Introduction to Security Networking and Packets

Ming Chow (mchow@cs.tufts.edu)

Twitter: @0xmchow

Tufts UNIVERSITY | SCHOOL OF ENGINEERING Computer Science

Learning Objectives

- By the end of this week, you will be able to:
  1. Dissect packet captures (PCAPs), network traffic
  2. Perform network reconnaissance and port scanning
  3. Understand the methods of conducting a distributed denial of service attack (DDoS)

Why Cover Networking and Network Security First?

- The "Connectivity" issue (recall Gary McGraw's "Trinity of Trouble")
- Where the "cool stuff" happens
- Critical to understanding the cyber attribution problem

What is the Cyber Attribution Problem?

- **Attribution** - "the action of regarding something as being caused by a person or thing."
- How do you attribute an act of war in traditional warfare?
  - Uniform of attackers
  - Types of weapons attackers used
  - Direction of strike
  - List goes on…
- What is cyber attribution like? See https://twitter.com/thegrugq/status/706545282645757952
  - So why is that?

What is Networking?

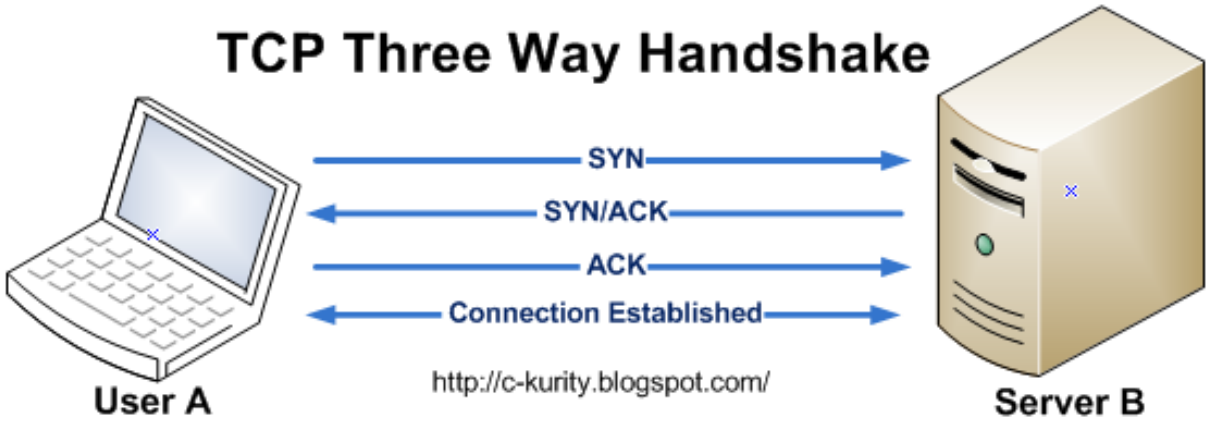
- Two or more computers talking to each other
- Basic definitions:
  - **Client** - A program running on your computer
    - Web browser - a client application that displays web pages (e.g., Chrome, Firefox, Microsoft Internet Explorer, Safari, Opera, lynx)
  - **Server** - A computer running web server software on a remote computer; delivers information to other clients
    - Example: Apache HTTP Server
  - **Internet** – The world's largest computer network
  - **World Wide Web (or the "web")** - A collection of web sites, pages, and content around the world
  - **Localhost** - home; this computer
  - **Socket** - an endpoint instance defined by an IP address and a port in the context of either a particular TCP connection or the listening state.
  - **Port** - a virtualization identifier defining a service endpoint (as distinct from a service instance endpoint aka session identifier); a number
    - Reference: https://stackoverflow.com/questions/152457/what-is-the-difference-between-a-port-and-a-socket

Abridged Analogy Describing How Two Computers Talk to Each Other

Telephone Conversation Between Two People	Conversation Between Two Computers
Telephone number	**IP address**. We will use IPv4 format extensively where an IP address is in octal format xxx.xxx.xxx.xxx where xxx is a number between 0-255 inclusive.
Telephone extension number	**Port number** - denotes a service provided by a computer. https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
Telephone lines	Ethernet cables
Telephone book, “Yellow Pages”	Domain Name Systems (DNS)

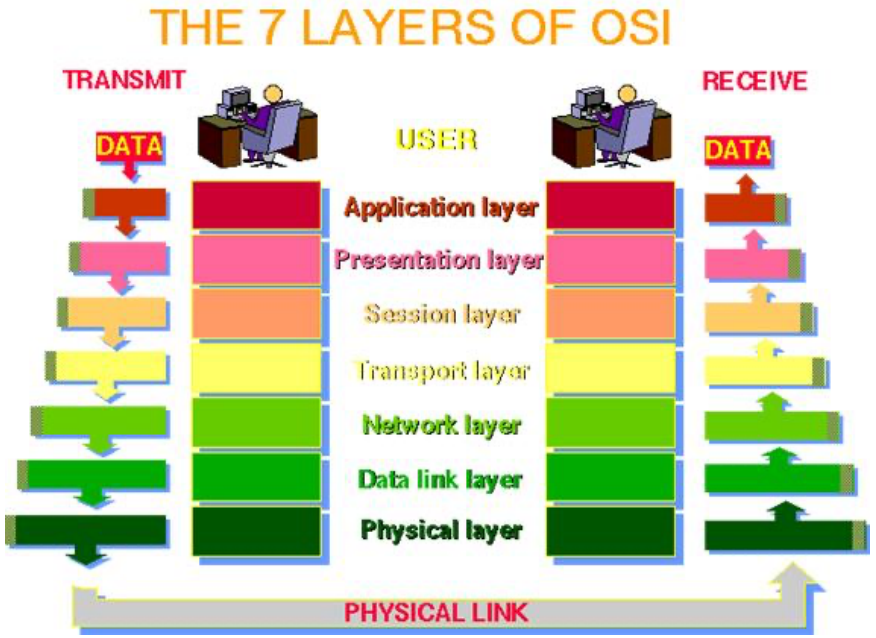
Abridged Analogy Describing How Two Computers Talk to Each Other (continued)

- The “three-way handshake” - method used by *TCP* set up a *TCP/IP* connection over an *Internet Protocol (IP)* based network
  - **IMPORTANT: note the TCP flags SYN, SYN/ACK, and ACK as they will come up again**
- References:
  - http://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml



How Two Computers Talk to Each Other

- The OSI model
  - OSI - Open Systems Interconnection
  - Provides standards that allow hardware to focus on one particular aspect of communication that applies to them and ignore others



The Seven Layers of the OSI Model

1. **Physical** - Lowest level, the bit level; primary role is communicating raw bit streams over physical medium (e.g., Ethernet cable and card, "wires")
2. **Data link** - Transferring data between two points connected by a physical layer; provides high level functions such as error correction and flow control (e.g., ARP, Ethernet)
3. **Network** – Middle ground; pass information between the lower and higher layers; provides addressing and routing (e.g., IP, ICMP) --delivery is NOT guaranteed
4. **Transport** - Provides transparent and reliable transfer of data between systems, including acknowledgement and segmentation (e.g., TCP, UDP)
5. **Session** - Establishes and maintains connections between network applications
6. **Presentation** - Allows for things like encryption and data compression (e.g., XML)
7. **Application** - The highest level interfaces, the services that you use on the Internet

Analogy to Understand the OSI Model via the US Postal Service

- **Physical** - The USPS' trucks, trains, and planes: this is how the letters actually get from point A to point B.
- **Data-link** - The envelope: you can't just put a handwritten letter in a mailbox and expect it to be sent somewhere.
- **Network** - The address: the USPS needs to know where to deliver the letter. This establishes a connection between two residences.
- **Transport** - Your name on the envelope: once it gets inside your house, it needs to be given to the correct person.
- **Session** - The standard letter format: this includes dating the letters, saying "dear so-and-so" and "yours truly."
- **Presentation** - The body of the letter itself: let's make sure both parties are writing in English.
- **Application** - The collection of letters exchanged: the point of the previous six layers was to enable the pen pal relationship between two people.
- *We will focus on the Network, Transport, and Application layers extensively*
- Source: https://www.quora.com/Can-you-explain-OSI-layers-and-TCP-IP-in-laymans-terms

Application Layer

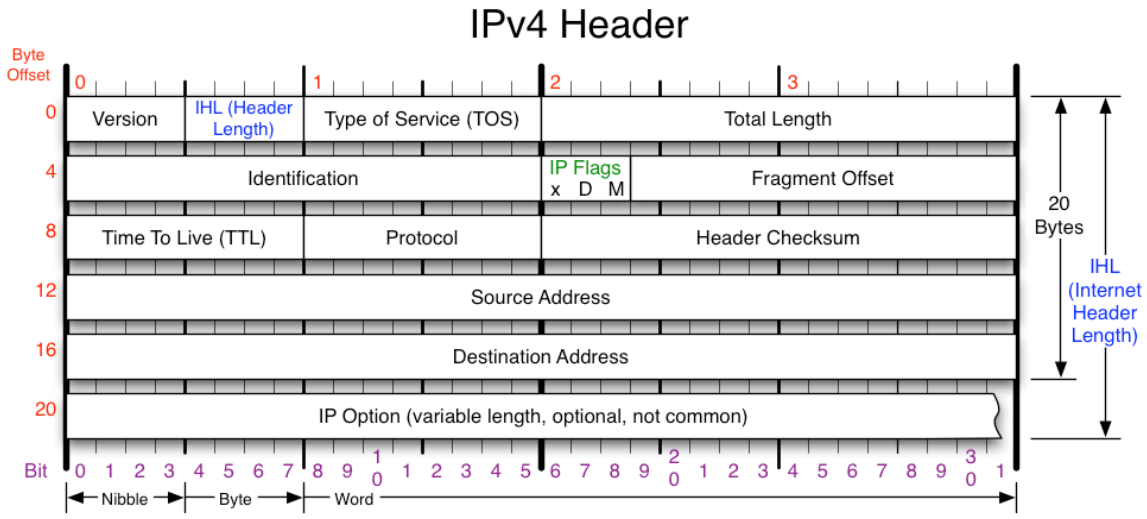
- The famous and insecure ones by default, data all unencrypted:
  - DNS – Domain Name Server (DNS)
    - Port 53
  - IMAP (Internet Message Access Protocol)
    - Email
    - Port 143
  - FTP (File Transfer Protocol)
    - File transfer
    - Port 21
  - HTTP (Hypertext Transfer Protocol)
    - The foundation of data communication for the World Wide Web
    - Port 80
  - Telnet
    - Protocol that allows you to connect to remote computers
    - Port 23
  - POP (Post Office Protocol)
    - Email
    - Port 110
    - Current version is 3 thus protocol is now known as POP3

Internet Protocol (IP)

- On the Network layer of OSI model
- Provides a connectionless, unreliable, best-effort datagram delivery service (delivery, integrity, ordering, non- duplication, and bandwidth is not guaranteed)
- RFC 791: http://www.ietf.org/rfc/rfc791.txt
  - RFC – Request For Comments, a publication from the Internet Engineering Task Force (IETF) and the Internet Society (ISOC), the principal technical development and standards-setting bodies for the Internet.

IP Header

- Source and reference: https://nmap.org/book/tcpip-ref.html



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

RFC 791

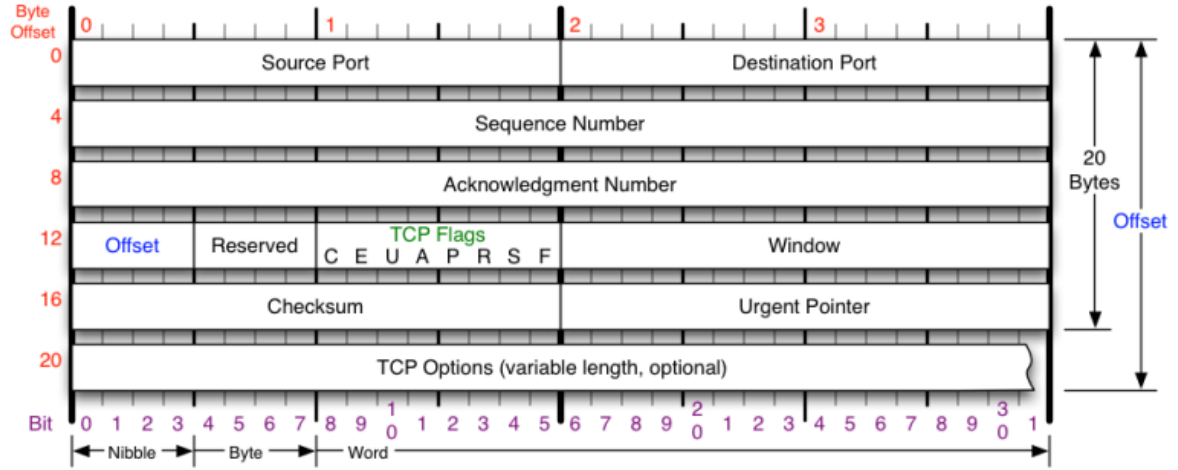
Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Transport Control Protocol (TCP)

- Guarantees delivery of data in proper order thanks to IP protocol; thus, it is commonly known as TCP/IP
- Transparent, bidirectional, and reliable
- On the Transport layer of OSI model
- RFC 793: http://www.ietf.org/rfc/rfc793.txt

TCP Header

- Source and reference: https://nmap.org/book/tcpip-ref.html



TCP Flags

C E U A P R S F

Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

TCP Options

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
4 Selective ACK ok
8 Timestamp

Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

Internet Control Message Protocol (ICMP)

- On Network layer of OSI model
- Testing and debugging protocol
- Used to determine whether a remote host is reachable
  - Thus generally speaking, ICMP is NOT used to exchange data between systems
- Other uses: inform about traffic overloads, obtain the network mask at boot time for diskless systems, synchronize clock
- Exchange control and error messages about the delivery of IP datagrams
  - Messages: Echo (request), Reply (response), Error
- RFC 792: http://www.ietf.org/rfc/rfc792.txt

Ping

- Utility to send `ICMP ECHO_REQUEST` packets to network hosts
  - More on what a packet is later
- Built in to almost all operating systems (e.g., Windows, Linux, Mac OS X)
- Documentation on Linux or Unix-based system: `man ping`
- Basic usage: `ping <host>`
  - Example: `ping google.com`
- What you cannot do with `ping`: check for open ports on a remote system

User Datagram Protocol (UDP)

- On Transport layer of OSI model
- Relies on IP to provide a connectionless, unreliable, best-effort datagram delivery service.
- In other words, may be dropped before reaching targets a.k.a., fast
- Delivery, integrity, non-duplication, ordering, and bandwidth is not guaranteed
- Unlike TCP/IP, no handshaking!
- No sequence numbers
- Usage: DNS, streaming videos, video games
- RFC 768: https://www.ietf.org/rfc/rfc768.txt

Ethernet

- On Data Link layer of OSI model
- A network protocol that controls how data is transmitted over a local area network (LAN)
- Addressing: Media Access Control (MAC) address
  - A unique identifier assigned to network interfaces (e.g., your wireless network hardware card) for communications at the data link layer of a network segment
  - 48 bits in the format XX:XX:XX:XX:XX:XX
  - Example: 09:45:FA:07:22:23

Address Resolution Protocol (ARP)

- On Data Link layer of OSI model
- The idea of ARP: get Ethernet address of host with IP address (very much like delivering mail to an office building)
  - ARP request message, think of it this way: "Hey who has this IP? If it's you, please respond and tell me your MAC address”
  - ARP reply message, think of it this way: "This is my MAC address and I have this IP address”
- Host A wants to know the hardware address associated with IP address of host B
- A broadcasts a special message to all the hosts on the same physical link
- Host B answers with a message containing its own link-level address
- A keeps the answer in its cache (20 minutes)
- To optimize traffic, when A sends its request, A includes its own IP address
- The receiver of the ARP request will cache the requester mapping
- RFC 826: https://www.ietf.org/rfc/rfc826.txt
- Reference: https://www.homenethowto.com/switching/arp-mac-ip/
- Tools: `arp`

Domain Name Systems (DNS)

- Analogy: telephone book for the Internet; mapping of IP addresses to domain names and vice versa
- On Application layer of OSI model
- The name space is hierarchically divided in domains
- Each domain is managed by a name server
  - Servers are responsible for mapping names in a zone
- Root servers are associated with the top of the hierarchy and dispatch queries to the appropriate domains
- A server that cannot answer a query directly forwards the query up in the hierarchy.
- The results are maintained in a local cache for a limited time (which can range from minutes to days).
- Queries can be recursive
- DNS uses mostly UDP and sometimes TCP for long queries and zone transfers between servers (port 53)
- Associated RFCs: https://en.wikipedia.org/wiki/Domain_Name_System#RFC_documents
- References:
  - https://www.verisign.com/en_US/website-presence/online/how-dns-works/index.xhtml
  - https://dyn.com/blog/dns-why-its-important-how-it-works/
- Tools: `dig`, `host`, `nslookup`

So far…

- …you have learned about the OSI model
- …you have learned about the TCP three way handshake
- …you have seen headers, network protocols, etc.
- There is a lot going on here…
- ***How can you comprehend all this tangibly? How can one visualize what's going on?***
- Next steps: packets, PCAPs, and Wireshark

Packet

- **Packet** - unit of data
- A data stream (e.g., video, a web page) is comprised of many packets
- In general, a packet contains the following information:
  - Source and destination IP addresses (in IP layer)
  - Source and destination port number (in TCP layer)
  - MAC address (in Data Link layer)
  - Time To Live (TTL; in IP layer)
  - Payload
- Thus, a packet contains implementations of all the protocol layers (including TCP, IP, application, data link)
  - Encapsulation model
  - Think of an onion

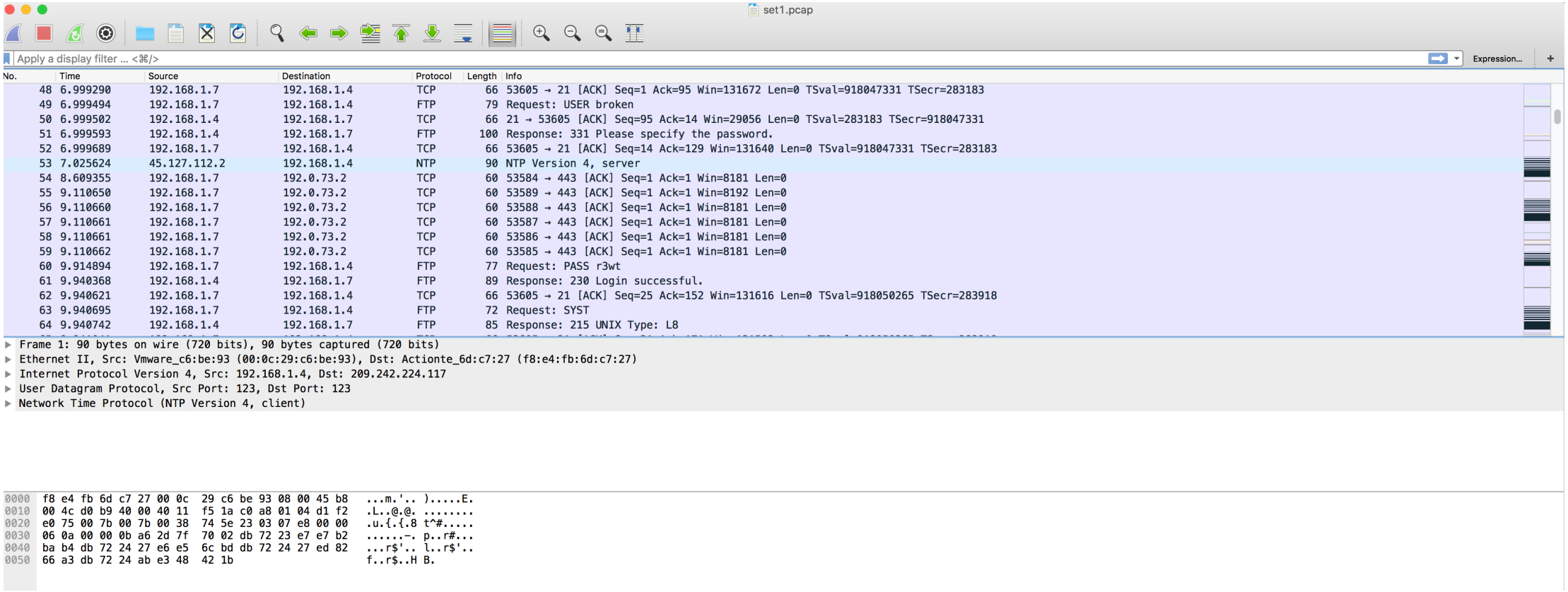
A .pcap File

- The common file extension for packet captures and is commonly used in many applications such as Wireshark, ettercap, tcpdump
- A 100 MB PCAP file contains tens of thousands of packets

Tool: Wireshark

- Graphical and extensive packet analyzer
- One of the most important tools in the field
- Very similar to `tcpdump`
- Open source and free
- Features include filtering, reconstructing conversations, reconstructing files based on packets
- https://www.wireshark.org/

Wireshark (continued)



Tufts UNIVERSITY | SCHOOL OF ENGINEERING Computer Science

Tool: tshark

- Dumps and analyzes network traffic
- Command-line-based Wireshark
- Installed with Wireshark
- The manual: `man tshark`
- Example, list the hosts in a PCAP file:
  - `tshark -r file.pcap -q -z hosts,ipv4`

Tool: tshark (continued)

```
$ tshark -r set3.pcap -q -z hosts,ipv4
# TShark hosts output
#
# Host data gathered from set3.pcap

50.22.4.220     api.south.kontagent.net
17.172.224.47   apple.com
199.59.148.20   api.twitter.com
52.10.76.66     external-nginx-api.prod.us-west2.twitch.tv
23.21.212.107   data-collector-linkedin-prod-1143471378.us-east-1.elb.amazonaws
68.67.129.117   ib.anycast.adnxs.com
54.235.163.76   elb051356-548148482.us-east-1.elb.amazonaws.com
17.167.193.235  gsp36-ssl.ls-apple.com.akadns.net
222.239.85.206  upload.inven.co.kr
17.172.232.166  4.courier-sandbox-push-apple.com.akadns.net
52.7.6.170      api.shopkeepapp.com
54.148.244.104  external-nginx-api.prod.us-west2.twitch.tv
58.251.139.219  imap.qq.com
17.172.232.190  4.courier-sandbox-push-apple.com.akadns.net
17.134.126.30   gsp-ssl.ls-apple.com.akadns.net
52.21.62.183    elb-ad-01-659338009.us-east-1.elb.amazonaws.com
108.168.211.132 api.south.kontagent.net
169.46.12.66    api.south.kontagent.net
108.168.211.135 api.south.kontagent.net
169.46.12.69    api.south.kontagent.net
199.59.149.230  twitter.com
54.183.107.128  aerios.cyngn.com
169.46.12.72    api.south.kontagent.net
104.25.56.25    cdn.inspectlet.com
```

Tool: tcpdump

- A packet analyzer that runs via command line
- To run: `sudo tcpdump -i <INTERFACE>`
- The manual: `man tcpdump`
- Cheat sheet via SANS Institute: https://www.sans.org/security-resources/tcpip.pdf
- Example: reading a PCAP file
  - `tcpdump -r file.pcap`
- Example: splitting a PCAP file into smaller ones (e.g., 10 MB)
  - `tcpdump -r old_file.pcap -w new_files -C 10`

Lab: Packet Sleuth

The Next Time: Attacking Networks

- Sniffing
- Network reconnaissance
- Denial of Service (DoS)
- Impersonation (spoofing)
- Hijacking (information access, delivery tampering)