce a much stronger signal in an external system.
3.1.1 Cables (Intercepting Traffic in Cables)

- **Fiber Optic Taps**
  - Inline network taps work similarly for fiber optic cables and copper cables.
  - 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.
  - Network engineers often use tools called optical time-domain reflectometers (OTDR) to analyze and troubleshoot fiber optic cable signals.
  - OTDRs can also be used to locate breaks in the cable, including splices inserted for taps.
3.1.2 Radio Frequency (1/2)

- Since the late 1990s, radio frequency has become an increasingly popular medium for transmission of packetized data and Internet connectivity.

- The Institute of Electrical and Electronics Engineers (IEEE) published a series of international standards (“802.11”) for wireless local area network (WLAN) communication.

- The term “Wi-Fi” is used to refer to certain types of RF traffic, which include the IEEE 802.11 standards.

- 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.
  - physical media limitations, all WLAN transmissions may be observed and intercepted by all stations within range.
  - Stations can capture the RF traffic
  - This attribute makes passive acquisition of WLAN traffic very easy—both for investigators and attackers.
3.1.2 Radio Frequency (2/2)

- This information commonly includes:
  - Broadcast SSIDs
  - WAP MAC addresses
  - Supported encryption/authentication algorithms
  - Associated client MAC addresses
  - In many cases, the full Layer 3+ packet contents
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. All the devices on the local segment that the hub provides are physically connected. Share the same physical medium.
When the hub receives a frame, it retransmits it on all other ports.

Every device connected to the hub physically receives all traffic destined to every other device attached to the hub.

If a hub exists in the network, then you can connect to it and trivially sniff all of the traffic on the segment.

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 are the most prevalent Layer 2 device. Like hubs, they also connect multiple stations together to form a LAN.

Unlike hubs, switches use software to keep track of which stations are connected to which ports, in its CAM table.

When a switch receives a packet, it forwards it only to the destination station’s port. Individual stations do not physically receive each other’s traffic.
  - This means that every port on a switch is its own collision domain.

Switches operate at Layer 2 (the data-link layer), and sometimes Layer 3 (the network layer).
3.1.4 Switches (2/2)

- Even the simplest switch maintains a CAM table.

- The purpose of the CAM table is to allow the switch to isolate traffic on a port-by-port basis so that each individual station only receives traffic that is destined for it, and not traffic destined for other computers.

- Switches populate the CAM table by listening to arriving traffic. When a switch receives a frame from a device, it looks at the source MAC address and remembers the port associated with that MAC address.

- Later, when the switch receives a packet destined for that device, it looks up the MAC address and corresponding port in the CAM table.

- It then sends the packet only to the appropriate port, encapsulated with the correct Layer 2 Ethernet address.

- In this way, a switch segments the traffic endpoint-by-endpoint, even while technically sharing the same physical medium.
3.1.4.1 Obtaining Traffic from Switches

- Investigators can, and often do, capture network traffic using switches.
- Switches have varying port mirroring capabilities, depending on their make and model.
- Port mirroring is inherently limited by the physical capacity of the switch itself.
- You need administrative access to the switch’s operating system in order to configure port mirroring.
  - Once you have mirrored the ports of interest, you can connect a sniffer to the mirroring port and capture all of the traffic.
- If you don’t have administrative access, it is still possible to sniff traffic from a switch.
3.1.4.1 Obtaining Traffic from Switches

Sniffing on Switches

• The attacker can flood the switch with bogus information for the CAM table by sending it many Ethernet packets with different MAC addresses.

  This attack is referred to as “MAC flooding.” Once the CAM table is filled, many switches by default will “fail open,” and send all traffic for systems not in the CAM table out to every port.

• An attacker can conduct an “ARP spoofing” attack.
  - In an ARP spoofing attack, the attacker broadcasts bogus ARP packets, which link the attacker’s MAC address to the victim’s IP address. Other stations on the LAN add this bogus information to their ARP tables, and send traffic for the router’s IP address to the attacker’s MAC address instead.
  - This causes all IP packets destined for the victim station to be sent instead to the attacker.
3.2 Traffic Acquisition Software

Once you gain physical access to network traffic, you need software to record it

- **libpcap** and **WinPcap**
- The Berkeley Packet Filter (BPF) Language
  - **BPF Primitives**
  - Filtering Packets by Byte Value
  - Filtering Packets by Bit Value
- **Tcpdump**
  - **Fidelity**
  - Filtering Packets with tcpdump
- **Wireshark**
- **tshark**
- **dumpcap**
3.2.1 libpcap and WinPcap

- Libpcap is a UNIX C library that provides an API for capturing and filtering data linklayer frames from arbitrary network interfaces.

- The purpose of libpcap was to provide a layer of abstraction so that programmers could design portable packet capture and analysis tools.

- 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.

- Many tools based on libpcap also include specialized functionality, such as the ability to merge packet captures, to split a capture up by TCP streams, or to conduct regular expression searches on packet contents.
3.2.2 BPF Language

- Libpcap includes an extremely powerful filtering language called the “Berkeley Packet Filter” (BPF) syntax.

- Using BPF filters, you can decide which traffic to capture and inspect and which traffic to ignore.

- 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.

- BPF filters can also consist of elaborate conditional chains, nesting logical ANDs and ORs.
3.2.2.1 BPF Primitives

• 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.

• Primitives, as defined by the “pcap-filter” manual page, “usually consist of an id (name or number) preceded by one or more qualifiers.”

• 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.

• Check the manual pages for your version of libpcap or libpcap-based tool.
3.2.2.2 Filtering Packets by Byte Value

- The BPF language can be used to compare the values of any byte-sized fields within a frame.

- The BPF language provides syntax to specify the byte offset relative to the beginning of common Layer 2, 3, and 4 protocols.
  - **Important**: Byte offsets are counted starting from 0!

- `ip[8] < 64` This filter would match all packets in which the single byte field starting at the eighth byte offset of the IP header, is less than 64.
3.2.2.3 Filtering Packets by Bit Value (1/2)

- We cite a specific byte or bytes, and then compare them bit-by-bit to some value that we hope to find.
  - This is called “bitmasking.”
- Essentially, we must identify one or more byte-sized chunks of data that contain the bits we are interested in, and then specify the particular bits of interest using a binary representation known as a “bitmask.”
- In the bitmask, “1” represents a bit of interest and “0” represents a bit we choose to ignore.

- The IP header is a minimum of 20 bytes long, by specification.
- It is possible to include optional header fields which increase the IP header length.
- However, IP options are not commonly used in practice
3.2.2.3 Filtering Packets by Bit Value (2/2)

- Let’s suppose that we’d like to filter for packets where IP options are set.

- The low-order nibble of the IP header represents the IP header length, measured in 32-bit “words”.

- 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.

- 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:
  - \( \text{ip}[0] \& 0x0F > 0x05 \)
3.2.3 tcpdump

- Tcpdump is a tool for capturing, filtering, and analyzing network traffic.
- 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.
  - Tcpdump can decode common Layer 2 through 4 protocols.

- There are two ways that tcpdump is most commonly employed.
  - It is used to facilitate on-the-fly analysis for troubleshooting network issues in a tactical way.
  - Tcpdump is also frequently used to capture traffic of interest passing on a target segment over a longer period of time, and store it for offline analysis and perhaps even future correlation with other data.
3.2.3.1 Fidelity (1/2)

- One reason that tcpdump is such a powerful tool
- it is capable of capturing traffic with high fidelity
- to the degree that the resulting packet capture can constitute evidence admissible in court.
  - However, the quality of the packet capture can be impacted by hardware limitations and configuration constraints.

- Especially on high-traffic networks, investigators may also be limited by disk space.
  - If the capturing workstation does not have enough disk space
  - Necessary to either filter traffic upon capture or provide more disk storage.
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.
- Selecting the correct snaplen for a packet capture is critical.
- 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.
- Perhaps cause violations of regulations such as the United States Wiretap Act, which prohibit capturing communications contents except in certain circumstances.

You should limit snaplen to the smallest number that will capture the protocol information you’re interested in.”
3.2.3.2 Filtering Packets with tcpdump (1/3)

- Filtering during capture is very important because resources such as disk space, CPU cycles, and traffic aggregation capacity are always limited.
  - however, can cause loss of evidence, which can never be recaptured.

- tcpdump is a libpcap-based tool
  - it incorporates the BPF language, which investigators can use to filter traffic during capture and analysis.
• 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.

• For example
  ▫ imagine that we have a packet capture that contains 75% web traffic (TCP port 80), which is fairly typical for an enterprise network.
    • # tcpdump -nni eth0 'not (tcp and port 80)'
3.2.3.2 Filtering Packets with tcpdump (3/3)

- tcpdump -i eth0 -w great_big_packet_dump.pcap
- tcpdump -i eth0 -s 0 -w biggest_possible_packet_dump.pcap
- tcpdump -i eth0 -s 0 -w targeted_full_packet_dump.pcap 'host 10.10.10.10'
- tcpdump -i eth0 -s 0 -C 100 -w rolling_split_100MB_dumps.pcap
- tcpdump -i eth0 -s 0 -w RFC3514_evil_bits.pcap 'ip[6] & 0x80 != 0'
3.2.4 Wireshark

- Wireshark is a graphical, open-source tool designed for capturing, filtering, and analyzing traffic.
- Wireshark allows you to capture packets on any system network interface, assuming you have appropriate permissions to do so and your network card supports sniffing.
3.2.5 tshark

- Tshark is a command-line network protocol analysis tool that is part of the Wireshark distribution.
- It is libpcap-based, and can read and save files in the same standard formats as Wireshark.

The example
- shows tshark capturing traffic on the network interface “eth0,” filtering out all port 22 traffic, and storing the results in the file “test.pcap.”

```bash
# tshark -i eth0 -w test.pcap 'not port 22' Capturing on eth0 235
```
3.2.6 dumpcap

• The Wireshark distribution also comes with a command-line tool, “dumpcap,” which is specifically designed to capture packets.
• Dumpcap is a specialized tool designed just for capturing packets, it takes up fewer system resources, maximizing your capture capabilities.
• Dumpcap automatically writes packet captures to a file.

$ dumpcap -i eth0 -w test.pcap 'not port 22'
File: test.pcap
Packets: 12
Packets dropped: 0
3.3 Active Acquisition

- evidence lives in many places throughout a network
  - network traffic, network devices, including firewalls, web proxies, logging servers, and more.
- By definition, active evidence acquisition modifies the environment.

- Investigators should be highly aware of the various ways in which live acquisition modifies the devices and environment under investigation, and work to minimize the impact.
3.3.1 Common Interfaces

- Console
- Secure Shell (SSH)
- Secure Copy (SCP) and SSH File Transfer Protocol (SFTP)
- Telnet
- Simple Network Management Protocol (SNMP)
- Trivial File Transfer Protocol (TFTP)
- Web and proprietary interfaces
3.3.1.1 Console

- The console is an input and display system, usually a keyboard and monitor connected to a computer.
- Many network devices have a serial port that you can use to connect a terminal to the console.
- Forensic workstation is connected to the serial port.
- Can use the Linux “screen” command to connect to the console and log your session:
  - `$ screen -L /dev/ttyUSB0`
- Whenever possible, it is best to connect directly to the console of a network device rather than connecting remotely over the network.
3.3.1.2 Secure Shell (SSH)

- The Secure Shell protocol (SSH) is a common way for investigators to gain remote commandline access to systems containing network-based evidence.
  - Developed as a replacement for the insecure Telnet and rlogin, SSH encrypts authentication credentials and data in transit.
  - Network devices now support SSH as a method for remote command-line interaction.
    - `$ ssh -p 4022 sherri@remote.lmgsecurity.com`
  - Use SSH to run commands remotely
    - `$ ssh -p 4022 sherri@remote.lmgsecurity.com 'hostname'`
3.3.1.3 Secure Copy (SCP) and SFTP

- In addition to providing interactive command-line access, SSH implements the Secure Copy Protocol (SCP).
- which is a command-line utility designed to transfer files between networked systems.
  - `scp -P 4022 jonathan@remote . lmgsecurity .com :/ etc/ passwd`

- The SSH File Transfer Protocol, or SFTP, is an alternative protocol used in conjunction with SSH for secure file transfer and manipulation
  - It is more portable and offers more capabilities than SCP, but file transfer tends to be slower
3.3.1.4 Telnet (yes, Telnet)

- Telnet is a command-line remote communications interface
- Which was originally developed in 1969 and eventually standardized by the IETF in RFCs 854 and 855
- The Telnet client can be used to interact with a wide variety of servers, such as SMTP and HTTP
- Telnet had only limited security built-in
  - All transactions are in plain text, so authentication credentials and data are sent unencrypted across the wire.
- Despite its serious security drawbacks, in many cases Telnet is the only option for remote access to a network device
3.3.1.5 Simple Network Management Protocol (SNMP) (1/2)

- Using SNMP, you can poll networked devices from a central server
- Push SNMP information from remote agents to such a central aggregation point
- SNMP is frequently used as a medium for communicating and aggregating both network management information and security event data
  - In network forensics, SNMP is commonly used in one of two ways: event-based alerting and configuration queries.
3.3.1.5 Simple Network Management Protocol (SNMP) (2/2)

- Designed to be extensible through the definition of the “management information base”

- SNMP Operations
  - Here’s a list of the basic SNMP operations:
    - **Polling**: GET, GETNEXT, GETBULK
      - These operations are employed to retrieve information from the managed device, including routing tables, system uptime, hostname, ARP tables, CAM tables, and more
    - **Interrupt**: TRAP, INFORM
      - can receive events via the SNMP TRAP operation from many sources as they occur and aggregate them.
    - **Control**: SET
      - SNMP can also be used to control the configuration of remote devices using the “SET” command.
3.3.1.6 Trivial File Transfer Protocol (TFTP)

- Designed as a simple, automated means of transferring files between remote systems.
  - Designed before most people were concerned about “bad actors” on the network.
- The features of TFTP are extremely limited
- The design goals was to keep the service very small so that it could run on systems with extremely limited storage space and memory
- It has been incorporated into many network devices
  - Forensic analysts may need to use TFTP to export files from a router, switch, or other device that does not support SCP or SFTP for such operations.
3.3.1.7 Web and Proprietary Interfaces

• These days most commercial network devices, from DSL routers to wireless access points, come with a web-based management interface.

• Through HTTP or HTTPS, you can access configuration menus, event logs, and other data that the device contains.

• Typically, web interfaces are available by default as unencrypted HTTP sessions, in which case the login credentials and any data transferred over the connection is unencrypted and easily intercepted.
  ▫ Many vendors also offer SSL/TLS-encrypted web interfaces.

• Many vendors such as Cisco and netForensics have also developed Java-based crossplatform interfaces or other types of proprietary interfaces for their devices.
3.3.2 Inspection Without Access

- In many cases, it is desirable to gain information about a device’s configuration or state without accessing the device at all via an interface.

- It is possible to gather extensive information about a device’s configuration and state through external inspection, using port scanning, vulnerability scanning, and other methods.
3.3.2.1 Port Scanning

- Port scanning, using a tool such as nmap, is an effective way to retrieve information about open ports and software versions of a device.

- Note that port scanning is an active process, meaning that you will generate network traffic and, in the process, modify the state of the targeted device.
3.3.2.2 Vulnerability Scanning

- Vulnerability scanning is the next level of active external inspection.

- In addition to port scanning, vulnerability scanners test target systems for a wide variety of known vulnerabilities.

- If you are concerned that your target of interest may be compromised, this can sometimes provide strong clues as to how the compromise may have occurred.
3.3.3 Strategy

- Refrain from rebooting or powering down the device.
  - A lot of network-based evidence exists as volatile data in the memory of network devices.
- Connect via the console rather than over the network.
  - Connecting to a device over the network will necessarily generate network traffic.
- Record the system time
  - Even a small time skew can make it very difficult to correlate evidence
- Collect evidence according to level of volatility.
  - Collect the most volatile evidence first
- Record your investigative activities
  - Recording your own commands also helps you stay organized
  - Take screen captures, photos or recordings of your graphical connections.
Q & A

Network Forensics

Jae Woong Joo