

CS 4910: Intro to Computer Security

Network Security I: Computer Network
Concepts & Network Attacks

Instructor: Xi Tan

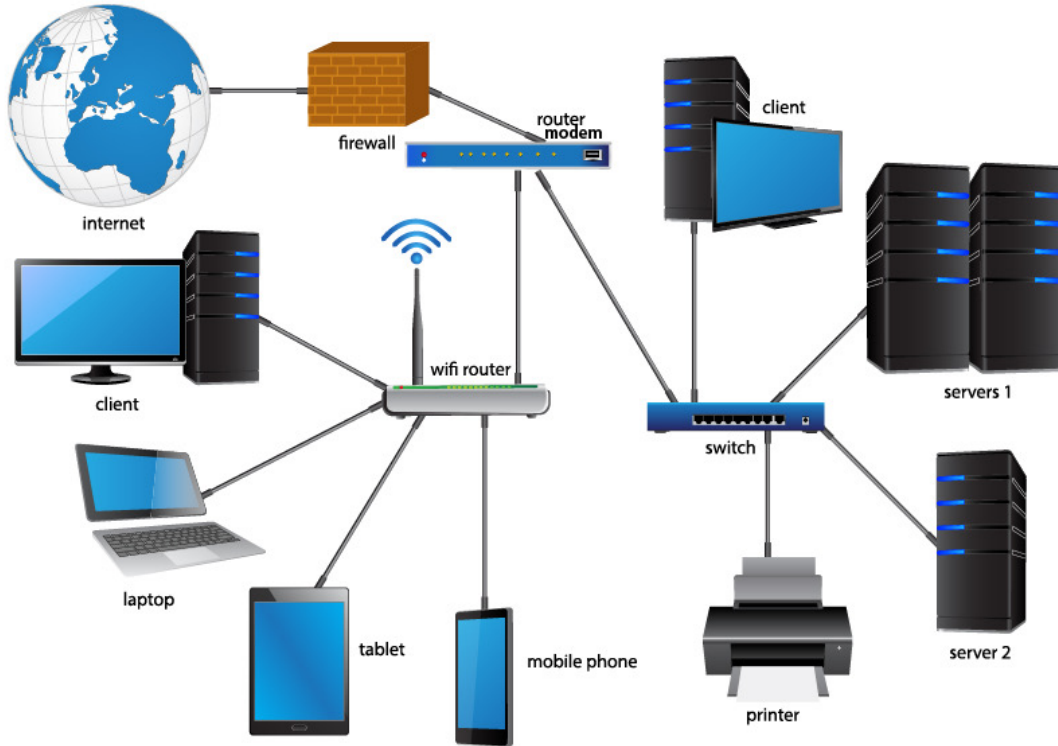
Updates

- Lab 2:
 - Task 1: Packet sniffing and spoofing
 - Task 2: Not required
 - **Deadline: 3/31**
- Homework 3
 - **Deadline: ~~04/02~~ 04/07**
- Research Paper:
 - Research Paper Topic Selection
 - **Deadline: 03/19**

Network Security

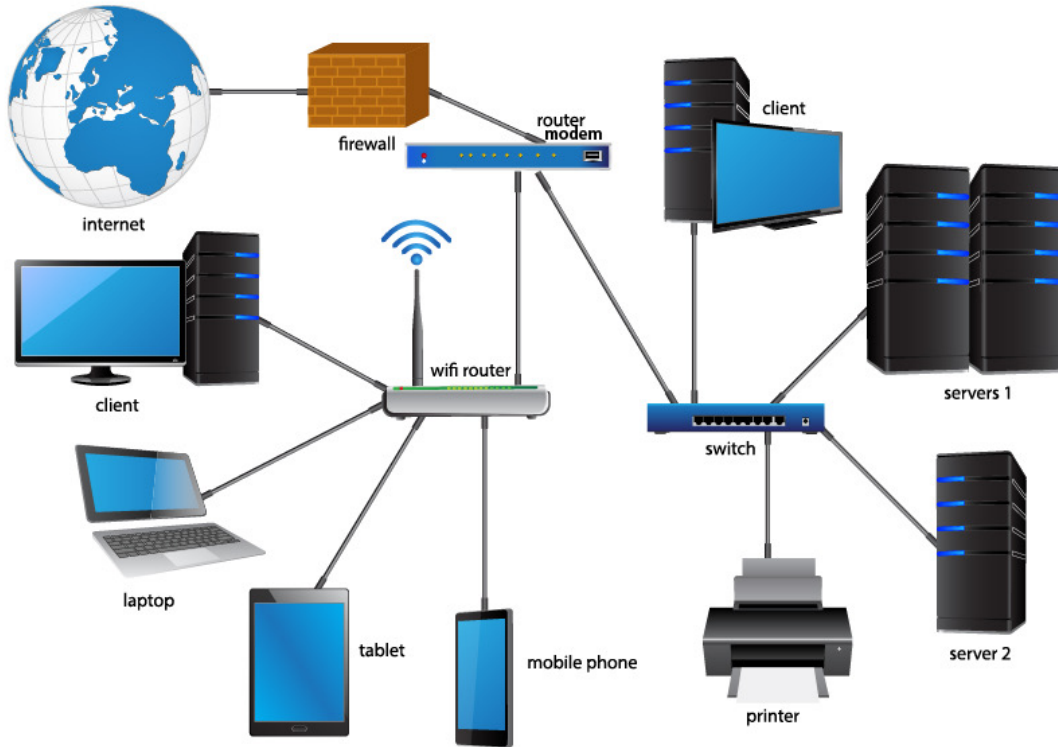
- Computer Network Concepts
- Network Attacks
- Network Security

What is A Computer Network?



A computer network is a **collection** of computers and other devices connected together to **communicate** and share resources.

Important Components for A Network

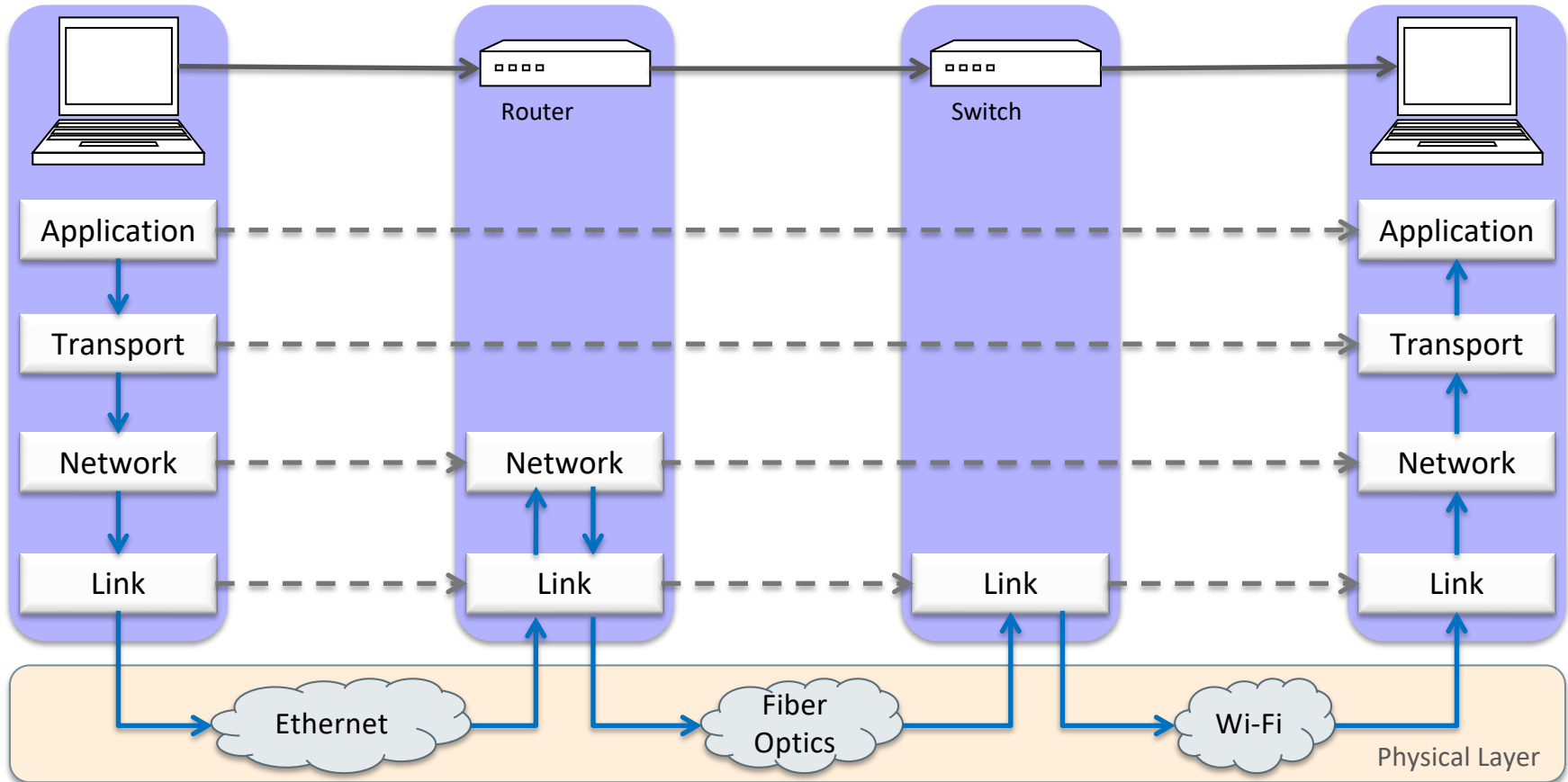


How are those devices connected?

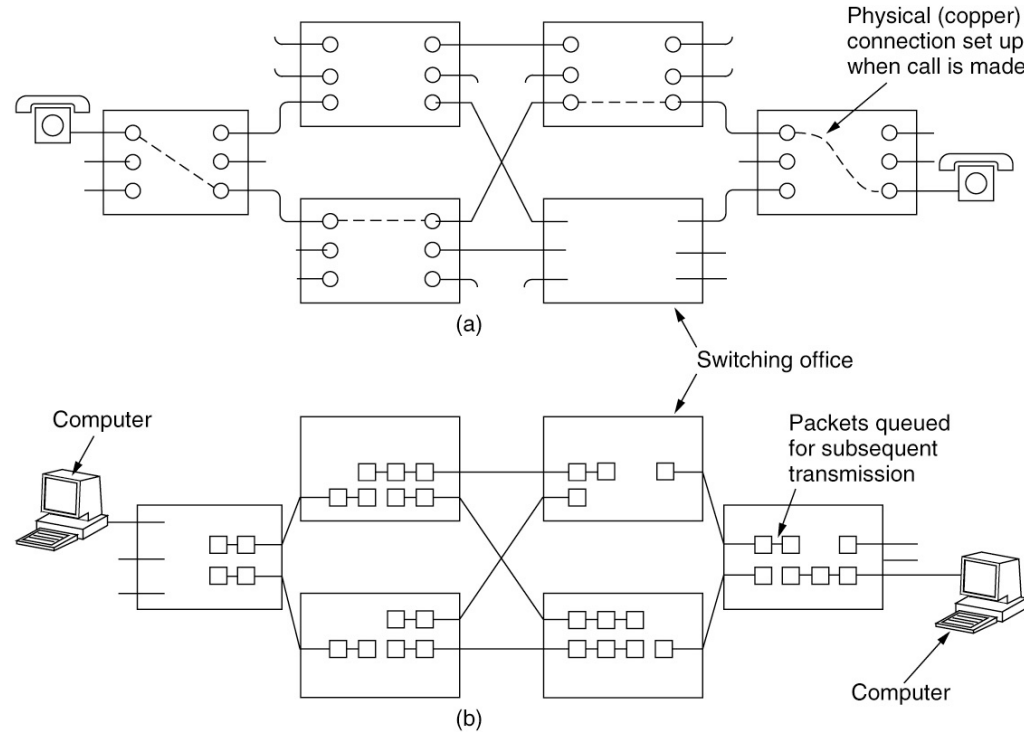
How is the data (frame/packet) transmitted through the connection?

How do those devices interpret the data?

How are those devices connected? -- Network Layers



How is the data transmitted through the connection?



(a) Circuit switching (b) Packet switching

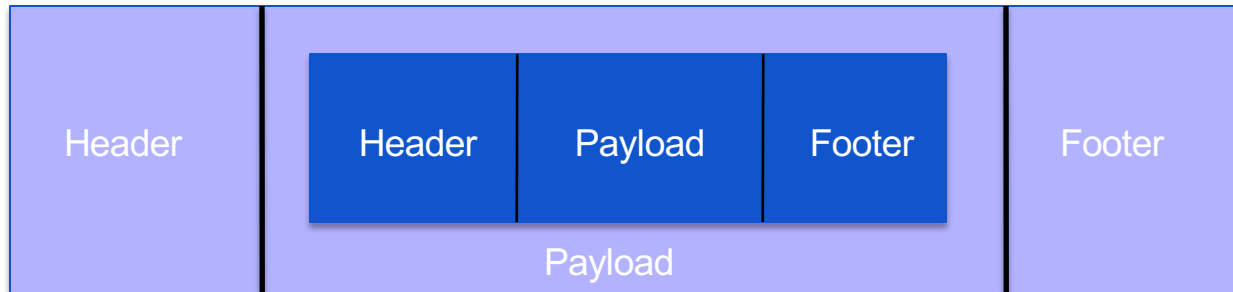
How do those devices interpret the data?

Protocols

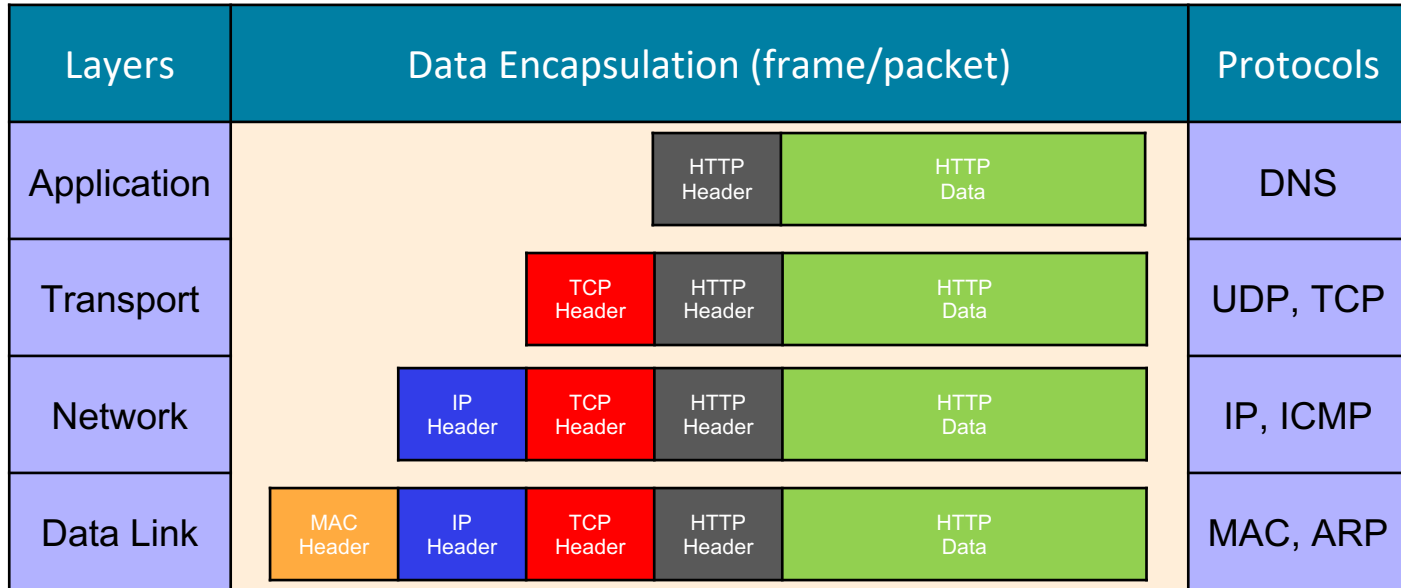
- A protocol defines the rules for communication between computers
- Protocols are broadly classified as connectionless and connection oriented
 - Connectionless protocol
 - Sends data out as soon as there is enough data to be transmitted
 - E.g., user datagram protocol (UDP)
 - Connection-oriented protocol
 - Provides a reliable connection stream between two nodes
 - Consists of set up, transmission, and tear down phases
 - Creates virtual circuit-switched network
 - E.g., transmission control protocol (TCP)

Encapsulation

- A packet typically consists of
 - **Control information** for addressing the packet: header and footer
 - Data: payload
- A network protocol N1 can use the services of another network protocol N2
 - A packet p1 of N1 is encapsulated into a packet p2 of N2
 - The payload of p2 is p1
 - The control information of p2 is derived from that of p1



Internet Communication



Network Interfaces

- Network interface: device connecting a computer to a network
 - Ethernet card
 - WiFi adapter
- A computer may have multiple network interfaces
- Packets transmitted between network interfaces
- Most local area networks, (including Ethernet and WiFi) broadcast frames
- In regular mode, each network interface gets the frames intended for it
- **Traffic sniffing** can be accomplished by configuring the network interface to read all frames

Packet Sniffers

- Packet sniffers “read” information traversing a network
 - Packet sniffers intercept network packets, possibly using ARP cache poisoning
 - Can be used as **legitimate** tools to analyze a network
 - Monitor network usage
 - Filter network traffic
 - Analyze network problems
 - Can also be used **maliciously**
 - **Steal** information (i.e. passwords, conversations, etc.)
 - Analyze network information to **prepare** an attack
- Packet sniffers can be either software or hardware based
 - Sniffers are dependent on network setup

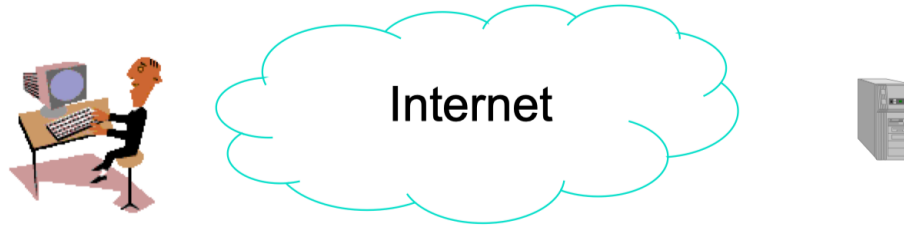
Packet Sniffers

- What can we get from packet sniffers?
 - Packet header
 - Payload data
 - Unencrypted sensitive data
 - Protocols in use
- Tools?
 - Wireshark, tcpdump, etc.

Detecting Sniffers

- Sniffers are almost always passive
 - They simply collect data
 - They do not attempt “entry” to “steal” data
- This can make them extremely **hard** to detect
- To reduce the impact of packet sniffing, **encryption** mechanisms should be utilized in **higher-level protocols** to prevent attackers from recovering **sensitive** data

Network Analyzer -- Wireshark



- User clicks on <http://www.nytimes.com/>
- Network analyzer captures all frames observed by its NIC
- Sequence of frames and contents of frame can be examined in detail down to individual bytes

The screenshot shows the Wireshark interface with two panes highlighted by red callouts:

- Left Pane (Packet Details):** Shows the structure of the captured packet. The selected packet is a DNS query (Transaction ID: 0x6575). The details include:
 - Ethernet II, Src: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d), Dst: All-HSRP-routers_
 - Internet Protocol Version 4, Src: 128.198.212.72, Dst: 128.198.4.52
 - User Datagram Protocol, Src Port: 55293, Dst Port: 53
 - Domain Name System (query)
 - Transaction ID: 0x6575
 - Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
- Right Pane (Packet Bytes):** Shows the raw data of the packet in hexadecimal and ASCII. The ASCII column shows the text representation of the data, including the domain name "es.com".

Red callouts with white text are placed over the panes:

- Left Pane shows encapsulation for a given packet
- Right Pane shows hex & text

Top pane: Frame Sequence

Wi-Fi: en0

Apply a display filter ... < %/ >

DNS Query

No.	Time	Source	Destination	Protocol	Length	Info
255	1.344089	128.198.212.72	128.198.4.52	DNS	75	Standard query 0x6575 A www.nytimes.com
256	1.344180	128.198.212.72	128.198.4.52	DNS	75	Standard query 0x9cd6 HTTPS www.nytimes.com
257	1.344267	128.198.212.72	128.198.4.52	DNS	80	Standard query 0xa567 A static01.nytimes.com
258	1.344324	128.198.212.72	128.198.4.52	DNS	80	Standard query 0xc973 HTTPS static01.nytimes.com
259	1.344406	128.198.212.72	128.198.4.52	DNS	70	Standard query 0x21fb A gl.nyt.com
260	1.344457	128.198.212.72	128.198.4.52	DNS	70	Standard query 0xd5b4 HTTPS gl.nyt.com
261	1.351101	128.198.4.52	128.198.212.72	DNS	196	Standard query response 0x6575 A www.nytimes.com CNAME www.prn.map.nytimes.co
262	1.351104	128.198.4.52	128.198.212.72	DNS	207	Standard query response 0xa567 A static01.nytimes.com CNAME static.prn.map.ny
263	1.351105	128.198.4.52	128.198.212.72	DNS	215	Standard query response 0x21fb A gl.nyt.com CNAME nyt5-assets.prn.map.nytimes
264	1.354893	128.198.4.52	128.198.212.72	DNS	191	Standard query response 0xc973 HTTPS static01.nytimes.com CNAME static.prn.ma
265	1.354896	128.198.4.52	128.198.212.72	DNS	257	Standard query response 0xd5b4 HTTPS gl.nyt.com CNAME nyt5-assets.prn.map.ny
266	1.355354	128.198.212.72	151.101.69.164	TCP	78	57839 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=441735617 TSecr=
267	1.355482	128.198.212.72	151.101.69.164	TCP	78	57840 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1405589582 TSecr=
268	1.356739	128.198.212.72	151.101.69.164	TCP	78	57841 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=220190727 TSecr=
269	1.358135	128.198.4.52	128.198.212.72	DNS	180	Standard query response 0x9cd6 HTTPS www.nytimes.com CNAME www.prn.map.nytime
270	1.358191	128.198.212.72	128.198.4.52	ICMP	70	Destination unreachable (Port unreachable)
271	1.360504	151.101.69.164	128.198.212.72	TCP	74	443 → 57840 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1382 SACK_PERM TSval=4
272	1.360507	151.101.69.164	128.198.212.72	TCP	74	443 → 57839 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1382 SACK_PERM TSval=4
273	1.360650	128.198.212.72	151.101.69.164	TCP	66	57840 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=1405589587 TSecr=4589806
274	1.360727	128.198.212.72	151.101.69.164	TCP	66	57839 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=441735622 TSecr=4589806
275	1.360817	151.101.69.164	128.198.212.72	TCP	74	443 → 57841 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1382 SACK_PERM TSval=2
276	1.360874	128.198.212.72	151.101.69.164	TCP	1436	57840 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=1405589587 TSecr=4589
277	1.360884	128.198.212.72	151.101.69.164	TCP	66	57841 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=220190731 TSecr=29040753
278	1.360887	128.198.212.72	151.101.69.164	TLSv1.3	783	Client Hello (SNI=gl.nyt.com)
279	1.361074	128.198.212.72	151.101.69.164	TCP	1436	57839 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=441735622 TSecr=4589
280	1.361078	128.198.212.72	151.101.69.164	TLSv1.3	681	Client Hello (SNI=static01.nytimes.com)
281	1.361245	128.198.212.72	151.101.69.164	TCP	1436	57841 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=220190731 TSecr=2904
282	1.361248	128.198.212.72	151.101.69.164	TLSv1.3	676	Client Hello (SNI=www.nytimes.com)
283	1.370815	151.101.69.164	128.198.212.72	TCP	66	443 → 57841 [ACK] Seq=1 Ack=1371 Win=147968 Len=0 TSval=290407544 TSecr=2201
284	1.370817	151.101.69.164	128.198.212.72	TCP	66	443 → 57839 [ACK] Seq=1 Ack=1986 Win=148992 Len=0 TSval=458980691 TSecr=44173
285	1.370819	151.101.69.164	128.198.212.72	TCP	66	443 → 57840 [ACK] Seq=1 Ack=2008 Win=148992 Len=0 TSval=458980691 TSecr=14055
286	1.370820	151.101.69.164	128.198.212.72	TCP	66	443 → 57841 [ACK] Seq=1 Ack=1981 Win=150528 Len=0 TSval=290407545 TSecr=2201
287	1.370822	151.101.69.164	128.198.212.72	TLSv1.3	519	Server Hello, Change Cipher Spec, Application Data, Application Data, Applica
288	1.370823	151.101.69.164	128.198.212.72	TLSv1.3	519	Server Hello, Change Cipher Spec, Application Data, Application Data, Applica
289	1.370825	151.101.69.164	128.198.212.72	TLSv1.3	519	Server Hello, Change Cipher Spec, Application Data, Application Data, Applica
290	1.370941	128.198.212.72	151.101.69.164	TCP	66	57841 → 443 [ACK] Seq=1981 Ack=454 Win=131008 Len=0 TSval=220190742 TSecr=29
291	1.370997	128.198.212.72	151.101.69.164	TCP	66	57839 → 443 [ACK] Seq=1986 Ack=454 Win=131008 Len=0 TSval=441735633 TSecr=45

> Frame 255: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interfa

> Ethernet II, Src: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d), Dst: All-HSRP-routers_

> Internet Protocol Version 4, Src: 128.198.212.72, Dst: 128.198.4.52

> User Datagram Protocol, Src Port: 55293, Dst Port: 53

> Domain Name System (query)

0000 00 00 0c 07 ac 01 2e de b7 57 c6 7d 08 00 45 00W}.E.

0010 00 3d 4a 92 00 00 00 11 56 15 80 c6 d4 48 00 c6 ... =J.~ V...H.

0020 04 34 d7 fd 00 35 00 29 a7 67 65 75 01 00 00 01 ... 4...5.) geu....

0030 00 00 00 00 00 00 03 77 77 77 07 6e 79 74 69 6dw ww.nytim

0040 65 73 03 63 6f 6d 00 00 01 00 01 ... es.com....

TCP connection
establishment:
SYN, SYN-ACK, ACK

TLS handshake:
Client hello, server
hello, key
exchange and final
handshake
(Required when
using HTTPS)

Left Pane: Encapsulation

Wi-Fi: en0

Apply a display filter ... <@/>

No.	Time	Source	Destination	Protocol	Length	Info
277	1.360884	128.198.212.72	151.101.69.164	TCP	66	57841 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=220190731 TSecr=29040753
278	1.360887	128.198.212.72	151.101.69.164	TLSv1.3	703	Client Hello (SNI=g1.nyt.com)
279	1.361074	128.198.212.72	151.101.69.164	TCP	1436	57839 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=441735622 TSecr=45898

Frame 278: 703 bytes on wire (5624 bits) · 703 bytes captured (5624 bits) on interface en0 id 0

Ethernet II, Src: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d), Dst: All-HSRP-routers_01 (00:00:0c:07:ac:01)

> Destination: All-HSRP-routers_01 (00:00:0c:07:ac:01)

> Source: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d)

> Type: IPv4 (0x0800)

[Stream index: 0]

Internet Protocol Version 4, Src: 128.198.212.72, Dst: 151.101.69.164

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 689

Identification: 0x0000 (0)

> 010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: TCP (6)

Header Checksum: 0x062f [validation disabled]

[Header checksum status: Unverified]

Source Address: 128.198.212.72

Destination Address: 151.101.69.164

[Stream index: 2]

> Transmission Control Protocol, Src Port: 57840, Dst Port: 443, Seq: 1371, Ack: 1, Len: 637

> [2 Reassembled TCP Segments (2007 bytes): #276(1370), #278(637)]

> Transport Layer Security

> TLSv1.3 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 2002

> Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 1998

> Version: TLS 1.2 (0x0303)

Random: 0d00cb56e1cd595d56da4648ead6142b87c692c8a0d66dae2d5bc624d5bbadcf

0000 16 03 01 07 d2 01 00 07 ce 03 03 0d 00 cb 56 e1

0010 cd 59 5d 56 da 46 48 ea d6 14 2b 87 c6 92 c8 a0

0020 d6 6d ae 2d 5b c6 24 d5 bb ad cf 20 fa c2 2d cf

0030 08 de 48 12 c4 9c 87 b1 0e ef af 7a f7 bb da 15

0040 b0 56 a5 ce 69 61 c0 b7 29 6e 15 ea 00 20 6a 6a

0050 13 01 13 02 13 03 c0 2b c0 2f c0 2c c0 30 cc a9

0060 cc a8 c0 13 c0 14 00 9c 00 9d 00 2f 00 35 01 00

0070 07 65 8a 8a 00 00 00 0a 00 0c 00 0a 2a 2a 63 99

0080 00 1d 00 17 00 18 ff 01 00 01 00 00 00 0f 00

0090 0d 00 00 0a 67 31 2e 6e 79 74 2e 63 6f 6d 00 23

00a0 00 00 00 33 04 ef 04 ed 2a 2a 00 01 00 63 99 04

00b0 c0 00 65 29 46 be 96 8f 5c 9d 35 8d 42 36 2a 3d

00c0 c3 66 df 0b ae 01 f2 80 33 ea d1 7f 4e b3 cf 5e

00d0 75 08 d3 0f fe 16 bd 6a dc aa ce 76 1e 7c 40 22

00e0 82 70 16 96 9b 63 ff c1 5f e5 38 cd ca a9 17 f0

00f0 76 3e 02 dc cc ae 66 43 71 c1 9e 75 71 07 55 b7

0100 17 2c 56 56 36 e0 20 39 f8 4c 6f 1b 7e 0c 0a 37

0110 f7 d0 05 a0 34 b1 ae 2c 5e 31 51 c6 fb 92 6c f0

0120 a2 1e 4c c1 1e f8 a5 7e 60 86 21 de b2 8c a5 ea

0130 ca 56 c8 78 85 2b 7d 1b 91 81 9c b6 4a 5d f8 b7

0140 82 37 bb d1 46 2c a6 34 8f 47 39 08 27 09 38 20

0150 d4 4d 0a 52 3e 49 99 9e f1 41 3c 5c 69 81 68 5a

0160 0c a8 f2 4e f0 2a c8 c2 61 3d 2b b4 73 59 27 b4

0170 47 82 92 2a f3 29 27 be 2e 9e 17 0a f7 a0 0e 13

0180 11 ce 6d db c4 03 c4 59 a0 3b 54 3e 21 be 5c b1

0190 02 2b e5 67 c5 29 b8 19 d7 3e 24 4c 70 d1 41 84

01a0 89 3b 36 6a 44 4e f0 6b 9d 75 d1 6c b3 57 58 65

01b0 12 8d ad c9 0d 47 88 38 a8 72 36 94 38 7c 15 1b

01c0 06 dd db 12 8f 86 40 5d fc 51 11 d5 0b f0 15 25

01d0 ae 84 6a 03 29 3c 82 44 1d 11 91 21 f0 8b 2c cf

01e0 72 b3 17 47 21 79 9b 92 a0 d3 c5 10 13 3f e2 48

01f0 66 77 52 02 ca 6c 94 64 9a 8f 68 95 bd c8 28 a5

0200 2b f6 c4 e5 81 82 a5 82 9f ec 70 46 5e 25 ae b3

0210 e5 a4 d9 91 1e 7a 5c cf 7b 06 24 95 02 8e 2a b6

0220 42 dc 44 93 b3 c8 40 e4 f1 b4 b2 01 5e 63 26 69

0230 70 c4 a6 08 95 cd 55 ea 3f 27 a6 78 62 02 0e 6d

0240 36 b2 76 16 02 33 65 36 1b e0 05 e5 35 ab 97 66

0250 bf 07 e5 48 04 20 94 a3 1a 87 11 55 1f 0e 47 a1

0260 f2 81 48 e0 25 78 c6 92 4d 3c c4 46 48 73 98 fb

0270 67 49 3c f5 00 32 c6 72 d7 28 17 5d 20 32 d1 1a

Ethernet frame

Protocol type

Ethernet destination and source addresses

Left Pane: Encapsulation

Wi-Fi: en0

Apply a display filter ... <@/>

No.	Time	Source	Destination	Protocol	Length	Info
277	1.360884	128.198.212.72	151.101.69.164	TCP	66	57841 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=220190731 TSecr=29040753
278	1.360887	128.198.212.72	151.101.69.164	TLSv1.3	703	Client Hello (SNI=g1.nyt.com)
279	1.361074	128.198.212.72	151.101.69.164	TCP	1436	57839 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=441735622 TSecr=45898

> Frame 278: 703 bytes on wire (5624 bits), 703 bytes captured (5624 bits) on interface en0, id 0

▼ Ethernet II, Src: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d), Dst: All-HSRP-routers_01 (00:00:0c:07:ac:01)

> Destination: All-HSRP-routers_01 (00:00:0c:07:ac:01)

> Source: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d)

Type: IPv4 (0x0800)

[Stream index: 0]

▼ Internet Protocol Version 4, Src: 128.198.212.72, Dst: 151.101.69.164

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 689

Identification: 0x0000 (0)

> 010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: TCP (6)

Header Checksum: 0x062f [validation disabled]

[Header checksum status: Unverified]

Source Address: 128.198.212.72

Destination Address: 151.101.69.164

[Stream index: 2]

> Transmission Control Protocol, Src Port: 57840, Dst Port: 443, Seq: 1371, Ack: 1, Len: 637

> [2 Reassembled TCP Segments (2007 bytes): #276(1370), #278(637)]

▼ Transport Layer Security

▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 2002

▼ Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 1998

> Version: TLS 1.2 (0x0303)

Random: 0d00cb56e1cd595d56da4648ead6142b87c692c8a0d66dae2d5bc624d5bbadcf

0000 16 03 01 07 d2 01 00 07 ce 03 03 0d 00 cb 56 e1

0010 cd 59 5d 56 da 46 48 ea d6 14 2b 87 c6 92 c8 a0

0020 d6 6d ae 2d 5b c6 24 d5 bb ad cf 20 fa c2 2d cf

0030 08 de 48 12 c4 9c 87 b1 0e ef af 7a f7 bb da 15

0040 b0 56 a5 ce 69 61 c0 b7 29 6e 15 ea 00 20 6a 6a

0050 13 01 13 02 13 03 c0 2b c0 2f c0 2c c0 30 cc a9

0060 cc a8 c0 13 c0 14 00 9c 00 9d 00 2f 00 35 01 00

0070 07 65 8a 8a 00 00 00 0a 00 0c 00 0a 2a 2a 63 99

0080 00 1d 00 17 00 18 ff 01 00 01 00 00 00 0f 00

0090 0d 00 00 0a 67 31 2e 6e 79 74 2e 63 6f 6d 00 23

00a0 00 00 00 33 04 ef 04 ed 2a 2a 00 01 00 63 99 0a

00b0 c0 0b 65 29 ae 06 9e 8f 5c 9d 35 8d 42 36 2a 3d

00c0 c3 66 df 0b 46 b1 f2 80 33 ea d1 7f 4e b3 cf 5e

00d0 75 08 d3 0f fe 16 bd 6a dc aa ce 76 1e 7c 40 22

00e0 82 70 16 96 9b 63 ff c1 5f e5 38 cd ca a9 17 f0

00f0 76 3e 02 dc cc ae 66 43 71 c1 9e 75 71 07 55 b7

0100 17 2c 56 56 36 e0 20 39 f8 4c 6f 1b 7e 0c 0a 37

0110 f7 d0 05 a0 34 b1 ae 2c 5e 31 51 c6 fb 92 6c f0

0120 a2 1e 4c c1 1e f8 a5 7e 60 86 21 de b2 8c a5 ea

0130 ca 56 c8 78 85 2b 7d 1b 91 81 9c b6 4a 5d f8 b7

0140 82 37 bb d1 46 2c a6 34 8f 47 39 08 27 09 38 20

0150 d4 4d 0a 52 3e 49 99 9e f1 41 3c 5c 69 81 68 5a

0160 0c a8 f2 4e f0 2a c8 c2 61 3d 2b b4 73 59 27 b4

0170 47 82 92 2a f3 29 27 be 2e 9e 17 0a f7 a0 0e 13

0180 11 ce 6d db c4 03 c4 59 a0 3b 54 3e 21 be 5c b1

0190 02 2b e5 67 c5 29 b8 19 d7 3e 24 4c 70 d1 41 84

01a0 89 3b 36 6a 44 4e f0 6b 9d 75 d1 6c b3 57 58 65

01b0 12 8d ad c9 0d 47 88 38 a8 72 36 94 38 7c 15 1b

01c0 06 dd db 12 8f 86 40 5d fc 51 11 d5 0b f0 15 25

01d0 ae 84 6a 03 29 3c 82 44 1d 11 91 21 f0 8b 2c cf

01e0 72 b3 17 47 21 79 9b 92 a0 d3 c5 10 13 3f e2 48

01f0 66 77 52 02 ca 6c 94 64 9a 8f 68 95 bd c8 28 a5

0200 2b f6 c4 e5 81 82 a5 82 9f ec 70 46 5e 25 ae b3

0210 e5 a4 d9 91 1e 7a 5c cf 7b 06 24 95 02 8e 2a b6

0220 42 dc 44 93 b3 c8 40 e4 f1 b4 b2 01 5e 63 26 69

0230 70 c4 a6 08 95 cd 55 ea 3f 27 a6 78 62 02 0e 6d

0240 36 b2 76 16 02 33 65 36 1b e0 05 e5 35 ab 97 66

0250 bf 07 e5 48 04 20 94 a3 1a 87 11 55 1f 0e 47 a1

0260 f2 81 48 e0 25 78 c6 92 4d 3c c4 46 48 73 98 fb

0270 67 49 3c f5 00 32 c6 72 d7 28 17 5d 20 32 d1 1a

IP packet

IP source and destination addresses

Protocol type

Left Pane: Encapsulation

TCP segment

Source and
destination port
numbers

Reassemble

TLS handshake
for HTTPS

No.	Time	Source	Destination	Protocol	Length	Info
276	1.360874	128.198.212.72	151.101.69.164	TCP	1436	57840 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=1370 TSval=1405589587 TSecr=45
277	1.360884	128.198.212.72	151.101.69.164	TCP	66	57841 → 443 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=220190731 TSecr=290407
278	1.360887	128.198.212.72	151.101.69.164	TLSv1.3	703	Client Hello (SNI=g1.nyt.com)
Frame 278: 703 bytes on wire (5624 bits), 703 bytes captured (5624 bits) on interface en0, id 0						
Ethernet II, Src: 2e:de:b7:57:c6:7d (2e:de:b7:57:c6:7d), Dst: All-HSRP-routers_01 (00:00:0c:07:ac:01)						
Internet Protocol Version 4, Src: 128.198.212.72, Dst: 151.101.69.164						
Transmission Control Protocol, Src Port: 57840, Dst Port: 443, Seq: 1371, Ack: 1, Len: 637						
Source Port: 57840						
Destination Port: 443						
[Stream index: 1]						
[Stream Packet Number: 5]						
[Conversation completeness: Incomplete, DATA (15)]						
[TCP Segment Len: 637]						
Sequence Number: 1371 (relative sequence number)						
Sequence Number (raw): 2445401441						
[Next Sequence Number: 2008 (relative sequence number)]						
Acknowledgment Number: 1 (relative ack number)						
Acknowledgment number (raw): 563439214						
1000 = Header Length: 32 bytes (8)						
Flags: 0x018 (PSH, ACK)						
Window: 2055						
[Calculated window size: 131520]						
[Window size scaling factor: 64]						
Checksum: 0x588c [unverified]						
[Checksum Status: Unverified]						
Urgent Pointer: 0						
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps						
[Timestamps]						
[SEQ/ACK analysis]						
TCP payload (637 bytes)						
TCP segment data (637 bytes)						
[2 Reassembled TCP Segments (2007 bytes): #276(1370), #278(637)]						
[Frame: 276, payload: 0-1369 (1370 bytes)]						
[Frame: 278, payload: 1370-2006 (637 bytes)]						
[Segment count: 2]						
[Reassembled TCP length: 2007]						
[Reassembled TCP Data [1: 16030107d2010007c003030d00cb56e1cd59d56d4648ead6142b87c692c8a0d66dae2d5b6c						
Transport Layer Security						
TLSv1.3 Record Layer: Handshake Protocol: Client Hello						
Content Type: Handshake (22)						
Version: TLS 1.0 (0x0301)						
Length: 2002						
Handshake Protocol: Client Hello						
Handshake Type: Client Hello (1)						

0020 d6 6d ae 2d 5b c6 24 d5 bb ad cf 20 fa
0030 08 de 48 12 c4 9c 87 b1 0e ef af 7a f7
0040 b6 56 a5 ce 69 61 c0 b7 29 6e 15 e4 00
0050 13 01 13 02 13 03 c0 2b c0 2f c0 2c c0
0060 cc a8 c0 13 c0 14 00 9c 00 9d 00 2f 00
0070 07 65 8a 8a 00 00 00 0a 00 c0 00 2a 0a
0080 00 1d 00 17 00 18 ff 01 00 01 00 00 00
0090 00 00 00 0a 67 31 2e 6e 79 74 2e 63 6f
00a0 00 00 00 33 04 ef 04 ed 2a 2a 00 01 00
00b0 c0 00 65 29 e6 be 96 8f 5c 9d 35 8d 42
00c0 c3 66 df 0b 46 01 f2 80 33 ea d1 7f 4e
00d0 75 08 d3 0f fe 16 bd fa dc aa ce 76 1e
00e0 82 70 16 96 9b 63 ff c1 5f e5 38 cd ca
00f0 76 3e 02 dc cc ae 66 43 71 c1 9e 75 71
0100 17 2c 56 56 36 e0 20 39 f8 4c 6f 1b 7e
0110 f7 d0 05 a0 34 b1 ae 2c 5e 31 51 c6 bf
0120 a2 1e 4c c1 1e f8 a5 7e 60 86 21 de b2
0130 ca 56 c8 78 85 2b 7d 1b 91 81 9c b6 4a
0140 82 37 bb d1 46 2c a6 34 8f 47 39 08 27
0150 d4 4d 0a 52 3e 49 99 9e f1 41 3c c5 69
0160 0c a8 f2 4e f0 2a c8 c2 61 3d 2b b4 73
0170 47 82 92 2a f3 29 27 eb 2e 9e 17 0a f7
0180 11 ce 6d db c4 03 c4 59 a0 3b 54 3e 21
0190 02 2b e5 67 c5 29 b8 19 d7 3e 24 4c 70
01a0 89 3b 36 6a 44 4e f0 6b 9d 75 d1 6c b3
01b0 12 8d ad c9 0d 47 88 38 a8 72 36 94 38
01c0 06 dd db 12 8f 86 40 5d fc 51 11 d5 0b
01d0 a6 84 6a 03 29 3c 82 44 1d 11 91 21 f0
01e0 72 b3 17 47 21 79 9b 92 a0 d3 c5 10 13
01f0 66 77 52 02 ca 6c 94 64 9a 8f 68 95 bd
0200 2b f6 c4 e5 81 82 a5 82 9f ec 70 46 5e
0210 e5 a4 d9 91 1e 7a 5c cf 7b 06 24 95 02
0220 42 dc 44 93 b3 c8 40 e4 f1 b4 b2 01 5e
0230 70 c4 a6 08 95 cd 55 ea 3f 27 a6 78 62
0240 36 b2 76 16 02 33 65 36 1b 0e 05 e5 35
0250 bf 07 e5 48 04 20 94 a3 1a 87 11 55 1f
0260 f2 81 48 e0 25 78 c6 92 4d 3c c4 46 48
0270 67 49 3c f5 00 32 c6 72 d2 28 17 5d 29
0280 77 19 f6 4c 3b b6 5c 6c 51 a5 10 6a b8
0290 f8 97 39 56 a6 87 24 2a 1b d1 f3 c1 b2
02a0 50 b9 28 3a 70 a5 c5 89 6a 6b 65 58 d9
02b0 65 b8 c7 1f c2 14 3c c8 e1 45 3d 1a 17
02c0 a5 28 13 cd 2b 40 ad 33 15 52 a7 a0
02d0 5b cc 03 31 90 a8 a8 aa 08 c2 7f b2 03
02e0 7d 4c 7b 30 3f 16 86 01 41 21 fc 17 a2
02f0 0a f4 0a 33 e3 2d b5 b9 9a 39 e4 02 06
0300 92 0b a5 79 80 b7 3b a7 ea 36 61 1d b3
0310 73 97 dc a9 47 eb c3 15 5b 67 57 60 73

Frame (703 bytes) Reassembled TCP (2007 bytes)

Network Attacks

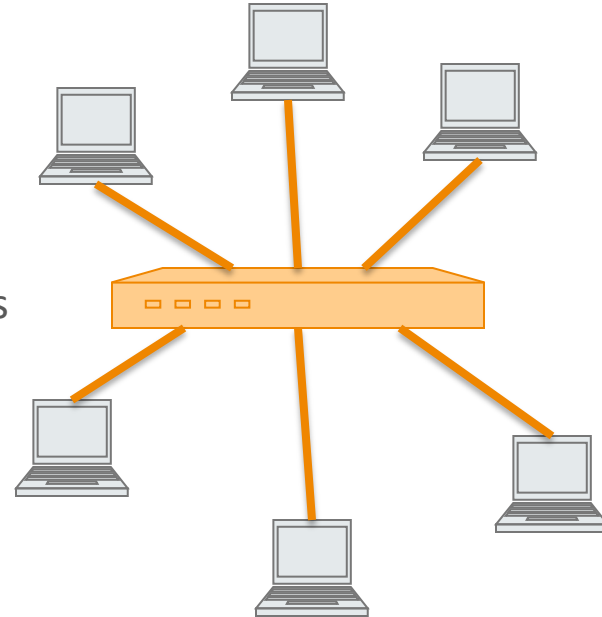
- MAC Spoofing, ARP Spoofing
- IP Spoofing
- Denial of Service
- DNS Cache Poisoning

MAC Addresses

- Most network interfaces come with a predefined MAC ([Media Access Control](#)) address
- A MAC address is a 48-bit number usually represented in hex
 - E.g., 00-1A-92-D4-BF-86
- The first three octets of any MAC address are IEEE-assigned Organizationally Unique Identifiers
 - E.g., Cisco 00-1A-A1, D-Link 00-1B-11, ASUSTek 00-1A-92
- The next three can be assigned by organizations as they please, with uniqueness being the only constraint
 - Organizations can utilize MAC addresses to identify computers on their network
 - MAC address can be reconfigured by network interface driver software

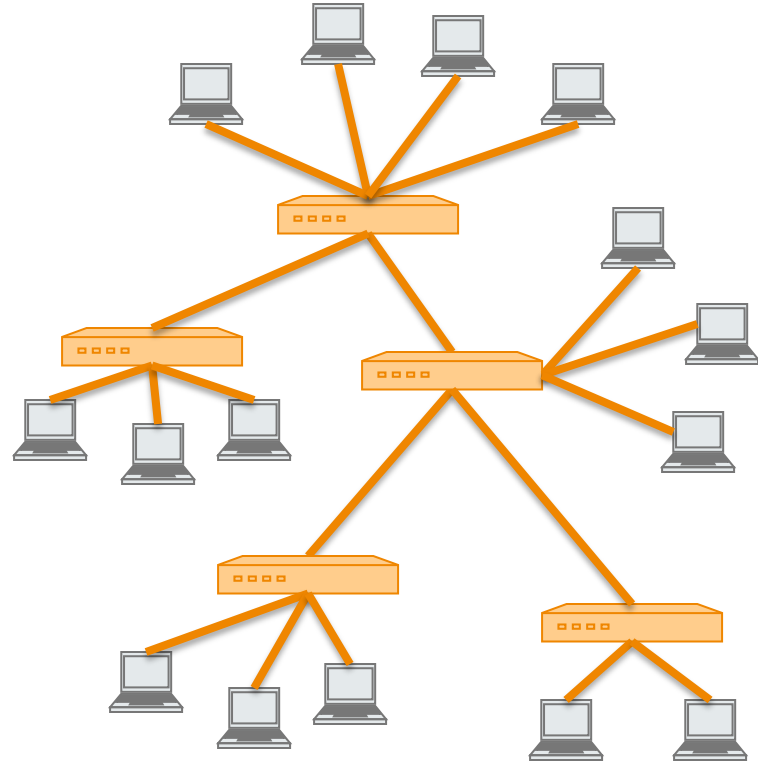
Switch

- A **switch** is a common network device
 - Operates at the link layer
 - Has multiple ports, each connected to a computer
 - Connects computers in an organization's internal Lan (local area network)
- Operation of a switch
 - Learn the MAC address of each computer connected to it
 - Forward frames **only** to the destination computer



Combining Switches

- Switches can be arranged into a **tree**
- Each port learns the MAC addresses of the machines in the segment (subtree) connected to it
- Fragments to unknown MAC addresses are **broadcast**
- Frames to MAC addresses in the same segment as the sender are ignored



MAC Spoofing

- A **switch** can be configured to provide service only to machines with specific MAC addresses
- Allowed MAC addresses need to be **registered** with a network administrator
- A **MAC spoofing attack impersonates** another machine
 - Find out MAC address of target machine
 - Reconfigure MAC address of rogue machine
 - Turn off or unplug target machine
- **Countermeasures**
 - Block port of switch when machine is turned off or unplugged
 - Disable duplicate MAC addresses

Viewing and Changing MAC Addresses

- Viewing the MAC addresses of the interfaces of a machine
 - Linux: `ifconfig`
 - Windows: `ipconfig /all`
- Changing a MAC address in Linux
 - Stop the networking service: `/etc/init.d/network stop`
 - Change the MAC address: `ifconfig eth0 hw ether <MAC-address>`
 - Start the networking service: `/etc/init.d/network start`
- Changing a MAC address in Windows
 - Open the Network Connections applet
 - Access the properties for the network interface
 - Click “Configure ...”
 - In the advanced tab, change the network address to the desired value
- Changing a MAC address requires administrator privileges

ARP

- The **address resolution protocol (ARP)** is a link-layer protocol that connects the network layer to the link layer by converting IP addresses to MAC addresses
- ARP works by **broadcasting** requests and **caching** responses for future use
- The protocol begins with a computer broadcasting a message of the form
who has <IP address1> tell <IP address2>
- Then the machine with **<IP address1>** responds the requestor with an ARP reply as
<IP address1> is <MAC address>
- The Linux and Windows command **arp - a** displays the ARP table

Internet Address	Physical Address	Type
128.148.31.1	00-00-0c-07-ac-00	dynamic
128.148.31.15	00-0c-76-b2-d7-1d	dynamic
128.148.31.71	00-0c-76-b2-d0-d2	dynamic
128.148.31.75	00-0c-76-b2-d7-1d	dynamic
128.148.31.102	00-22-0c-a3-e4-00	dynamic
128.148.31.137	00-1d-92-b6-f1-a9	dynamic

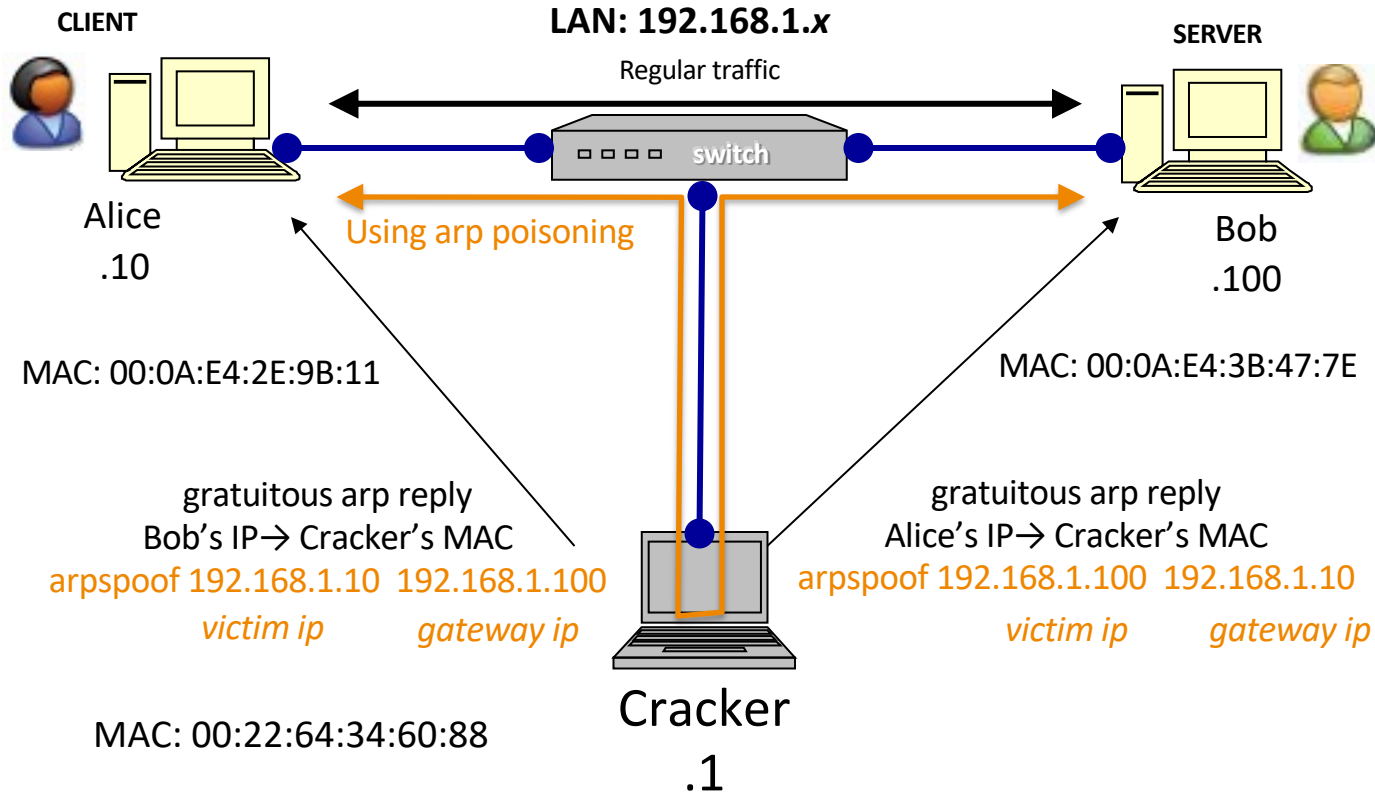
ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not **tracked**
- ARP announcements are not **authenticated**
- Machines trust each other
- A rogue machine can **spoof** other machines

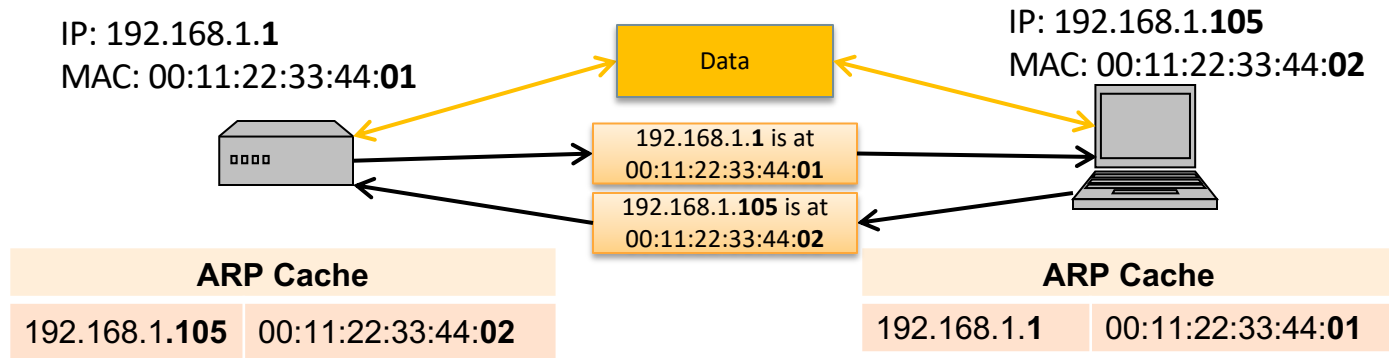
ARP Spoofing (ARP Poisoning)

- According to the standard, almost all ARP implementations are **stateless**
 - An ARP cache updates every time that it receives an ARP reply... even if it did not send any ARP request!
 - It is possible to “poison” an ARP cache by sending **gratuitous ARP replies**
 - Using static entries solves the problem but it is almost impossible to manage!

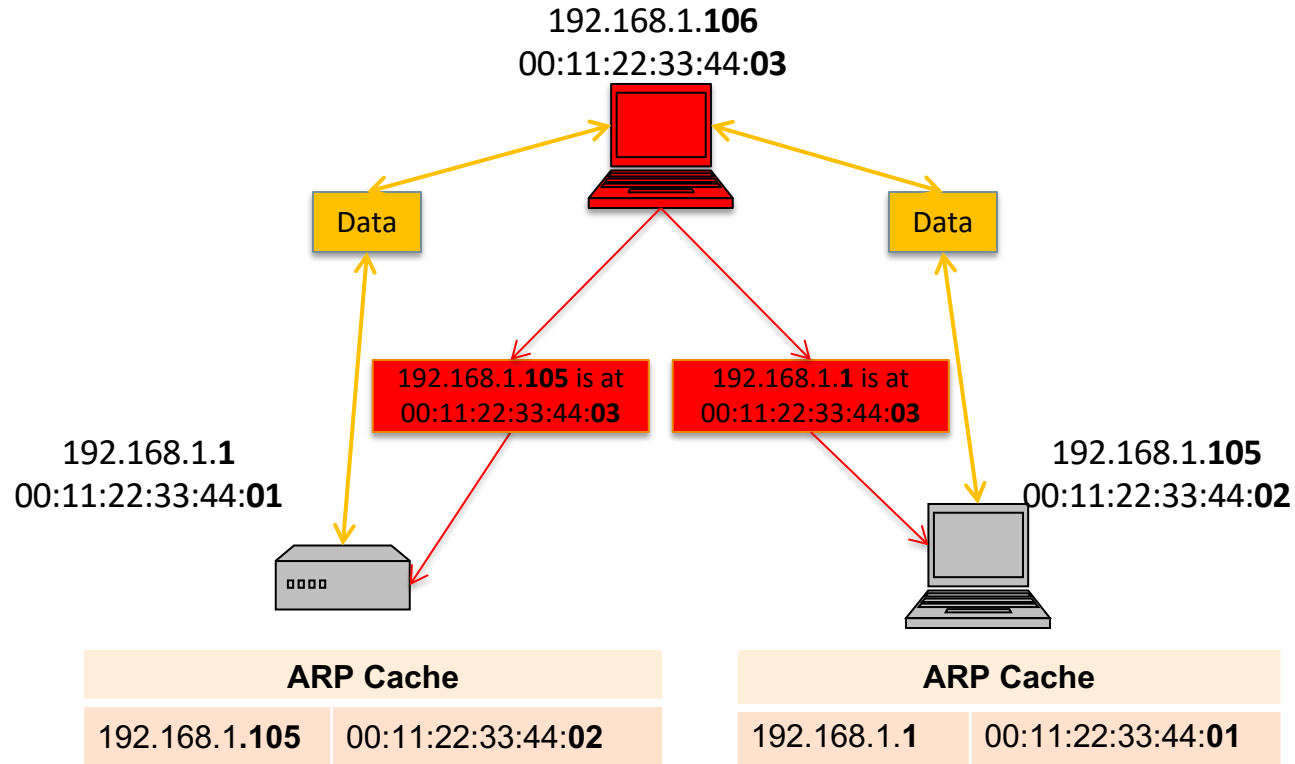
Example 1: ARP Spoofing



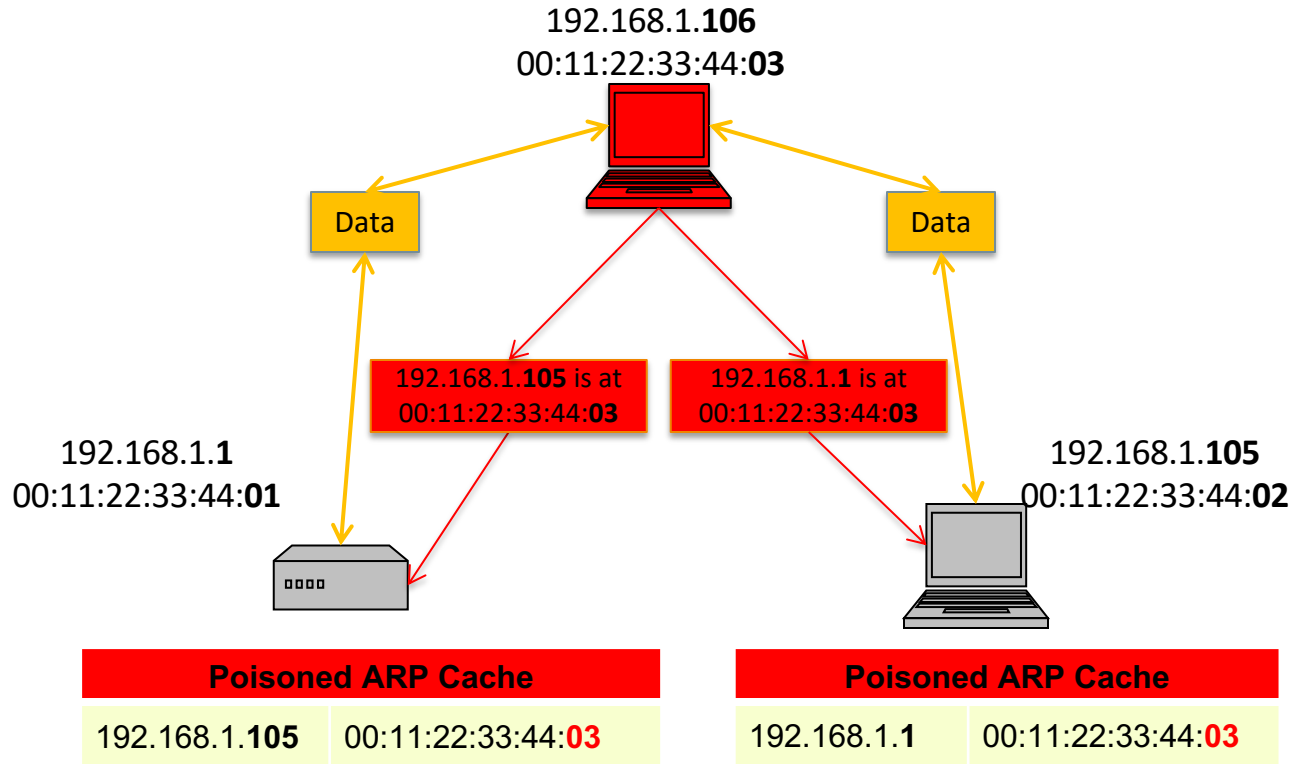
ARP Caches



Example 2: Poisoned ARP Caches



Example 2: Poisoned ARP Caches



ARP Spoofing (or ARP poisoning)

- Send **fake** ARP messages to an Ethernet LAN (no authentication)
 - this causes other machines to associate IP addresses with attacker's MAC
- **Defenses**
 - static ARP table
 - DHCP snooping (access control based on IP, MAC, and port)
 - detection: Arpwatch, reverse ARP

Next

Network Attacks

- MAC Spoofing, ARP Spoofing
- IP spoofing
- Denial of service
- DNS cache poisoning