# Introduction to Security Attacking Networks

Ming Chow (mchow@cs.tufts.edu)

Twitter: @0xmchow

Tufts UNIVERSITY | SCHOOL OF ENGINEERING Computer Science

# Motivation

- You may be wondering how the PCAPs for the Packet Sleuth lab were obtained, especially the one from arguably the world's most dangerous network.
  - Answer: network sniffing
- What other activities can you do with packets?

# Network Sniffing

Tufts UNIVERSITY | SCHOOL OF ENGINEERING
Computer Science

# Network Sniffing

- Look at network traffic
- 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

# 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

# Two Types of Networks

1. **Unswitched** - packets flow through all devices on network but you look at only the packets addressed to you.....
   - This is trivial to do: set your network interface to promiscuous mode and open a packet analyzer to see all the traffic, but unswitched hubs are rare nowadays 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

# 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: Bettercap

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

# Tool: Bettercap (continued)

# 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.veracode.com/security/arp-spoofing

# Methods of Preventing Sniffing on Switched Networks

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

# Network Scanning

Tufts UNIVERSITY | SCHOOL OF ENGINEERING
Computer Science

# 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/

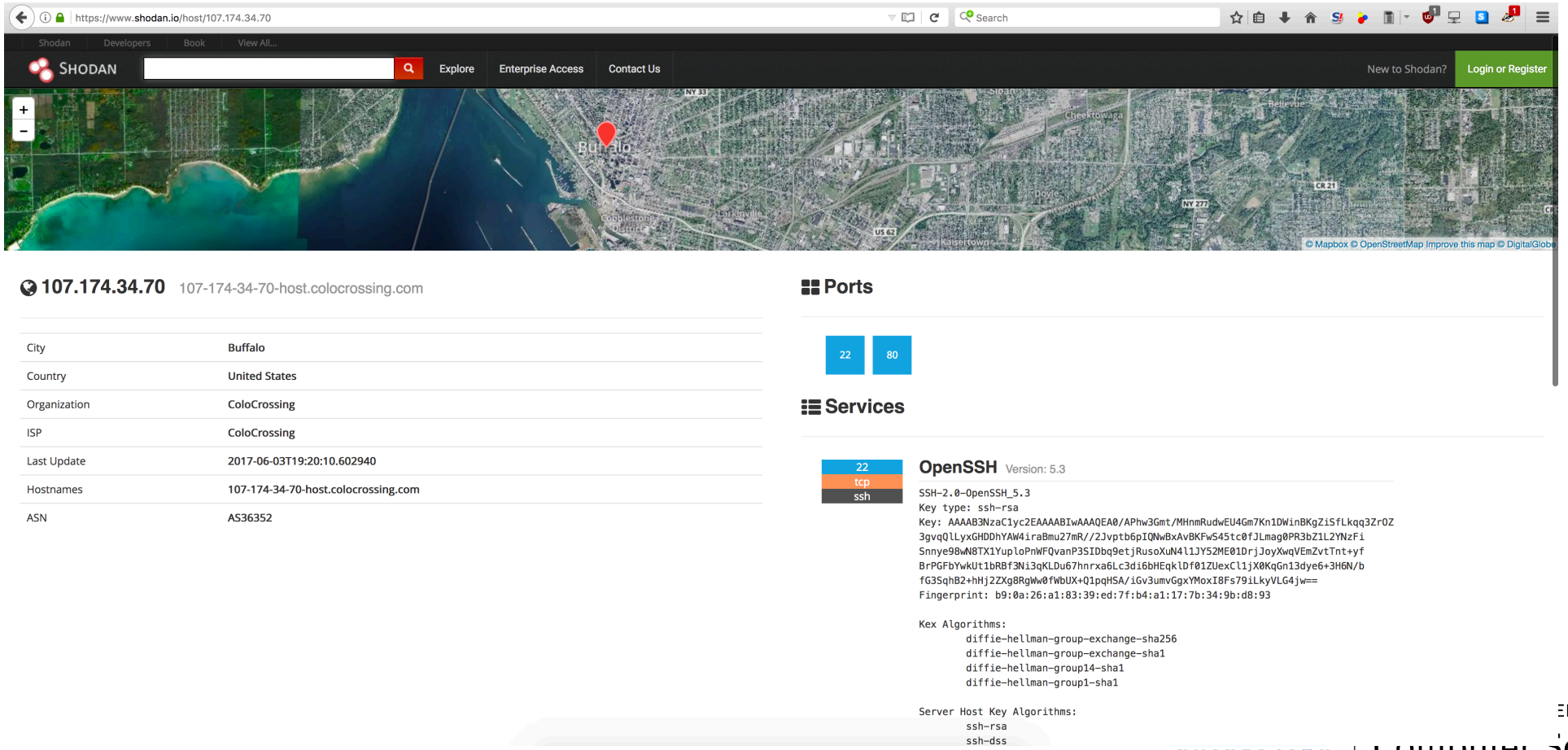
# Tool: Nmap (continued)

```
$ 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

Tufts UNIVERSITY | SCHOOL OF ENGINEERING Computer Science

# 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

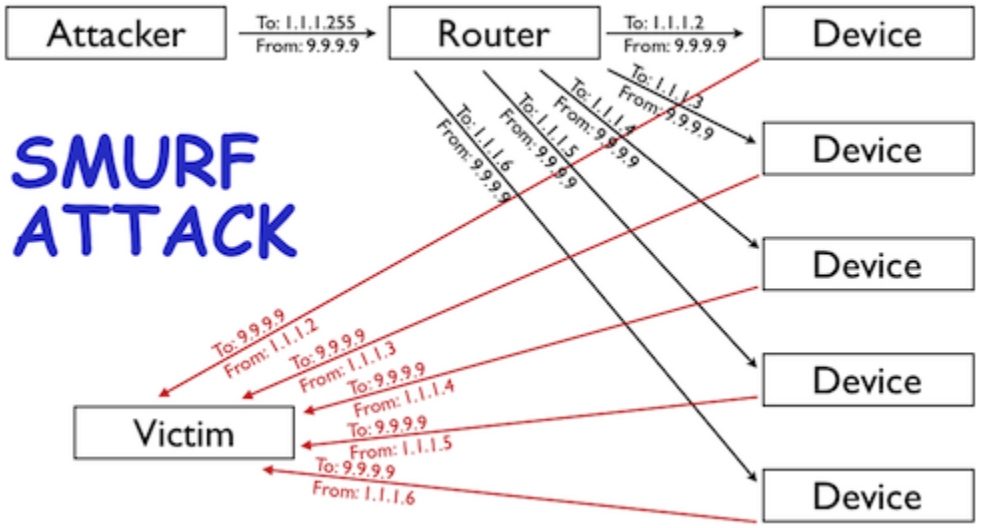
- Overload victim with a huge number of ICMP echo requests with spoofed source IP addresses

# UDP Flood Attack

- Same idea of ICMP flood attack but using UDP packets

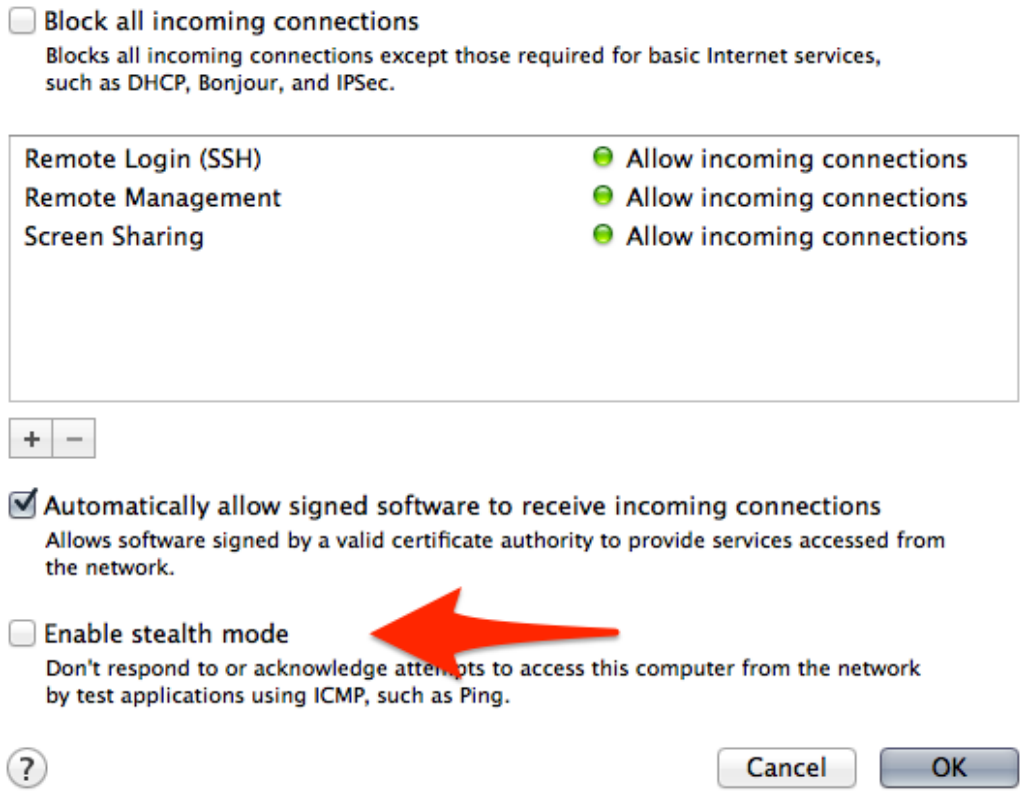
# Smurf Attack

- 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