CSE508 Network Security

2024-02-06

Lower Layers (Part 2)

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 number
L3	*Network*		**IP**, ICMP, IPsec, …	Dst. host	IP address
L2	*Data Link*		**Eth**, 802.11, ARP, …	Next hop	MAC address
L1	*Physical*			Wire/air/pigeon	Network interface (NIC)

L3

Internet Protocol (IP)

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

- Through several hops (routers) – see `traceroute`, `tracepath`
- 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

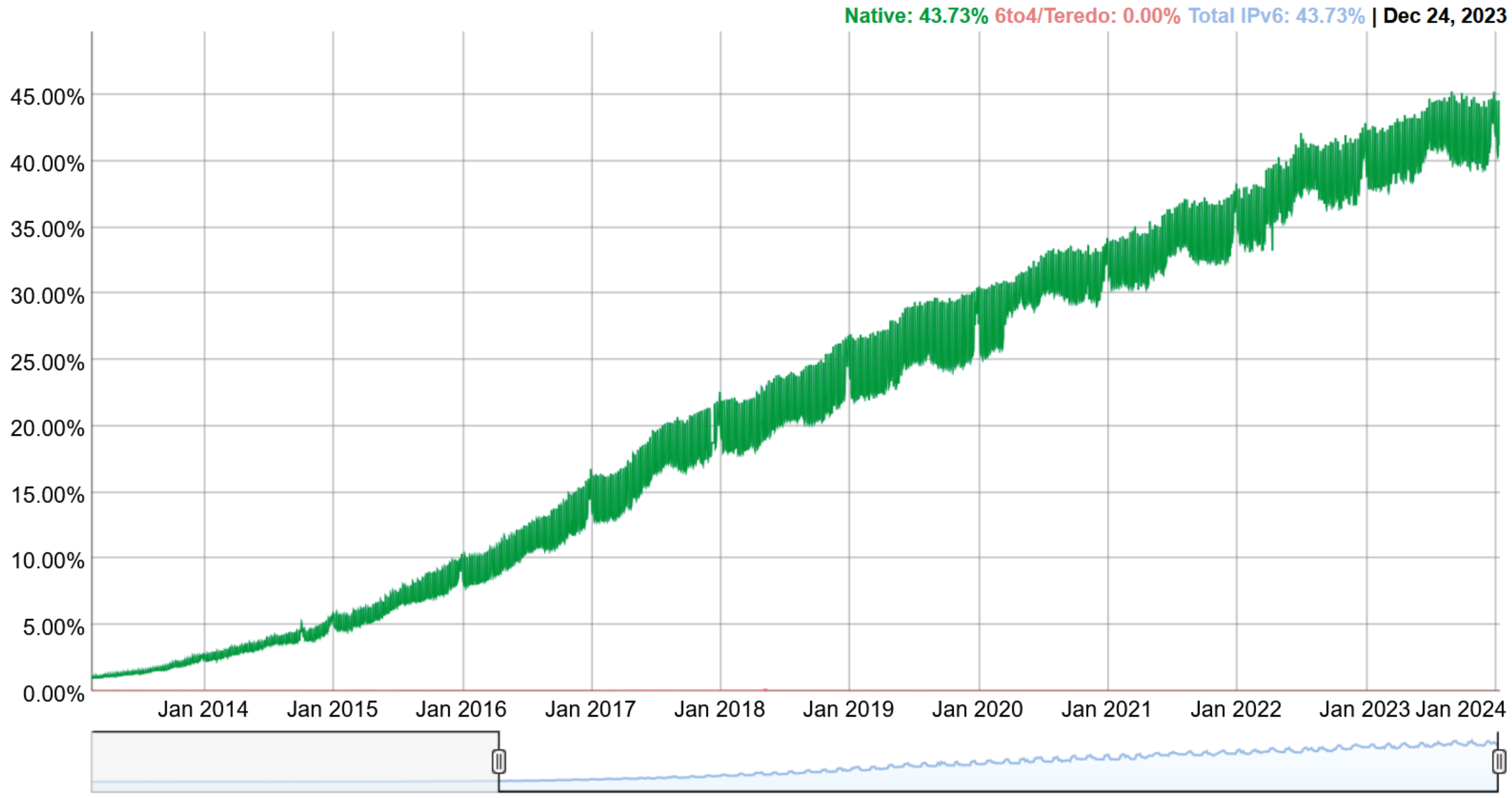
- IPv4 space was exhausted in 2011
- IPv6 deployment has been slow but is now ramping up

Packets too large for the next hop are *fragmented* into smaller ones

- Maximum transmission unit (MTU)

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



https://www.google.com/intl/en/ipv6/statistics.html

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 (L3) Attacks

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

- Smurf Attack (DoS with spoofed broadcast Echo request) (future lecture)
- Reconnaissance (future lecture)
- Exfiltration using ICMP tunneling (future lecture)
- ICMP redirect MitM
- *Organizations typically block incoming/outgoing ICMP traffic due to all the above*

IP spoofing: conceal the real IP address of the sender

- Mostly used in DDoS attacks (future lecture)
- Ingress and egress filtering limit its applicability

IP fragmentation: confuse packet filters and intrusion detection systems

- Split important information across two or more packets (future lecture)

L4

Transmission Control Protocol (TCP)

Provides *reliable* virtual circuits to user processes

- Connection-oriented
- Reliable data 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 3-way Handshake

Sequence/acknowledgement numbers

Retransmission, duplicate filtering, flow control

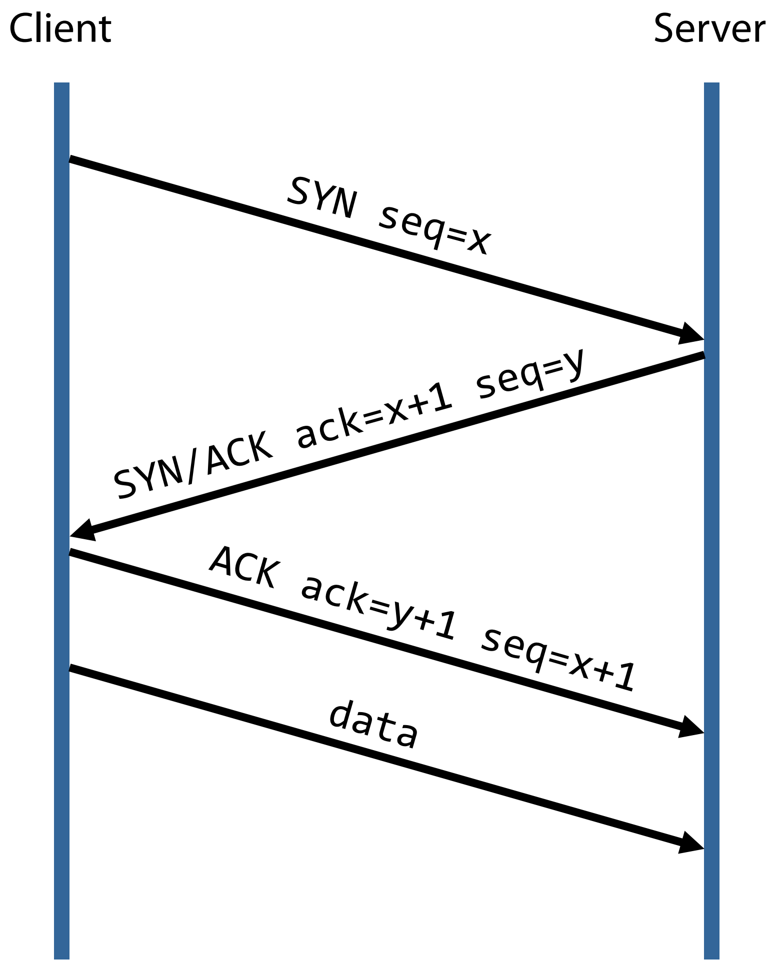
Seq: the position of the segment's data in the stream

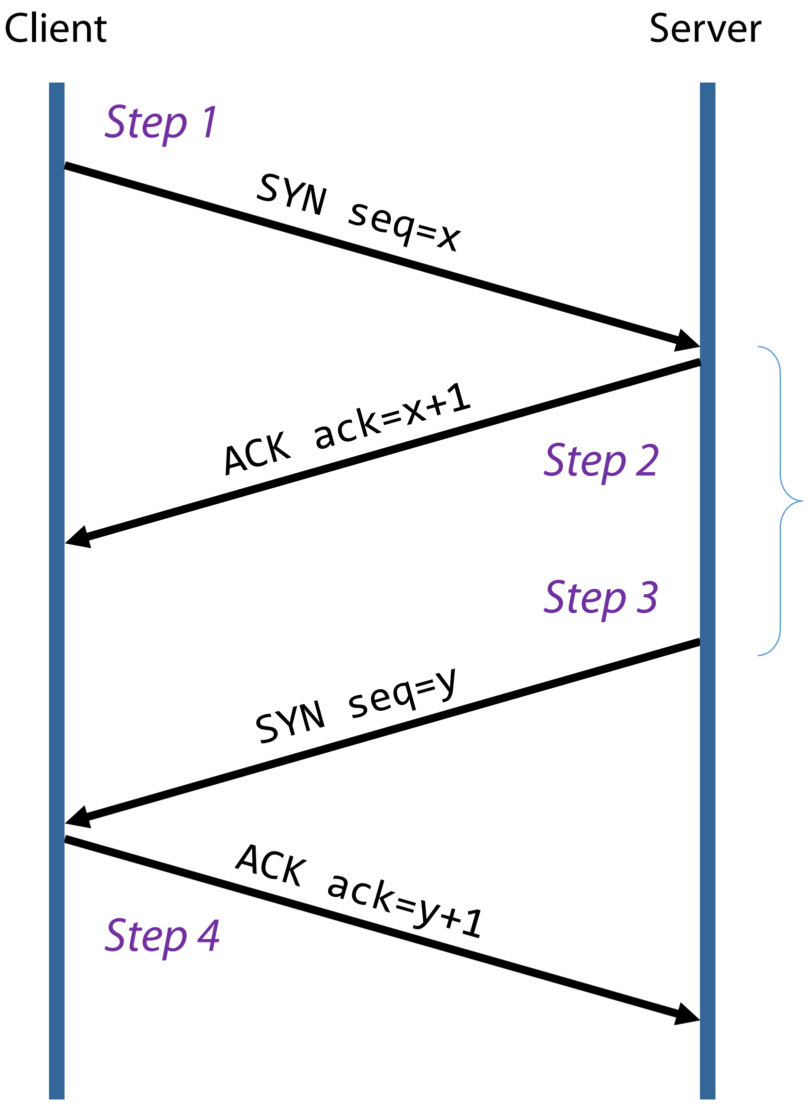
The payload of this segment contains data starting from X

Ack: the position of the next expected byte in the stream

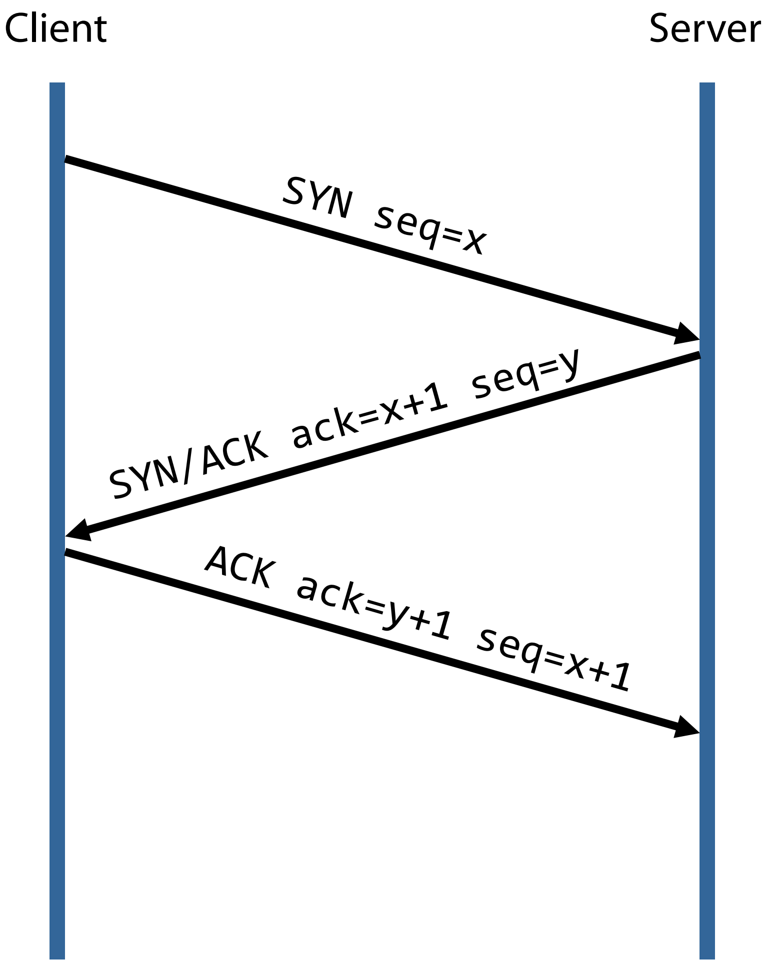
All bytes up to X received correctly, next expected byte is X+1

A SYN packet initiates a connection attempt





Steps 2+3 are combined in a single packet



Transport Layer (L4) Attacks

Sequence Number Attacks

- TCP connection hijacking/spoofing
- DoS (connection termination through informed RST injection)

Port scanning (future lecture)

OS Fingerprinting (future lecture)

- Intricacies of TCP/IP stack implementations

Denial of Service (DoS) (future lecture)

- Resource exhaustion
- Blind RST injection

Content injection/manipulation (MotS, MitM)

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 the *predictability* of initial sequence number (ISN) generation in old systems

TCP sessions are established with a three-way handshake:

- Client → Server: SYN(ISN_C)
- Server → Client: SYN($\mathbf{ISN_S}$), ACK(ISN_C)
- Client → Server: ACK(ISN_S)

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

Impersonating a Trusted Host

Old TCP stacks would increment the sequence number once per second

Highly predictable with a single observation at a known time [Bellovin '89]

Host impersonation based on a previous **ISN** observation

Attacker	➔	Server:	SYN(ISN_{A1}), SRC IP = Attacker	*Information gathering*
Server	➔	Attacker:	SYN(**ISN_{S1}**), ACK(ISN_{A1})	*Attacker learns current ISN value*
Attacker	➔	Server:	SYN(ISN_{A2}), SRC IP = Trusted	*Attack initiation*
Server	➔	Trusted:	SYN(**ISN_{S2}**), ACK(ISN_{A2})	*Attacker doesn't see this packet (!)*
Attacker	➔	Server:	ACK(**Predictable ISN_{S2}**), SRC IP = Trusted, **ATTACK DATA**	

Execute malicious commands based on lists of trusted hosts

`rsh`, `rcp`, other "`r`" commands (fit in a single packet) *hopefully not used these days*

Solution: randomized ISN generation

Man-on-the-Side Attack

On-path attacker: sniff for requests and forge responses

Required capabilities: packet capture + packet injection

Requires a privileged position between the victim and the destination

Attackers **need** to *observe* transmitted packets and *inject* new ones

Attackers **do not need** to *modify* or *drop* transmitted packets

But a *less privileged* position than what is required for a MitM 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 *right away*

MitM: need to compromise the access point or perform ARP poisoning

Man-on-the-Side Attack

Race condition

- The attacker's forged response should arrive to the victim *before* the real response from the server
- Most operating systems accept the first packet they see as valid
- Easy to win: the attacker is located closer (in terms of network hops) to the victim than the destination server

No need to guess Seq/Ack numbers

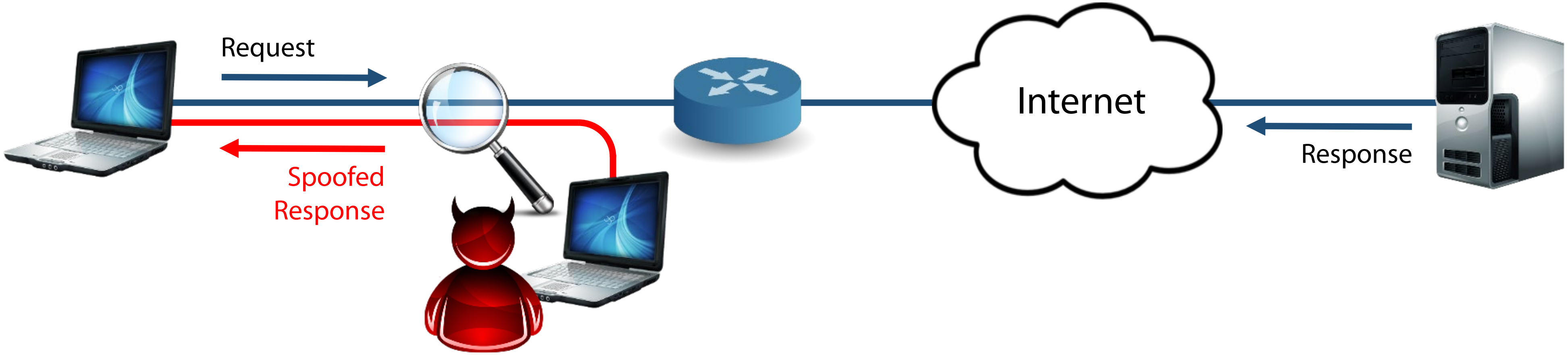
- Just sniff them from the request (!)
- The rest of the original stream follows after the injected packet

Powerful capability

- Redirect to rogue server, alter content, inject exploits, …

Man-on-the-Side Attack

***Step 1:** attacker observes outgoing TCP request packet (including Seq/Ack numbers)*



***Step 2:** attacker spoofs malicious TCP response packet*

***Step 3:** original response packet eventually arrives from the server (and is ignored by the client)*

Airpwn (2006)

Sniffs packets and acts on interesting HTTP requests based on rules

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

```
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>
...
```

https://airpwn.sourceforge.net/

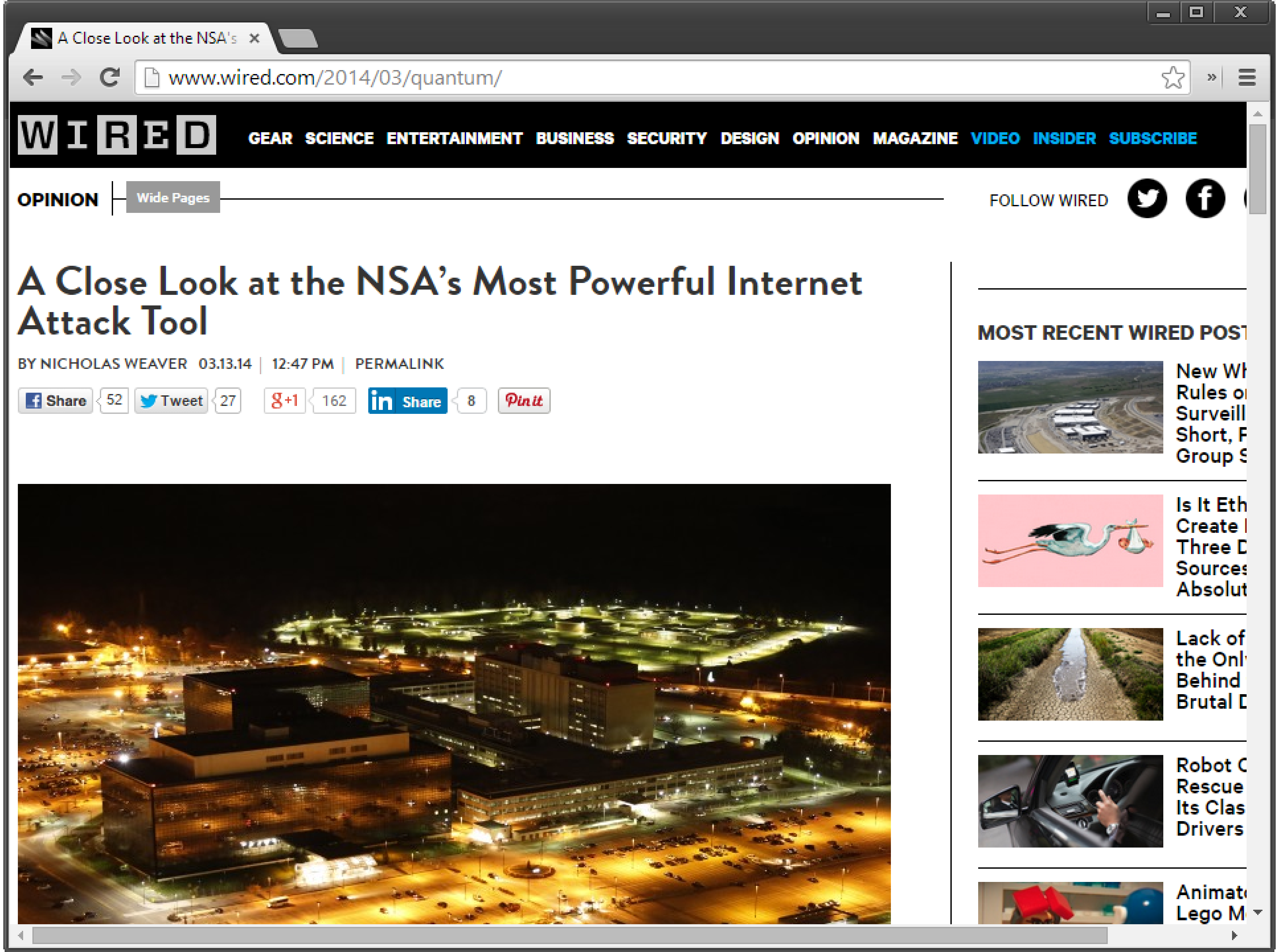
airpwn-ng https://github.com/ICSec/airpwn-ng

Overview

- We force the target's browser to do what we want
  - Most tools of this type simply listen to what a browser does, and if they get lucky, they get the cookie.
  - What if the user isn't browsing the vulnerable site at the point in time which you are sniffing?
  - Wait, you say I can't force your browser to do something? I sure can if you have cookies stored...
- Demo video: https://www.youtube.com/watch?v=hiyaUZh-UiU
- Find us on IRC (Freenode) at ##ha

Features

- Inject to all visible clients (a.k.a Broadcast Mode)
- Inject on OPEN, WEP and WPA protected networks
- Targeted injection with -t MAC:ADDRESS [MAC:ADDRESS]
- Gather all visible cookies (Broadcast Mode)
- Gather cookies for specific websites (--websites websites_list.txt)
  - In this scenario, airpwn-ng will auto-generate invisible iframes for injection that trigger the request for each website in websites_list.txt
  - [BETA] Can be used with --covert flag that attempts to inject a big iframe with the real requested website along with the generated invisible iframes. If successful, the victim should get no indication of compromise. This is still beta and doesn't work with all websites.
  - [BETA] Airpwn-ng API so you can make your own custom attacks. Examples: https://github.com/ICSec/airpwn-ng/blob/master/work-in-progress/api-examples/


A Close Look at the NSA's
www.wired.com/2014/03/quantum/
WIRED
GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIGN OPINION MAGAZINE VIDEO INSIDER SUBSCRIBE
OPINION
Wide Pages
FOLLOW WIRED
A Close Look at the NSA's Most Powerful Internet Attack Tool
BY NICHOLAS WEAVER 03.13.14 | 12:47 PM | PERMALINK
Share 52 Tweet 27 +1 162 Share 8 Pin it
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

PREDATOR IN THE WIRES

Ahmed Eltantawy Targeted with Predator Spyware After Announcing Presidential Ambitions

By Bill Marczak, John Scott-Railton, Daniel Roethlisberger, Bahr Abdul Razzak, Siena Anstis, and Ron Deibert

September 22, 2023 Arabic translation

> *"The following reply was injected by an on-path middlebox, and the legitimate reply from the server was suppressed:"*
>
> ```
> HTTP/1.1 307 Temporary Redirect
> Via: 1.0 middlebox
> Location: https://c.betly[.]me/[REDACTED]
> Connection: close
> ```

Apple has just issued an update for Apple products including iPh
Watches. We encourage all users to immediately update their device

Key Findings

- Between May and September 2023, former Egyptian MP Ahmed Eltantawy was targeted with Cytrox's Predator spyware via links sent on SMS and WhatsApp. The targeting took place after Eltantawy publicly stated his plans to run for President in the 2024 Egyptian elections.
- In August and September 2023, Eltantawy's Vodafone Egypt mobile connection was persistently selected for targeting via network injection; when Eltantawy visited certain websites not using HTTPS, a device installed at the border of Vodafone Egypt's network automatically redirected him to a malicious website to infect his phone with Cytrox's Predator spyware.
- During our investigation, we worked with Google's Threat Analysis Group (TAG) to

Man-in-the-Middle Attack

In-path attacker: can inject new and modify or drop existing packets

More powerful than an *on-path* adversary (Man-on-the-Side) who can inject new packets but *cannot* alter existing packets (just observe them)

Many ways to achieve an in-path position

ARP poisoning

Rogue or compromised router, VPN server, firewall, gateway, access point, …

Physically interjected network bridge or transparent/intercepting/inline proxy

Software-level interception (browser extension, parental control filter, anti-virus, …)

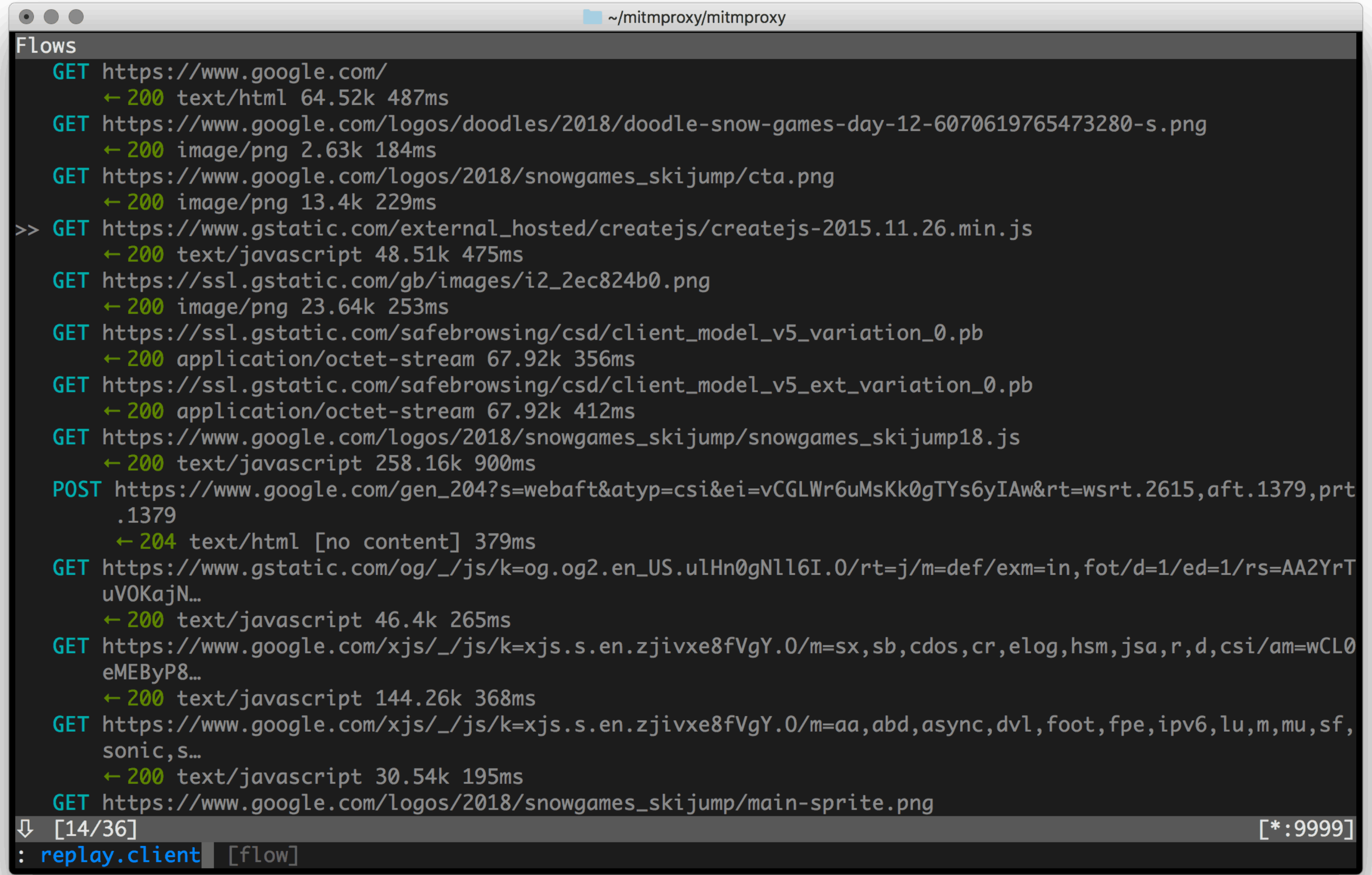
Bettercap https://www.bettercap.org/

bettercap is a powerful, easily extensible and portable framework written in Go which aims to offer to security researchers, red teamers and reverse engineers an **easy to use**, **all-in-one solution** with all the features they might possibly need for performing reconnaissance and attacking WiFi networks, Bluetooth Low Energy devices, wireless HID devices and Ethernet networks.

Main Features

- **WiFi** networks scanning, deauthentication attack, clientless PMKID association attack and automatic WPA/WPA2 client handshakes capture.
- **Bluetooth Low Energy** devices scanning, characteristics enumeration, reading and writing.
- 2.4Ghz wireless devices scanning and **MouseJacking** attacks with over-the-air HID frames injection (with DuckyScript support).
- Passive and active IP network hosts probing and recon.
- **ARP, DNS and DHCPv6 spoofers** for MITM attacks on IP based networks.
- **Proxies at packet level, TCP level and HTTP/HTTPS** application level fully scriptable with easy to implement **javascript plugins**.
- A powerful **network sniffer** for **credentials harvesting** which can also be used as a **network protocol fuzzer**.
- A very fast port scanner.
- A powerful REST API with support for asynchronous events notification on websocket to orchestrate your attacks easily.
- An easy to use web user interface.
- More!

mitmproxy https://mitmproxy.org/



CoffeeMiner https://github.com/arnaucube/coffeeMiner

Collaborative (mitm) cryptocurrency mining pool in wifi networks

Warning: this project is for academic/research purposes only.

A blog post about this project can be read here: http://arnaucode.com/blog/coffeeminer-hacking-wifi-cryptocurrency-miner.html

Concept

- Performs a MITM attack to all selected victims
- Injects a js script in all the HTML pages requested by the victims
- The js script injected contains a cryptocurrency miner
- All the devices victims connected to the Lan network, will be mining for the CoffeeMiner