Introduction to Cyber Security Week 1: Networking and Attacking Networks

Ming Chow (mchow@cs.tufts.edu)

Twitter: @0xmchow

Introduction

Welcome to the course!

What This Course Is NOT

- Hack all things
- Be an 31337 h4x0r (“leet speek” for “elite hacker”)
- Introduction to Cryptography
- Save the world
- The answer is “black or white”

What the Online Course Will NOT Cover

- Forensic and anti-forensics
- Social engineering
- x86, x64, ARM reverse engineering
- Privacy
- Policy and politics
- Legal issues
- Cyber warfare
- Cryptocurrency
- Security management

What This Course Is

- Controversial
- WARNING: A little about a lot
  - Why: Cyber Security is a very broad field as evident by list of topics that will not be covered in this course
- Provides exposure to tools that the attackers and defenders are using
- Understand tradeoffs
- Make you think like an attacker
- Be informed of the issues. Because most in Computer Science or in tech just do not know them.
  - Many developers don't even consider security in software.
  - You can't complete a Civil and Environmental Engineering degree without learning anything about health and safety but you *can* complete a Computer Science degree without learning anything about critical infrastructure, health, and safety.

Software and Hardware Requirements

- **Absolute Requirements**
  - A modern web browser (e.g., Firefox, Google Chrome, Chromium, Safari, Microsoft Edge)
  - A command line interface to run Unix/Linux commands
- **Strongly Recommended Requirements**
  - A computer with at least 40 GB of hard disk space free and 4 GB of RAM
  - Kali Linux and a Virtual Machine Hypervisor
  - Why is this strongly recommended and not mandatory? Telling students to have beefy computers especially to run virtual machines in order to take this course is sending the wrong message. Cyber Security must be accessible as possible.

The Basic Skill Necessary: Exposure, Experience, and Comfort with Command Line Interface (CLI) / Terminal

- Versatility, productivity, accessibility, scripting
- Not everything can be accomplished by fancy graphics and GUIs (sometimes constrained to certain features too)
- Many systems do not use a windows manager or have a graphical desktop interface (e.g., servers)
- Graphical interface (especially on servers) => more software requirements, more overhead, more bloat, more vulnerabilities
- Many security tools are command line based
- Remote execution of commands –for good and bad.
- *This is one of the first labs*

Why is it Called Cyber Security?

- Should it be called “Information Security”? No. Because there’s a lot more than information we are concerned with. Case-in-point: hardware and the “Internet of Things”
- Should it be called “Computer Security”? No. Technology is not everything. Sometimes the best solution (or attack) is not technical. Examples: social engineering, lock picking, impersonation, phishing.

Running List of Thoughts on What Cyber Security Is (from former students)

- Make sure your data is protected
- Want data to be private
- Want to preserve the integrity of the data
- Preservation of the status quo
- Access control
- Protection of hardware and networks => physical security
- User security (protecting themselves)
- Knowledge of what the attackers / adversaries are doing
- Maintaining good governance
- Punishment of the attackers, strike back?
- Containment, stop the bleeding
- Resilience, separation of concerns, redundancy
- …and the list goes on…

By Definition, What is Cyber Security?

- The “CIA Triad”
  - Confidentiality
  - Integrity
  - Availability

Convention Used in Notes

- **bold** – keyword or definition
- *italic* – to be discussed in more details later
- `code via Courier New font` – code or command

Basic Definitions

- Why is vocabulary important?
  - There is a problem with vocabulary in this field. Many words have different context and meaning to different groups (e.g., the policy folks in the field).
  - Many words are also misused by media.
- **Event** - Could be anything
- **Incident** - A malicious event
- **Bug** - An error that exists in the implementation-level (i.e. only exist in source code); very correctable
- **Flaw** - An error at a much deeper level, particularly in the design, and likely in the code level; can be very difficult and costly to correct
- **Hacker** - A creative programmer; a positive connotation
- **Cracker** - The bad guy, the attacker, what media coins "hacker" (the negative connotation). We'll use **attacker** in this class.

Basic Definitions (continued)

- **Black hat** - An attacker with malicious intents
- **White hat** - An attacker with good intents (i.e., the white knight)
- **Gray hat** - An attacker with good and bad intents
- **Script kiddie** or **skiddie** - Nuisance; not going away any time soon; 1337 (i.e., elite) wannabes; use scripts and exploits written by others and do not understand how they really work; always a lamer
- **Vulnerability** - A security bug (thanks Giovanni Vigna); a weakness in a system that can potentially be exploited by an attacker
- **Exploiting** or **exploitation** - The act of taking advantage of a vulnerability
- **Exploit** - Software program that performs the exploiting
- **Risk** - The likelihood that an attacker will take advantage of that vulnerability
- **Threat** - The likelihood that an incident will happen
- **pwn3ed** - Owned; successful exploitation; computer system completely compromised
- **Zero day** - an undisclosed vulnerability that attackers can take advantage of. A zero-day attack happens once that flaw, or software/hardware vulnerability, is exploited and attackers release malware before a developer has an opportunity to create a patch to fix the vulnerability—hence "zero-day." https://www.fireeye.com/current-threats/what-is-a-zero-day-exploit.html

Violating the CIA Triad: If You Were An Attacker, What Are Your Goals?

- Preventing enemies from communicating over network
- Steal information for attacker's benefit and get away with it
- Disruption of business, daily life, day-to-day operations
- Inserting information that "shouldn't be there”
- Destroy information, resources
- Gain access to a system and maintain access to system for a long time
- Monitoring people what they are doing (e.g., webcams)
- Challenging adversaries, pinpointing weaknesses
- For fun and profit (e.g., the black market)
- Spread propaganda
- Building a blueprint of weaknesses –and keep it for future reference

Why the Rash of Incidents: Behind the Breaches and Attacks (thoughts from former students)

- Trust relationships, lots of implicit trust
- Data is very valuable
- Convenient to put everything online; sharing
- Lack of education; people are not being informed
- No barriers to entry
- Lack of deterrence
- Software vulnerabilities
- Misconfigurations
- Human elements, social engineering
- Scapegoating

The Trinity of Trouble: Why the Problem is Growing (by Gary McGraw)

- Connectivity
- Extensibility
- Complexity
- Read: https://freedom-to-tinker.com/2006/02/15/software-security-trinity-trouble/

Basic Networking

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?

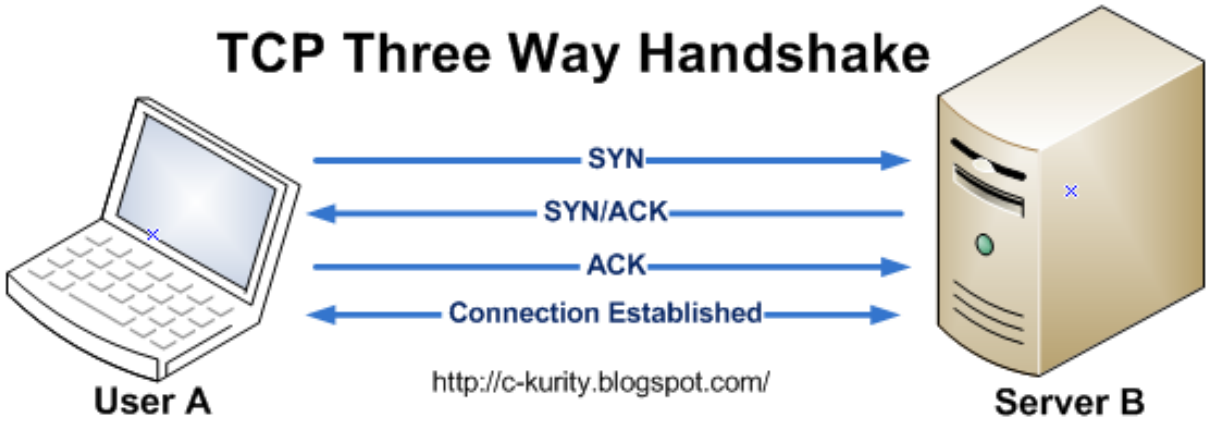
- Two or more computers talking to each other
- Basic definitions:
  - **Client** - A program running on your computer
    - Web browser - a client application that displays web pages (e.g., Chrome, Firefox, Microsoft Internet Explorer, Safari, Opera, lynx)
  - **Server** - A computer running web server 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)

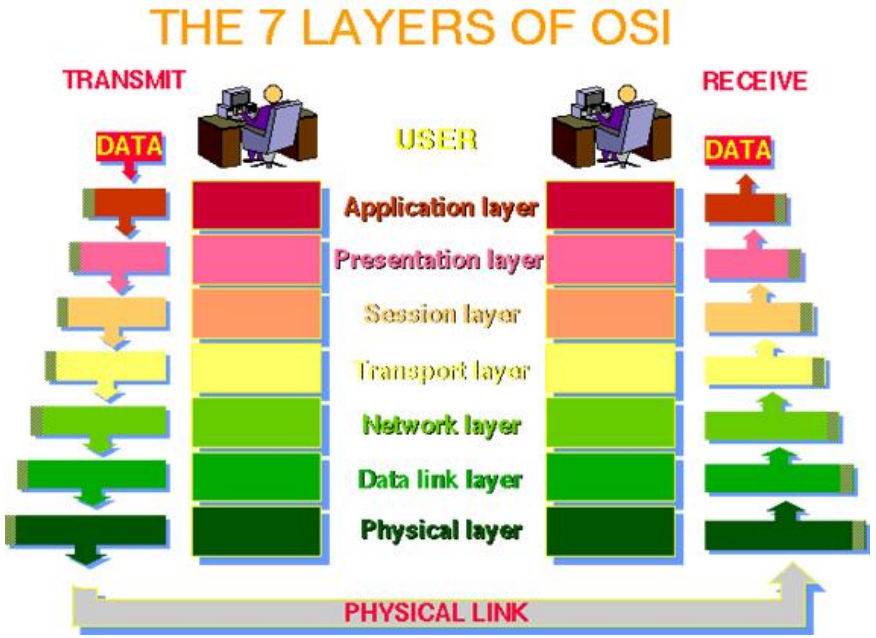
Abridged Analogy Describing How Two Computers Talk to Each Other (continued)

- 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.
- Source: https://www.quora.com/Can-you-explain-OSI-layers-and-TCP-IP-in-laymans-terms
- *We will focus on the Network, Transport, and Application layers extensively*

Application Layer

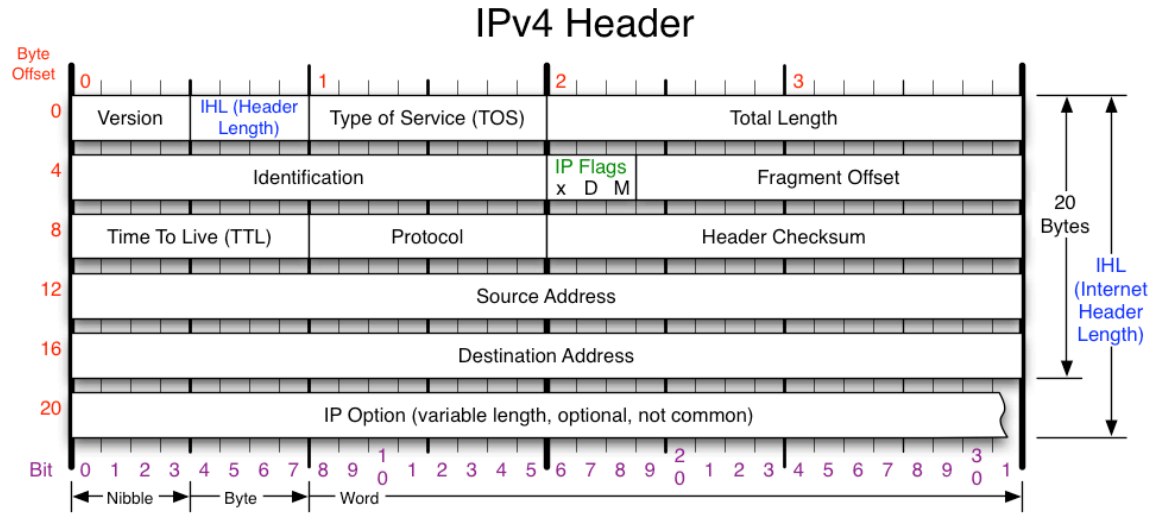
- The famous and insecure ones by default, data all unencrypted:
  - DNS – Domain Name Server (DNS)
    - Port 53
  - IMAP (Internet Message Access Protocol)
    - Email
    - Port 143
  - FTP (File Transfer Protocol)
    - File transfer
    - Port 21
  - HTTP (Hypertext Transfer Protocol)
    - The foundation of data communication for the World Wide Web
    - Port 80
  - Telnet
    - Protocol that allows you to connect to remote computers
    - Port 21
  - 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

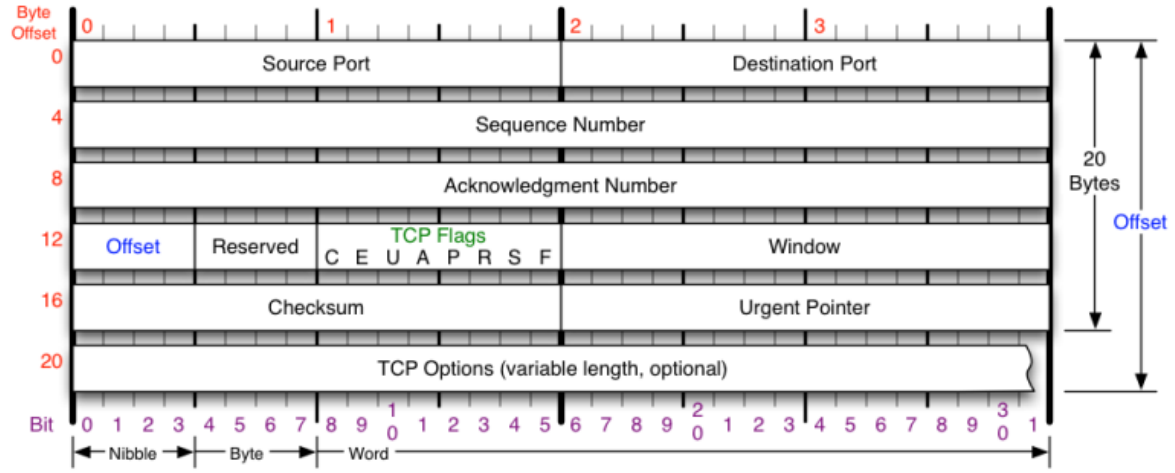
Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

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`

Putting It All Together: Attacking Networks

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

Network Sniffing

Network Sniffing

- Look at network traffic
- Also known as analyzing *packets*
- Most of the traffic on a network is still unencrypted, plaintext ("in the clear")
- Things you can do with network sniffing:
  - Troubleshoot networking issues
  - Record communications (e.g., email, voice, chat)
  - Catch usernames and passwords, personal information, and other sensitive information

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

Two Types of Networks

1. **Unswitched** - packets flow through all devices on network but you look at only the packets addressed to you.....
   - - Welp... https://superuser.com/questions/191191/where-can-i-find-an-unswitched-ethernet-hub
2. **Switched** - packets flow through specific devices on network; most common today

Getting Started: What You Need

- A computer with wired or wireless networking. Any platform is acceptable
- You can also choose a Linux distro live-CD aimed at penetration testing such as Kali to get up-and-running quickly
- Administrative access on computer is required!
- Access to a ***span port***, ***LAN tap***, or a ***network hub***

Span Port

- Also known as port mirroring
- All the packets on one switch port (or an entire virtual LAN) to another port

LAN Tap

- Typically small devices
- Used to monitor Ethernet communications
- You can buy one at https://greatscottgadgets.com/throwingstar/

Network Hub

- Device for connecting multiple Ethernet devices to a single network segment
- Divides bandwidth across all the ports

Getting Started: First Things First

- Step 1: Put your network card to *promiscuous mode*
  - **Promiscuous mode** - look at all packets regardless of destination address
  - Analogy: look inside everyone's mailbox on your street
- Step 2: Disable the use of the Address Resolution Protocol (ARP)
- For Unix/Linux/Macs: `sudo ifconfig -i <INTERFACE> promisc -arp`
- An **interface** is the network hardware you want to use for sniffing. To see list of interfaces, run `ifconfig` (or as of recent, `ip`)
  - `eth0` is typically the interface for wired Ethernet
  - `wlan0` is an interface for wireless networking, en0 on Macs

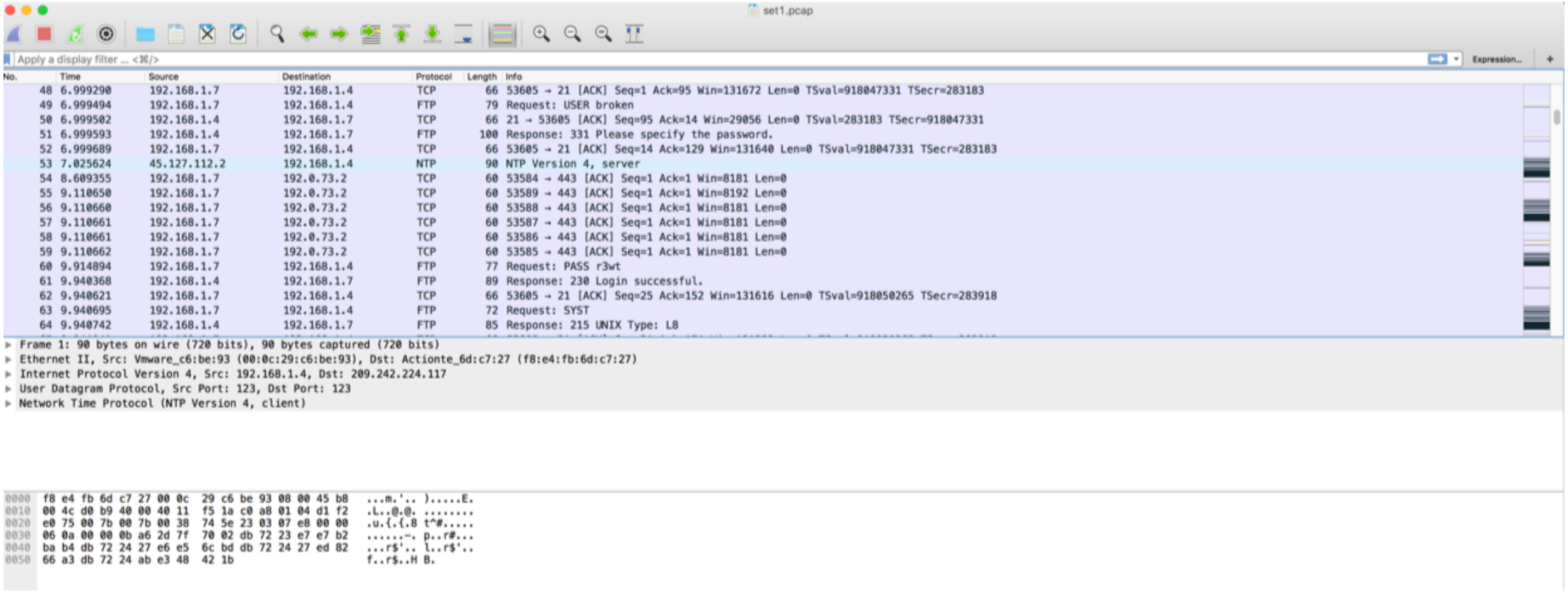
Tool: tcpdump

- 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`

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)



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: Ettercap

- Graphical and command-line based
- Is not intended for network traffic analysis but has capabilities for:
  - Capturing passwords
  - Conducting man-in-the-middle (eavesdropping) attacks
  - Hijacking sessions
- The manual: `man ettercap`
- https://ettercap.github.io/ettercap/
- Example: to list plaintext passwords captured in a PCAP file
  - `ettercap -T -r set3.pcap | grep "PASS:"`

Tool: dsniff

- Suite of networking sniffing tools including
  - `dsniff` - password sniffer
  - `webspy` - intercepts URLs entered
  - `mailsnarf` - intercepts POP or SMTP-based mail
- Written by Dug Song in 2000
- To run: `sudo dsniff -i <INTERFACE>`
- Warning: can be flaky at times (e.g., can't detect username:password pairs from an FTP PCAP); no longer suppor

Tool: ngrep

- Recall `grep` for searching
- Network `grep`
- Currently recognizes IPv4/6, TCP, UDP, ICMPv4/6, IGMP, etc.
- http://ngrep.sourceforge.net/
- Example, search for text strings in a PCAP file:
  - `ngrep -q -I file.pcap | grep -i github`

Tool: Bettercap

- https://bettercap.org/
- Written by Simone Margaritelli (@evilsocket)
- Ruby-based
- Very similar to Ettercap, a better Ettercap

Tool: Bettercap (continued)

Lab: Packet Sleuth

Prevent Sniffing?

- Use encryption and encrypted network protocols
  - Use HTTPS instead of HTTP
  - Use SSH instead of RSH or Telnet
  - Use SCP instead of FTP
  - Use IMAP or POP3 over SSL
- Use a Virtual Private Network (VPN)
- Use switched network.....?
  - NO!

Sniffing a Switched Network

- ARP spoofing (a.k.a., ARP poisoning)
- The idea is very simple: you pretend to be the router and thus all the traffic goes to you (your computer). In other words, Man-in-the-Middle (MitM)
- More background: https://www.irongeek.com/i.php?page=security/arpspoof
- More: https://www.bettercap.org/docs/intro.html
- More: https://www.veracode.com/security/arp-spoofing
- ARP spoofing via Bettercap:
  - `sudo bettercap -S ARP`

Methods of Preventing Sniffing on Switched Networks

- Packet filtering
- Avoid trust relationships
- Tools:
  - anti-arpspoof
  - ArpON
  - Antidote
  - Arpwatch

Network Scanning

Network Scanning

- Why? Network reconnaissance. Warfare 101
- What devices and computers are up?
- What ports are open on a computer?
- What services are running?
- Determine possible vulnerabilities?
- Still extremely relevant today
- Think poking holes, "ask questions”
  - Poking holes: finding interesting and unwanted stuff on networks

Method: Ping Sweep

- Tool: `fping` (circa 1992)
  - http://fping.sourceforge.net/
  - Can be used in scripts
  - Can use a range of IP addresses
- Problems with ping:
  - Recall: you cannot check for open ports on a remote system using `ping`
  - Many systems have turned off responding to `ping`

Tool: Netcat

- The TCP/IP Swiss-Army Knife
- Written by Hobbit
- Built into most Linux and Unix distributions
- Manual: `man nc`
- Cheat sheet via SANS Institute: https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf
- Example: port scan an IP address (via SANS Institute cheat sheet):
  - `nc -v -n -z -w1 [TargetIPaddr] [start_port]-[end_port]`
  - Example: `nc -v -n -z -w1 192.168.1.1 1-10000`

Tool: Nmap

- Network exploration tool and security / port scanner
- https://nmap.org/
- Written by Gordon "Fyodor Vaskovich" Lyon
- One of the most important tools in the field
- Very well documented
  - Official book and documentation: https://nmap.org/book/man-port-scanning-techniques.html
  - http://tools.kali.org/information-gathering/nmap
- Example: Scan in verbose mode ***(-v)***, enable operating system detection, version detection, script scanning, and traceroute ***(-A)***, with version detection ***(-sV)*** against the target IP ***(192.168.1.1)***:
  - `nmap -v -A -sV 192.168.1.1`
- More Nmap examples: https://highon.coffee/docs/nmap/

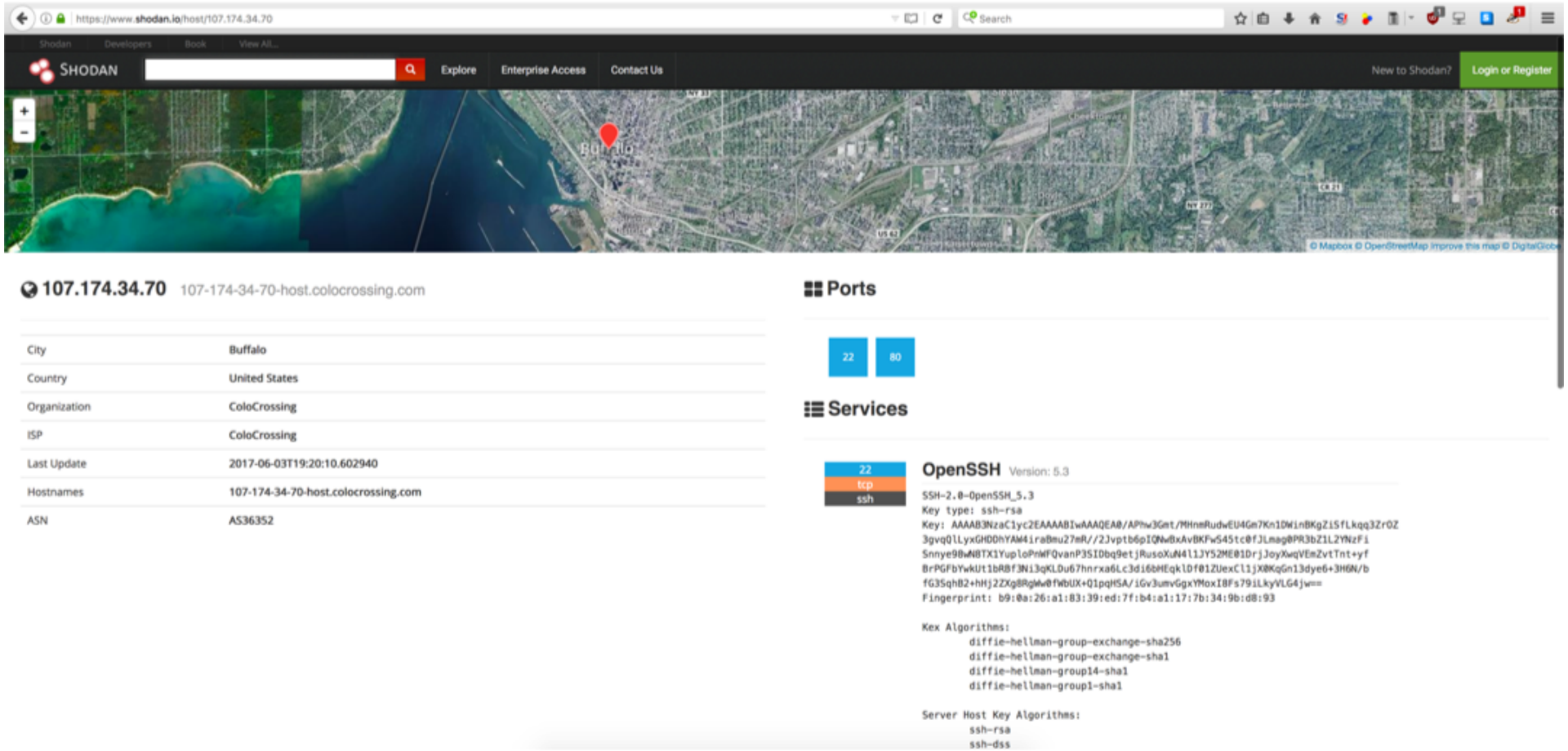
Tool: Nmap (continued)

```
$ nmap 192.168.1.66

Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-05 18:17 EDT
Nmap scan report for 192.168.1.66
Host is up (0.0083s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql
5001/tcp open  commplex-link
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

Tool: SHODAN

- https://www.shodan.io/
- Website, search engine
- Written by John Matherly in 2009
- Free upgrade if you sign up using academic email address: https://twitter.com/shodanhq/status/826703889550438403?lang=en
- Cheat sheet via SANS Institute: https://pen-testing.sans.org/blog/2015/12/08/effective-shodan-searches/
- Example search to get details on an IP address:
  - https://www.shodan.io/host/212.187.208.158
  - Generic form: https://www.shodan.io/host/<IP address>

Tool: SHODAN (continued)



What Could Possibly Go Wrong With Using Nmap?

- You will be detected by Intrusion Detection Systems (IDS), flagged, noticed, logged
- By default, using Nmap with no flags (e.g., `nmap <IP address>`) will perform a TCP SYN scan which many modern firewalls and IDSes will detect.
  - https://security.stackexchange.com/questions/19576/what-scanning-tools-are-unlikely-to-set-off-network-ids
- You want to be *stealthy*!

Stealthy Nmap Scans

- On page 65 of RFC 793 for TCP: “If the state is CLOSED (i.e., TCB does not exist) then all data in the incoming segment is discarded. An incoming segment containing a RST is discarded. An incoming segment not containing a RST causes a RST to be sent in response. The acknowledgment and sequence field values are selected to make the reset sequence acceptable to the TCP that sent the offending segment.”
- In other words, if ports are closed and you send "junk" to it, RST packet will be sent!

Stealthy Nmap Scans (continued)

- Three stealthy scans using Nmap:
  1. FIN scan: `sudo nmap -sF ...` [only TCP FIN flag in packet]
  2. NULL scan: `sudo nmap -sN ...` [No flags in packet]
  3. Christmas Tree (XMAS) scan: `sudo nmap -sX ...` [FIN, PSH, URG flags in packet]
- Documentation under "-sN; -sF; -sX (TCP NULL, FIN, and Xmas scans) " https://nmap.org/book/man-port-scanning-techniques.html
  - "The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers. Another advantage is that these scan types are a little more stealthy than even a SYN scan. Don't count on this though—most modern IDS products can be configured to detect them. "

Defending Against Scanners

- Close services on a computer that are not necessary
- Packet filtering
- Firewalls?
  - Well, there are numerous firewall evasion techniques in Nmap
  - Documentation: https://nmap.org/book/man-bypass-firewalls-ids.html

Lab: Scanning and Reconnaissance

Decoy Scanning with Nmap

- The idea: blame someone else
- Also known as a cloak scan
- “which makes it appear to the remote host that the host(s) you specify as decoys are scanning the target network too. Thus their IDS might report 5–10 port scans from unique IP addresses, but they won't know which IP was scanning them and which were innocent decoys. While this can be defeated through router path tracing, response-dropping, and other active mechanisms, it is generally an effective technique for hiding your IP address.” (https://nmap.org/book/man-bypass-firewalls-ids.html)
- `sudo nmap -D <IP of decoy 1>, [<IP of decoy 2>] …`
- IMPORTANT! Must use real + alive IP address, else accidental *SYN flood…*

Distributed Denial of Service (DDoS) Attacks

Significance

- The idea: to make a resource unavailable (the “A” in the CIA Triad) using many remote computing devices. That is, overwhelm or flood a target (e.g., with so much network traffic)
- Imagine if an important service you use like Gmail, Google, Netflix, Twitter, GitHub is down

Mirai and the Dyn Cyber Attack in October 2016

- Terabit scale Distributed Denial of Service (DDoS) attacks from September 2016 to late 2016
- How: using thousands of infected devices, mostly cameras. Devices infected via weak username:password hardcoded on device (e.g., root:root, admin:admin)
- Results: took down Brian Kreb's blog in September 2016; GitHub, Twitter, Netflix, and many major services were affected via DDoS on Dyn DNS in October 2016 (i.e., "all eggs in one basket")
- Source code of Mirai botnet: https://github.com/jgamblin/Mirai-Source-Code
- Rob Graham's presentation on "Mirai and IoT Botnet Analysis" at RSA Conference 2017: https://vinceinthebay.files.wordpress.com/2017/02/rsac-slides-hta-w10-mirai-1.pdf

Definitions

- **Zombie** – an infected and compromised machine or computing device
- **Botnet** – a network of infected machines; can be used to perform Distributed Denial of Service attacks
- **Bot herder** or **bot master** – attacker(s) who controller a botnet
- **Command and Control (C&C)** – infrastructure (e.g., servers, software) to control malware and botnet

SYN Flood

- Recall the TCP/IP “three way handshake”
- Imagine if many people crank call you: you pick up the phone, say “hello” but no answer. Repeat.
- The idea: exhaust states in the TCP/IP stack
- Attacker sends SYN packets with a spoofed source address, the victim, (that goes nowhere)
- Victim sends SYN/ACK packet but attacker stays slient
- Half-open connections must time out which may take a while
- Alas, good SYN packets will not be able to go through
- References:
  - https://www.cert.org/historical/advisories/CA-1996-21.cfm
  - RFC 4987: https://tools.ietf.org/html/rfc4987
  - https://www.juniper.net/documentation/en_US/junos12.1x44/topics/concept/denial-of-service-network-syn-flood-attack-understanding.html

Defending Against SYN Flood

- Reduce the SYN-received timeout
- Drop half-open connections when the limit has been reached and new requests for connection arrive
- Limit the number of half-open connections from a specific source
- Increase the length of the half-open connection queue
- Use SYN cookies; they use special algorithm for determining the initial sequence number of the server
  - Read: https://cr.yp.to/syncookies.html

Teardrop

- Affects older operating systems including Windows 3.1x, Windows 95, Windows NT, and versions of the Linux kernel prior to 2.1.63
- The idea: "involves sending fragmented packets to a target machine. Since the machine receiving such packets cannot reassemble them due to a bug in TCP/IP fragmentation reassembly, the packets overlap one another, crashing the target network device." https://security.radware.com/ddos-knowledge-center/ddospedia/teardrop-attack/
- Recall RFC 791 (IP), the IP packet fields in question: Fragment Offset, Flag (namely "Don't fragment" and "More fragments")
- Result: "Since the machine receiving such packets cannot reassemble them due to a bug in TCP/IP fragmentation reassembly, the packets overlap one another, crashing the target network device."
- Reference: https://www.juniper.net/techpubs/software/junos-es/junos-es92/junos-es-swconfig-security/understanding-teardrop-attacks.html

Ping of Death

- The idea: violate the IP contract
- In RFC 791, the maximum size of an IP packet is 65,535 bytes --including the packet header, which is typically 20 bytes long. - An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes long. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes (65,535 - 20 - 8 = 65,507)
- Result: "However, many ping implementations allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions such as denial of service (DoS), crashing, freezing, and rebooting."
- Reference: https://www.juniper.net/documentation/en_US/junos/topics/concept/denial-of-service-os-ping-of-death-attack-understanding.html

ICMP Flood Attack

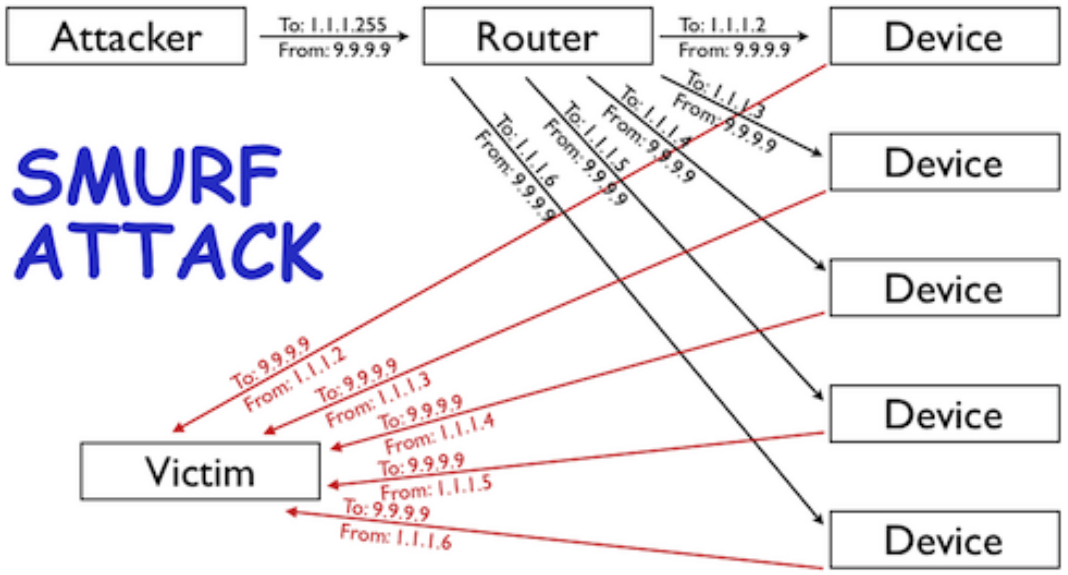
- Overload victim with a huge number of ICMP echo requests with spoofed source IP addresses

UDP Flood Attack

- Same idea of ICMP flood attack but using UDP packets

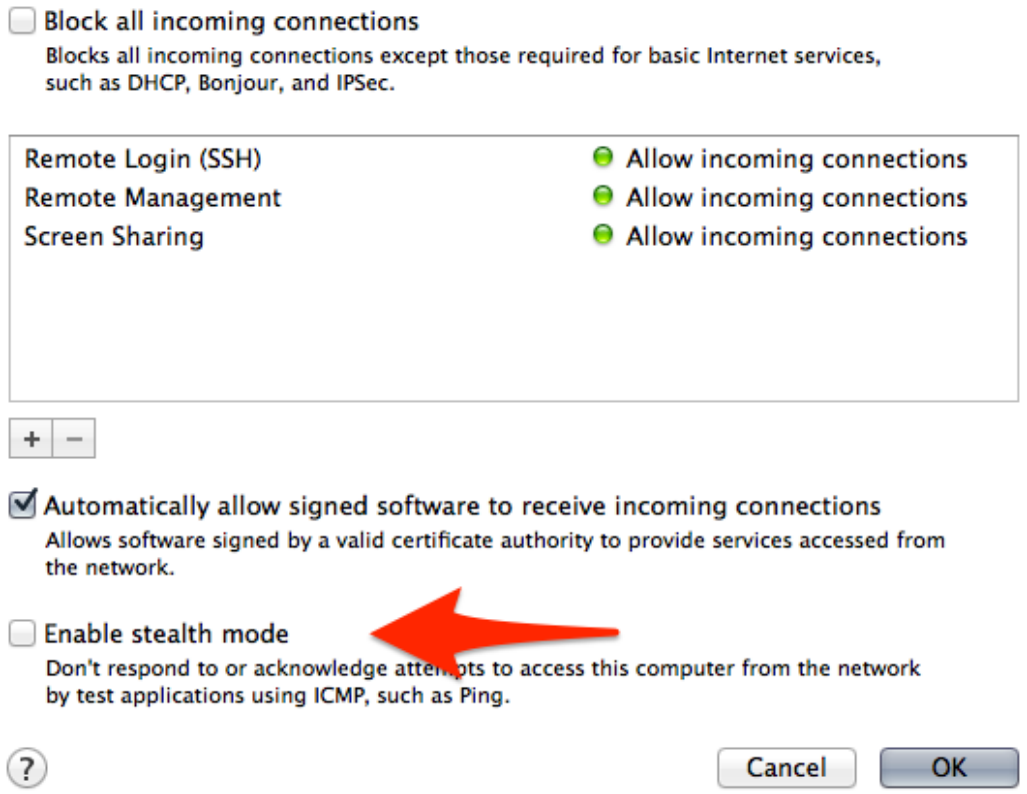
Smurf Attack

- Old, circa 1990s
- Significance: ***amplification***
  1. Create a network packet setting the source IP address as the victim or target (i.e., spoofing), and a destination IP address (some machine)
  2. Inside the packet is an ICMP ping message, asking destination that receive the packet to send back a reply –to victim
  3. The replies are sent to the victim, alas, overwhelming it
- Image source: https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/
- Reference: https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html



Defending Against ICMP Flood and Smurf Attacks

- Configure host to not respond to ICMP requests or broadcasts
- Image source: https://apple.stackexchange.com/questions/99996/which-setting-in-osx-could-block-ping-localhost



DNS Amplification

- Recall: DNS server port number 53
- The idea: "relies on the use of publically accessible open DNS servers to overwhelm a victim system with DNS response traffic."
- References:
  - https://www.us-cert.gov/ncas/alerts/TA13-088A
  - https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/
- Case study and recall Mirai: Brian Krebs http://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

One Last Thing

- How easy it is to spoof packets?
- Python's Scapy: allow extensive packet manipulation
- Example, to make a DNS query: https://gist.github.com/thepacketgeek/6928674
- Lab for next week