CSE508 Network Security

9/7/2017

Lower Layers

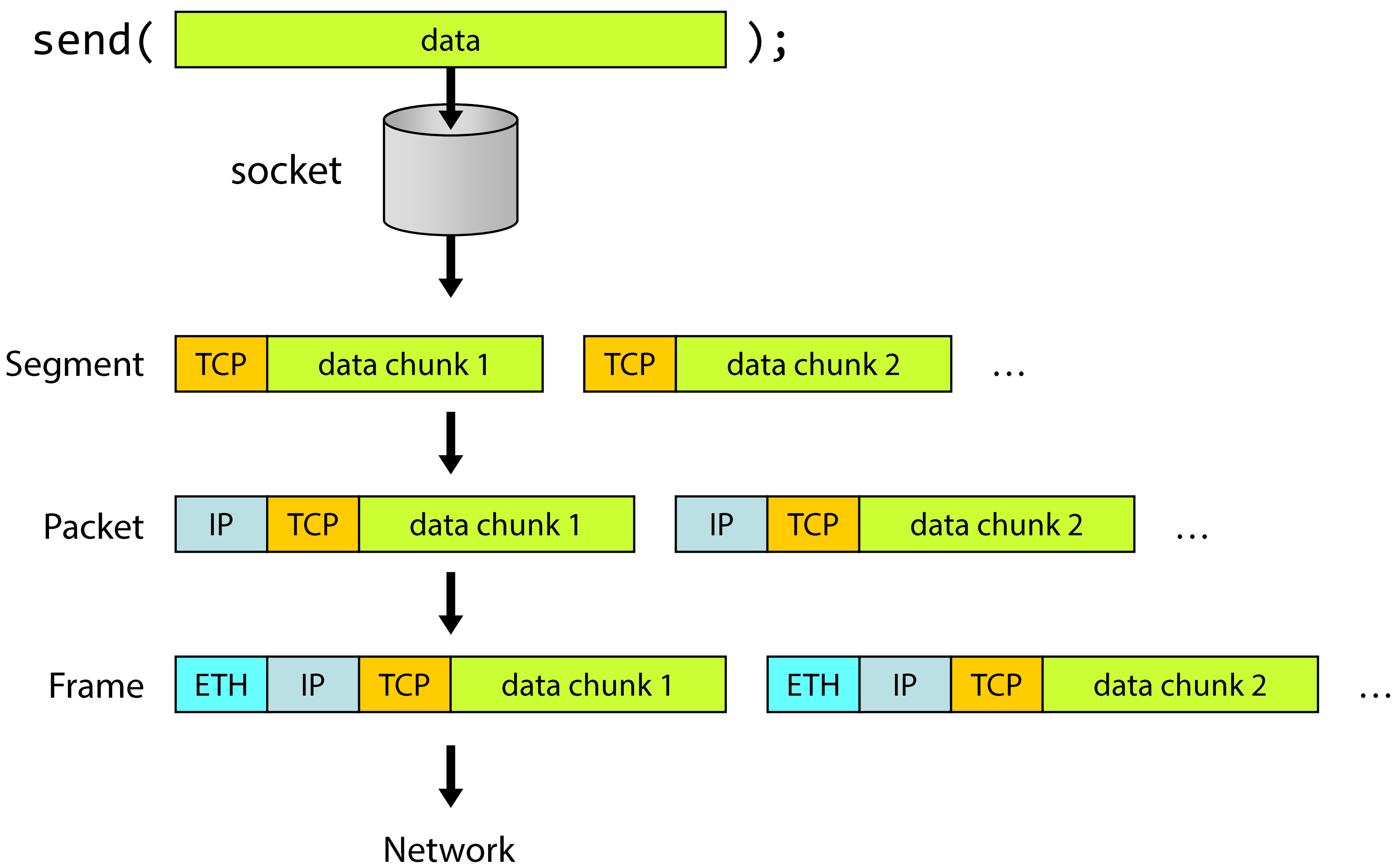
Michalis Polychronakis

Stony Brook University

Basic Internet Protocols (OSI Model vs. Reality)

				Deliver to:	Based on:
L7	*Application*	End-to-End	**HTTP**, BGP, DHCP, DNS, SPDY, SMTP, FTP, SMTP, IMAP, SSH, SSL/TLS, LDAP, NTP, RTP, SNMP, TFTP, …		
L6	*Presentation*				
L5	*Session*				
L4	*Transport*		**TCP**, UDP, SCTP, …	Dst. application	Port
L3	*Network*		**IP**, ICMP, IPsec, …	Dst. machine	IP
L2	*Data Link*		**Eth**, 802.11, ARP, …	Next hop	MAC
L1	*Physical*			Wire/air/pigeon	NIC

Streams vs. Packets



Active vs. Passive Attacks

Passive: the attacker eavesdrops but does not modify the message stream in any way

Traffic snooping, wiretapping, passive reconnaissance, listening for unsolicited/broadcast traffic, traffic analysis, …

Active: the attacker may transmit messages, replay old messages, modify messages in transit, or drop selected messages from the wire

Spoofing, session replay, data injection/manipulation (man-in-the-middle), DoS, malicious requests/responses, …

Physical Layer Attacks

Network eavesdropping

- NIC in promiscuous mode captures all traffic

Wiretapping (wire, optical fiber)

- Not needed for WiFi networks!

Wirecutting

Jamming

Electronic emanations/side channels

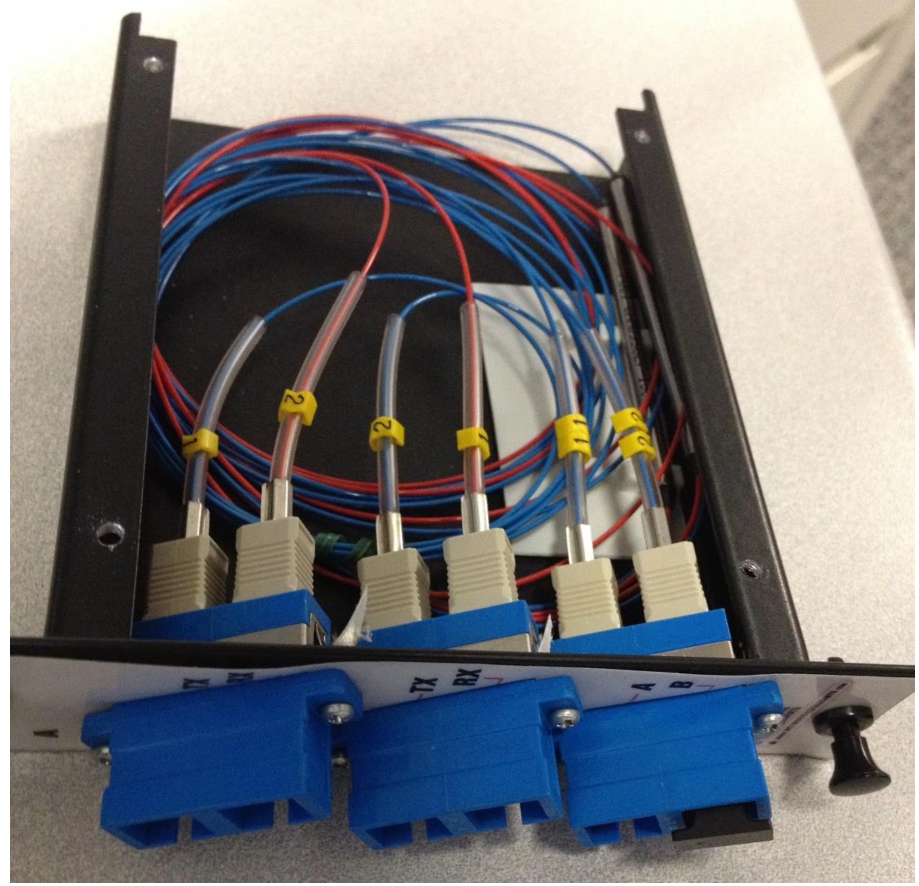
Tracking

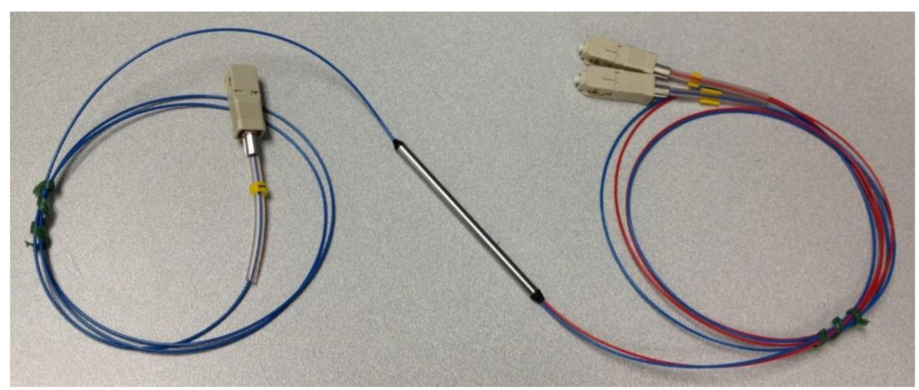
- Device fingerprinting
- Location tracking (cellular, WiFi)
- Many techniques of varying precision: trilateration/triangulation, nearest sensor, received signal strength, …

Network Taps

Up to 100Mbit/s can be completely passive

Most high-end switches can also mirror traffic





Ethernet

Most commonly used data link layer protocol for LANs

Communication based on *frames*

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|    48 bits                    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Destination Address        |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    48 bits                    |
|                                    Source Address             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (16 bits)        |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                    Payload (46-1500 bytes)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    32 bits        CRC                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Link Layer Attacks

Eavesdropping

WiFi: shared medium → *trivial*

Hub: broadcasts packets to all ports → *trivial*

Switch: learns which device is connected to which port and forwards packets only to the appropriate port → *still possible!*
ARP cache poisoning, CAM table exhaustion

Spoofing

Impersonate another machine and receive its traffic

Bypass address-based authentication

Change MAC address to get 30’ more of free WiFi

Hide the device’s vendor (first three bytes of MAC address)

DoS: flooding, deauth (WiFi), … (future lecture)

Rogue access point

Address Resolution Protocol (ARP)

Allows mapping of IP addresses to physical addresses

A new machine joins a LAN; How can it find the MAC addresses of a neighbor machine?

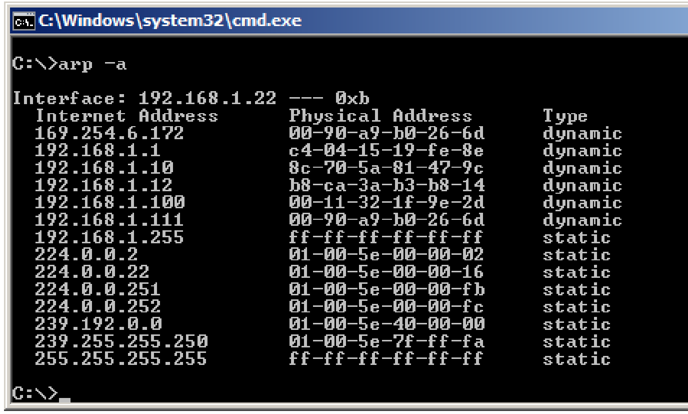
- **ARP request (broadcast):** *Who has IP 192.168.0.1?*
- **ARP reply by 192.168.0.1:** *Hey, here I am, this is my MAC address*

Each host maintains a local ARP cache

- Send request only if local table lookup fails

ARP announcements (*gratuitous ARP*)

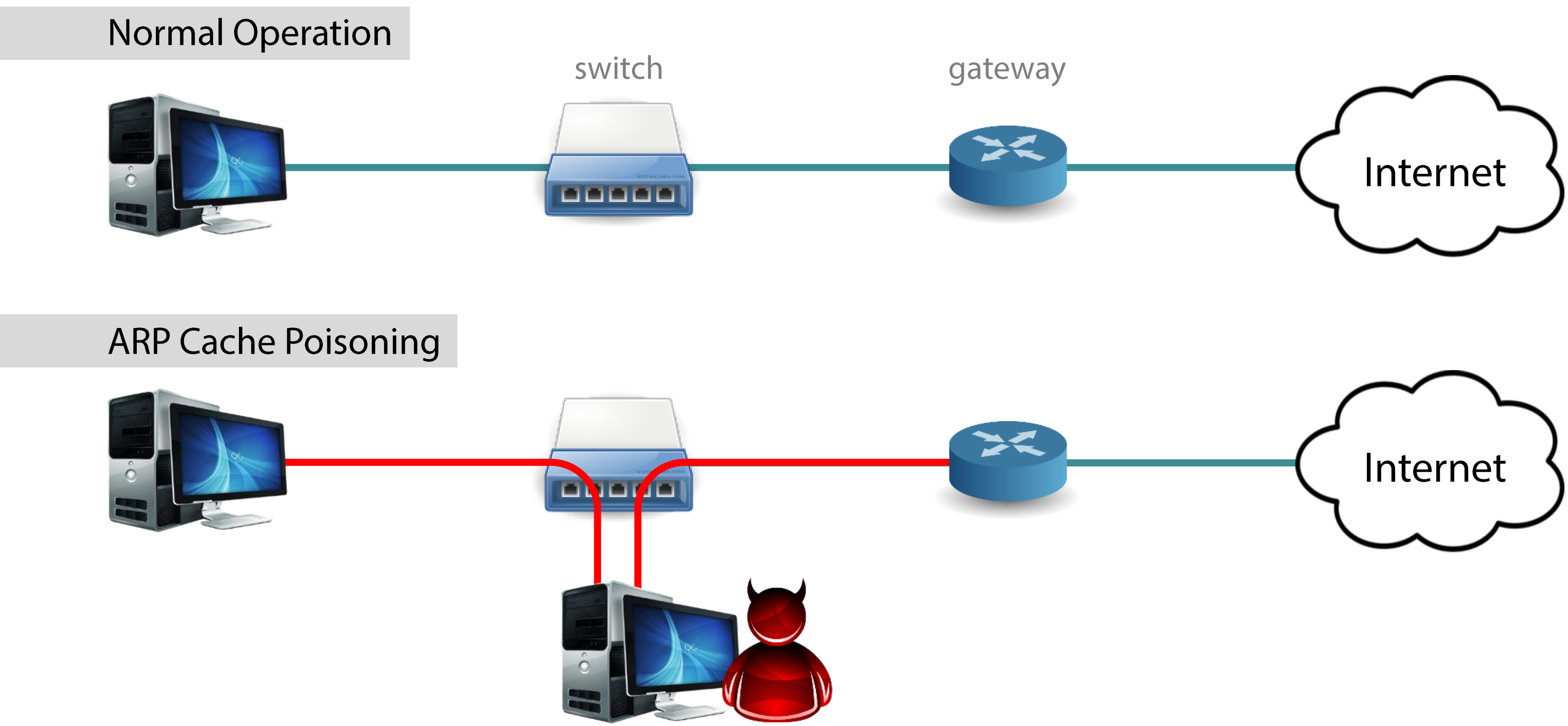
- Voluntarily announce address updates
- (NIC change, load balancing/failover, …)
- *Can be abused…*



ARP Cache Poisoning

ARP replies can be **spoofed:** IP to MAC mapping is not authenticated!

Enables traffic sniffing/manipulation through MitM



ARP Cache Poisoning

Attack steps

1. ARP reply to victim, mapping gateway's IP to attacker's MAC
2. ARP reply to gateway, mapping victim's IP to attacker's MAC
3. Just forward packets back and forth

Tools: `arpspoof` (`sslstrip`), `ettercap`, `nemesis`, …

Various Defenses

Static ARP entries: ignore ARP reply packets

OS configuration: ignore unsolicited replies, …

ARPwatch and other detection tools

Managed switches

CAM Table Exhaustion

Switches use Content Addressable Memory (CAM) to keep MAC address to port mappings

- Finite resource!

Flooding a switch with a large number of randomly generated MAC addresses can fill up the CAM table

- Failsafe operation: send all packets to all ports
- Essentially the switch turns into a hub → eavesdropping!
- Noisy attack, can be easily detected

Tool: `macof` (part of `dsniff`)

Rogue Access Points

No authentication of the AP to the client

Set up fake access point with an existing SSID or just an enticing name

- Starbucks-FREE-WiFi
- “Auto-connect”/”Ask to join network” mobile phone features greatly facilitate this kind of attacks
- Pineapple, Power Pwn, …

Wireless backdoor

- Ship an iPhone/special purpose device to an office and use 4G connection for C&C
- Hide a tiny AP in a wall plug etc.

Detection

- NetStumbler: show all WiFi networks
- RF monitoring systems
- Wireless IDS/IPS




Internet Protocol (IP)

Routing: deliver packets from a source to a destination based on the destination IP address

- Through several hops (routers) – see `traceroute`
- Connectionless, best effort: no ordering or delivery guarantees
- Source IP address is not authenticated ➔ can be easily spoofed!

IPv6: most recent version, uses 128-bit addresses

- IPv4 space has been exhausted
- IPv6 deployment is slow

If a packet is too large for the next hop, it can be *fragmented* into smaller ones

- Maximum transmission unit (MTU)

IPv4

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            Source                             |
|                           Address                             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Destination                           |
|                           Address                             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Network Layer Attacks

ICMP (Internet Control Message Protocol): Used to exchange error messages about IP datagram delivery

- Smurf Attack (DoS with spoofed broadcast Echo request)
- Reconnaissance
- Exfiltration using ICMP Tunneling
- ICMP redirect MitM
- Organizations typically block incoming/outgoing ICMP traffic

IP spoofing: conceal the real IP address of the sender

- Mostly used in DDoS attacks
- Ingress and egress filtering limit its applicability

IP fragmentation: confuse packet filters and intrusion detection systems

- Split important information across two or more packets

Transmission Control Protocol (TCP)

Provides *reliable* virtual circuits to user processes

- Connection-oriented, reliable transmission
- Packets are shuffled around, retransmitted, and reassembled to match the original data ***stream***

Sender: breaks data stream into packets

- Attaches a sequence number on each packet

Receiver: reassembles the original stream

- Acknowledges receipt of received packets
- Lost packets are sent again

TCP

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   .... data ....                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Handshake

Sequence/acknowledgement numbers

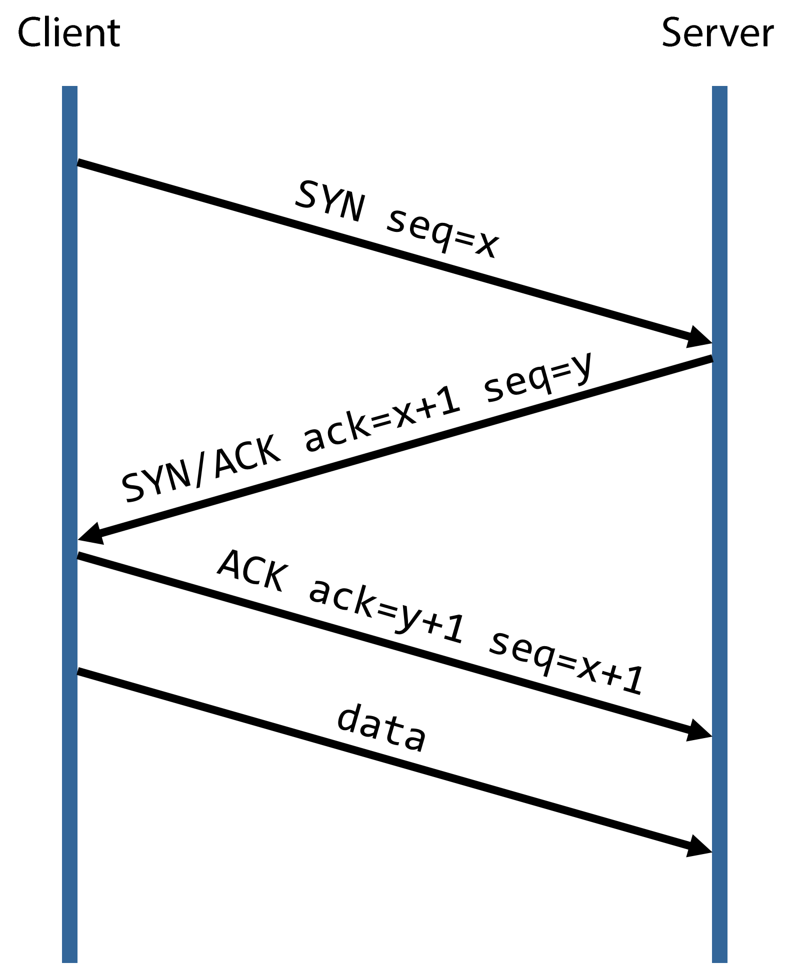
- Retransmissions, duplicate filtering, flow control

***Seq*:** the position of the segment's data in the stream

- *The payload of this segment contains data starting from X*

***Ack*:** the position of the next expected byte

- *All bytes up to X received correctly, next expected byte is X+1*



TCP Issues

Sequence Number Attacks

- TCP connection hijacking/spoofing
- DoS (connection reset)

Port scanning (future lecture)

OS Fingerprinting

- Intricacies of TCP/IP stack implementations

DoS: (future lecture)

- Resource exhaustion
- Blind RST injection

Content injection/manipulation (MitM, MotS)

TCP Sequence Number Prediction

Goal: spoof a trusted host

- Initially described by Robert Morris in 1985

Construct a valid TCP packet sequence without ever receiving any responses from the server

- Exploits predictability in initial sequence number (ISN) generation

TCP sessions are established with a three-way handshake:

- Client → Server: SYN(ISN_C)
- Server → Client: SYN(**ISN_S**), ACK(ISN_C)
- Client → Server: ACK(ISN_S)

If the ISNs generated by a host are predictable, an attacker does not need to see the SYN response to successfully establish a TCP session

Impersonating a Trusted Host

Old TCP stacks would increment the sequence number by a constant amount once per second

- Highly predictable with a single observation at a known time

Attacker impersonates trusted host, predicts **ISN_S**

- Attacker → Server: $SYN(ISN_A)$, SRC = Trusted
- Server → Trusted: $SYN(ISN_S)$, $ACK(ISN_A)$
- Attacker → Server: ACK(**ISN_S**), SRC = Trusted
- Attacker → Server: ACK(**ISN_S**), SRC = Trusted, ***attack data***

Execute commands based on lists of trusted hosts

- `rsh`, `rcp`, other "r" commands… (hopefully not used these days)

Solution: randomized ISN generation

Man-on-the-Side Attack

Packet capture + packet injection

- Sniff for requests, and forge responses

Requires a privileged position between the victim and the destination server

- Attackers **can** *observe* transmitted packets and *inject* new ones
- Attackers **cannot** *modify* or *drop* transmitted packets

But a *less privileged* position than what is required for a man-in-the-middle attack!

- Also much easier: no need to keep per-connection state and relay traffic

Example: unprotected (non-encrypted) WiFi network

- MotS: any client that joins the network can mount it
- MitM: need to compromise the access point

Man-on-the-Side Attack

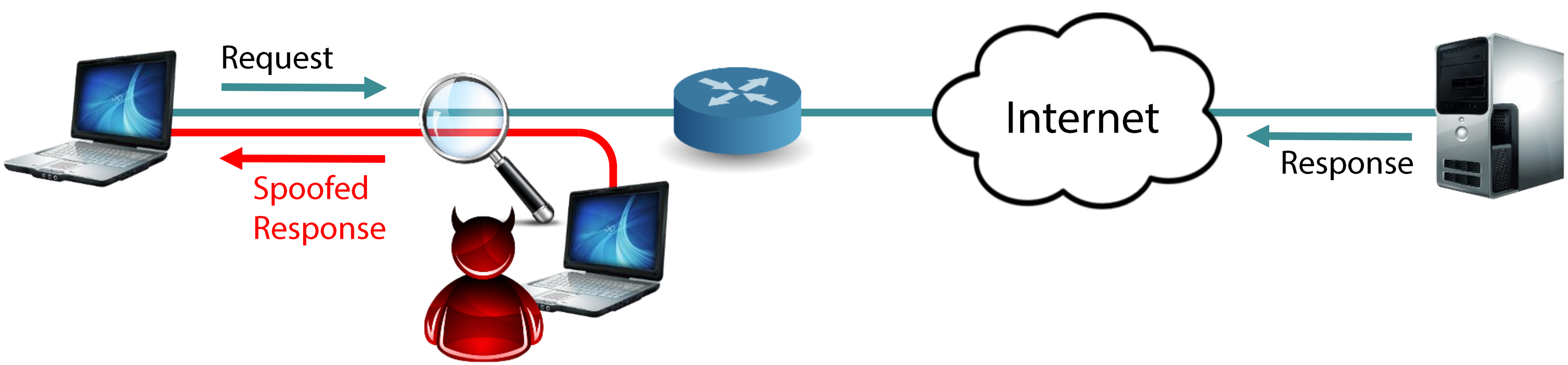
Race condition: attacker's forged response should arrive before the actual server's response

- Most OSes will accept the first packet they see as valid

No need to guess TCP seq/ack numbers!

- The rest of the original stream can follow after the injected packet

Powerful: redirect to malicious server, manipulate content, inject exploits, …



Airpwn

Listens to wireless packets and acts on interesting HTTP requests based on predefined rules

- Beating server's response is easy: the server is several hops away (10s-100s ms) while the attacker is local

```
GET / HTTP/1.1
Host: www.google.com
...
```

```
HTTP/1.1 OK
Content-length: 1462
...
<html>
<head>
<title>Google</title>
</head>
...
```

```
HTTP/1.1 OK
Content-length: 1462
...
<html>
<head>
<title>Airpwned!</title>
</head>
...
```

A Close Look at the NSA's Most Powerful Internet Attack Tool

BY NICHOLAS WEAVER 03.13.14 | 12:47 PM | PERMALINK

MOST RECENT WIRED POST

New Wh
Rules o
Surveill
Short, F
Group S

Is It Eth
Create
Three D
Sources
Absolut

Lack of
the Onl
Behind
Brutal D

Robot C
Rescue
Its Clas
Drivers

Animat
Lego M

Passive Network Monitoring

- Packet capture
  - Headers or full payloads
  - Network taps
  - Router/switch span/mirror ports
- Netflow export
  - Connection-level traffic summaries
  - Built-in capabilities in most routers
- Non-intrusive: invisible on the network
- Basis for a multitude of defenses
  - IDS/IPS
  - Anomaly detection
  - Network forensics
- Sophisticated attackers may erase all evidence on infected hosts
  - Captured network-level data might be all that is left

```
15:07:16.609603 IP 139.91.70.46.631 > 139.91.70.255.631: UDP, length 122
15:07:16.821924 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:16.821980 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:16.822297 IP 139.91.70.148.8008 > 239.255.255.250.1900: UDP, length
15:07:16.822370 IP 139.91.70.26.8008 > 239.255.255.250.1900: UDP, length
15:07:16.825070 IP 139.91.70.254 > 224.0.0.13: PIMv2, Assert, length: 28
15:07:16.826708 IP 139.91.70.253 > 224.0.0.13: PIMv2, Assert, length: 28
15:07:16.869700 endnode-hello endnode vers 2 eco 0 ueco 0 src 1.10 blksiz
rtr 0.0 hello 10 data 2
15:07:16.929894 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:17.040099 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:17.119970 IP 139.91.70.254.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
tandby group=70 addr=139.91.70.80
15:07:17.149897 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:17.259974 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:17.284411 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:17.369924 IP 139.91.171.116.1049 > 239.255.255.250.1900: UDP, lengt
15:07:17.696390 endnode-hello endnode vers 2 eco 0 ueco 0 src 1.10 blksiz
rtr 0.0 hello 10 data 2
15:07:18.764737 IP 139.91.70.253 > 224.0.0.13: PIMv2, Assert, length: 28
15:07:18.963784 IP 139.91.70.253.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
ctive group=70 addr=139.91.70.80
15:07:18.988021 IP 139.91.70.254 > 224.0.0.10: EIGRP Hello, length: 40
15:07:18.999754 IP 139.91.70.253 > 224.0.0.10: EIGRP Hello, length: 40
15:07:19.291410 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:19.351836 00:d0:d3:36:6f:54 > 01:00:0c:dd:dd:dd sap aa ui/C
15:07:19.923630 endnode-hello endnode vers 2 eco 0 ueco 0 src 1.10 blksiz
rtr 0.0 hello 10 data 2
15:07:20.004023 IP 139.91.70.254.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
tandby group=70 addr=139.91.70.80
15:07:20.821598 IP 139.91.70.148.8008 > 239.255.255.250.1900: UDP, length
15:07:21.292518 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:21.609511 IP 139.91.70.46.631 > 139.91.70.255.631: UDP, length 153
15:07:21.883722 IP 139.91.70.253.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
ctive group=70 addr=139.91.70.80
15:07:22.129438 IP 139.91.70.46.41988 > 139.91.70.255.111: UDP, length 11
15:07:22.864093 IP 139.91.70.254.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
tandby group=70 addr=139.91.70.80
15:07:23.293656 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:23.440208 IP 139.91.70.254 > 224.0.0.10: EIGRP Hello, length: 40
15:07:23.671846 IP 139.91.70.253 > 224.0.0.10: EIGRP Hello, length: 40
15:07:24.009474 IP 139.91.70.46.631 > 139.91.70.255.631: UDP, length 117
15:07:24.594258 arp who-has 139.91.70.181 tell 139.91.70.254
15:07:24.755842 IP 139.91.70.253.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
ctive group=70 addr=139.91.70.80
15:07:25.294625 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:25.609338 IP 139.91.70.46.631 > 139.91.70.255.631: UDP, length 138
15:07:25.864144 IP 139.91.70.254.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
tandby group=70 addr=139.91.70.80
15:07:26.139315 IP 139.91.70.46.41988 > 139.91.70.255.111: UDP, length 11
15:07:26.869271 endnode-hello endnode vers 2 eco 0 ueco 0 src 1.10 blksiz
rtr 0.0 hello 10 data 2
15:07:27.295746 802.1d config 2000.00:d0:00:dc:50:45.2105 root 2000.00:d0
50:45 pathcost 0 age 0 max 20 hello 2 fdelay 15
15:07:27.695642 endnode-hello endnode vers 2 eco 0 ueco 0 src 1.10 blksiz
rtr 0.0 hello 10 data 2
15:07:27.743866 IP 139.91.70.253.1985 > 224.0.0.2.1985: HSRPv0-hello 20:
ctive group=70 addr=139.91.70.80
15:07:28.067904 IP 139.91.70.253 > 224.0.0.10: EIGRP Hello, length: 40
15:07:28.264320 IP 139.91.70.254 > 224.0.0.10: EIGRP Hello, length: 40
```

Packet Capture Tools

Libpcap/Winpcap: user-level packet capture

- Standard interface used by most passive monitoring applications

PF_RING: High-speed packet capture

- Zero-copy, multicore-aware

tcpdump: just indispensable

Wireshark: tcpdump on steroids, with powerful GUI

dsniff: password sniffing and traffic analysis

ngrep: name says it all

Kismet: 802.11sniffer

many more…

Packet Parsing/Manipulation/Generation

Decode captured packets (L2 – L7)

Generate and inject new packets

Tools

- *Libnet:* one of the oldest
- *Scapy:* powerful python-based framework
- *Nemesis:* packet crafting and injection utility
- *Libdnet:* low-level networking routines
- *dpkt:* packet creation/parsing for the basic TCP/IP protocols

many more…