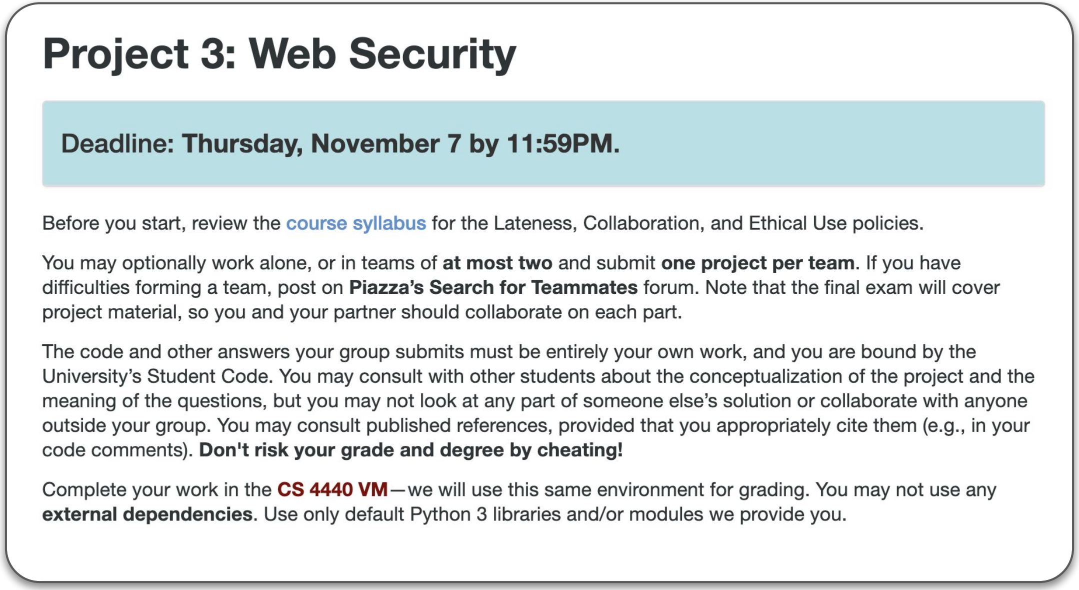
Week 10: Lecture A

Application-layer Network Attacks

Tuesday, October 29, 2024

Announcements

- **Project 3: WebSec** released
  - **Deadline:** Thursday, November 7th by 11:59PM (**next week**)



Project 3: Web Security

Deadline: **Thursday, November 7 by 11:59PM.**

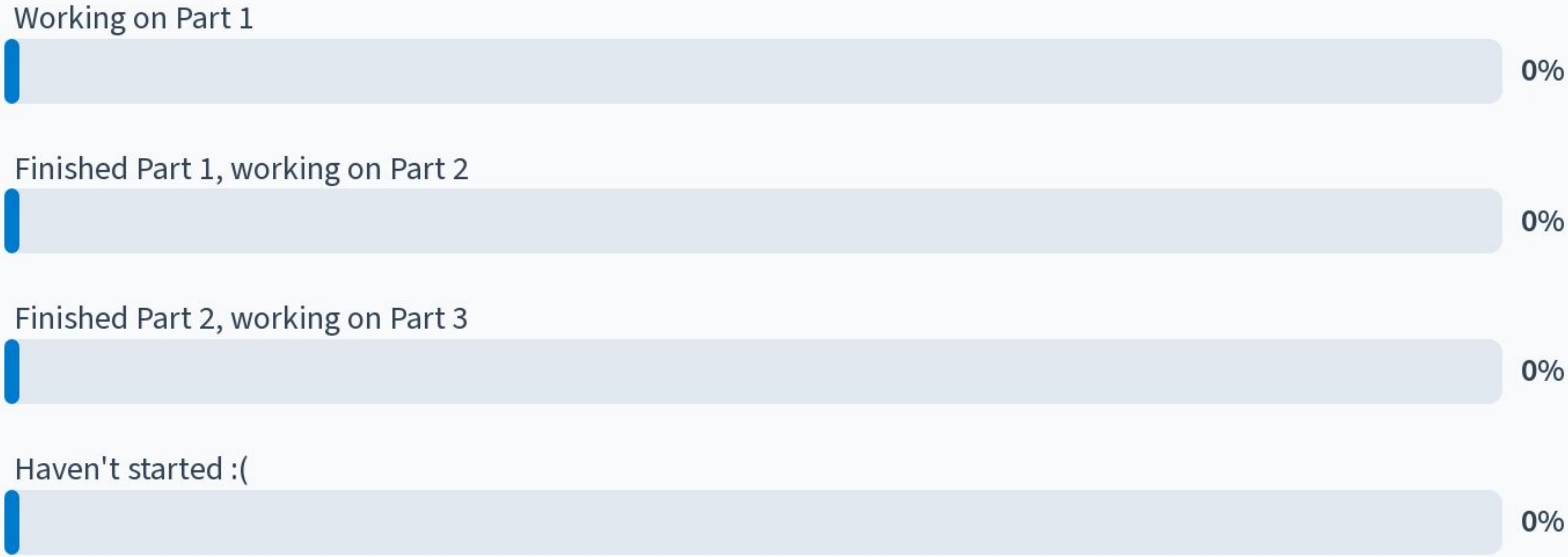
Before you start, review the course syllabus for the Lateness, Collaboration, and Ethical Use policies.

You may optionally work alone, or in teams of **at most two** and submit **one project per team**. If you have difficulties forming a team, post on **Piazza's Search for Teammates** forum. Note that the final exam will cover project material, so you and your partner should collaborate on each part.

The code and other answers your group submits must be entirely your own work, and you are bound by the University's Student Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with anyone outside your group. You may consult published references, provided that you appropriately cite them (e.g., in your code comments). **Don't risk your grade and degree by cheating!**

Complete your work in the **CS 4440 VM**—we will use this same environment for grading. You may not use any **external dependencies**. Use only default Python 3 libraries and/or modules we provide you.

Project 3 progress



Announcements

utahsec

See Discord for meeting info!

www.utahsec.com

Questions?

Last time on CS 4440…

Introduction to Networking
The Physical, Link, Network,
Transport, and Application Layers

What is the Internet?

- **What is it?**
  - How you trash-talk players in COD game lobbies
  - How Wall Street trades shares faster than you
  - How the CS 4440 website is distributed to you

VOICE CHAT ON

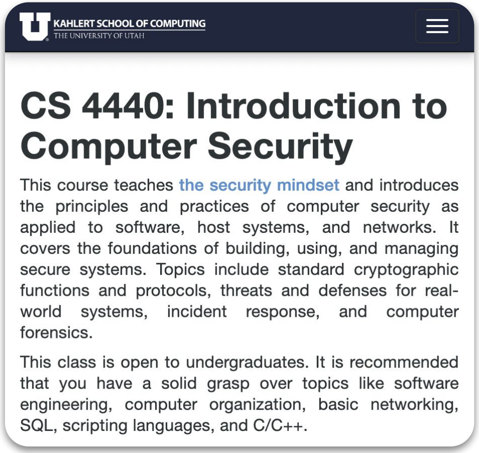
GAME VOICE CHANNEL ALL LOBBY

VOICE CHAT DEVICE

VOICE CHAT RECORDING MODE

VOICE CHAT (PUSH TO TALK)

OPEN MIC RECORDING THRESHOLD 70



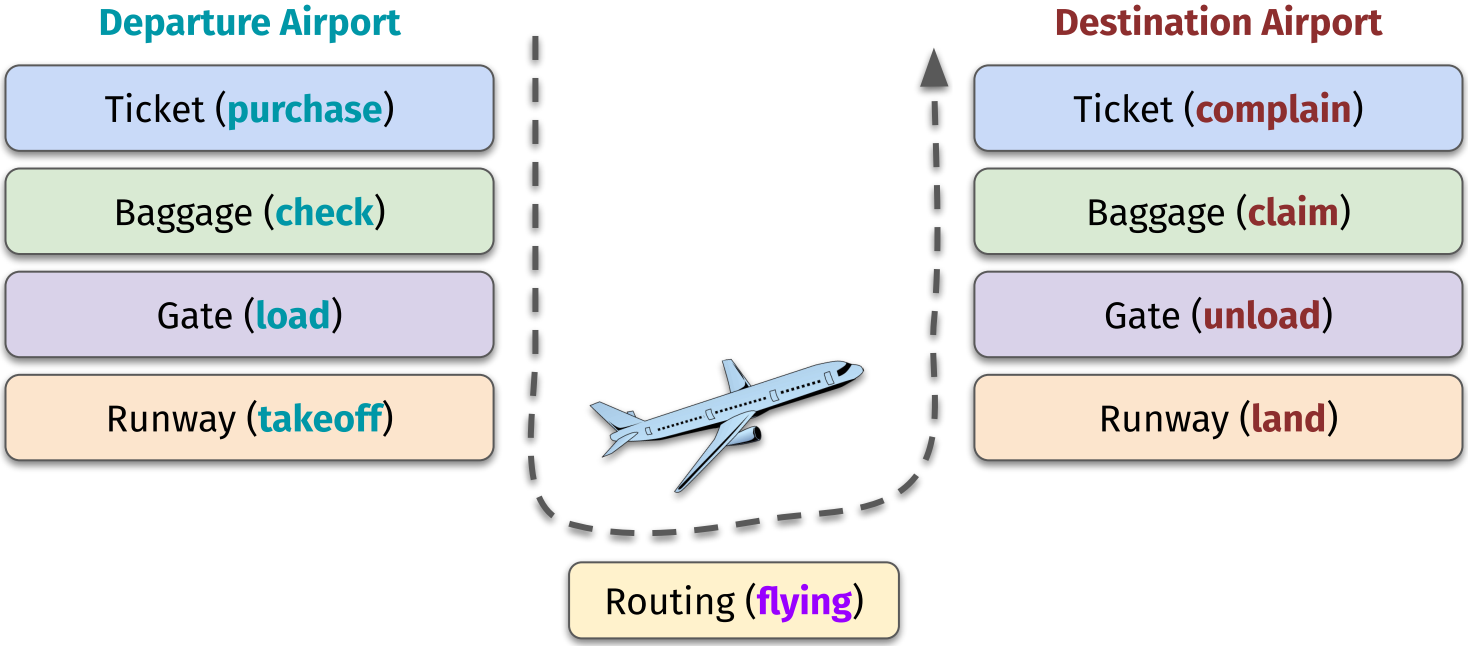
CS 4440: Introduction to Computer Security

This course teaches the security mindset and introduces the principles and practices of computer security as applied to software, host systems, and networks. It covers the foundations of building, using, and managing secure systems. Topics include standard cryptographic functions and protocols, threats and defenses for real-world systems, incident response, and computer forensics.

This class is open to undergraduates. It is recommended that you have a solid grasp over topics like software engineering, computer organization, basic networking, SQL, scripting languages, and C/C++.

What really is the internet?

- **A group of layers**—each implementing a **service**



Departure Airport

Ticket (**purchase**)

Baggage (**check**)

Gate (**load**)

Runway (**takeoff**)

Destination Airport

Ticket (**complain**)

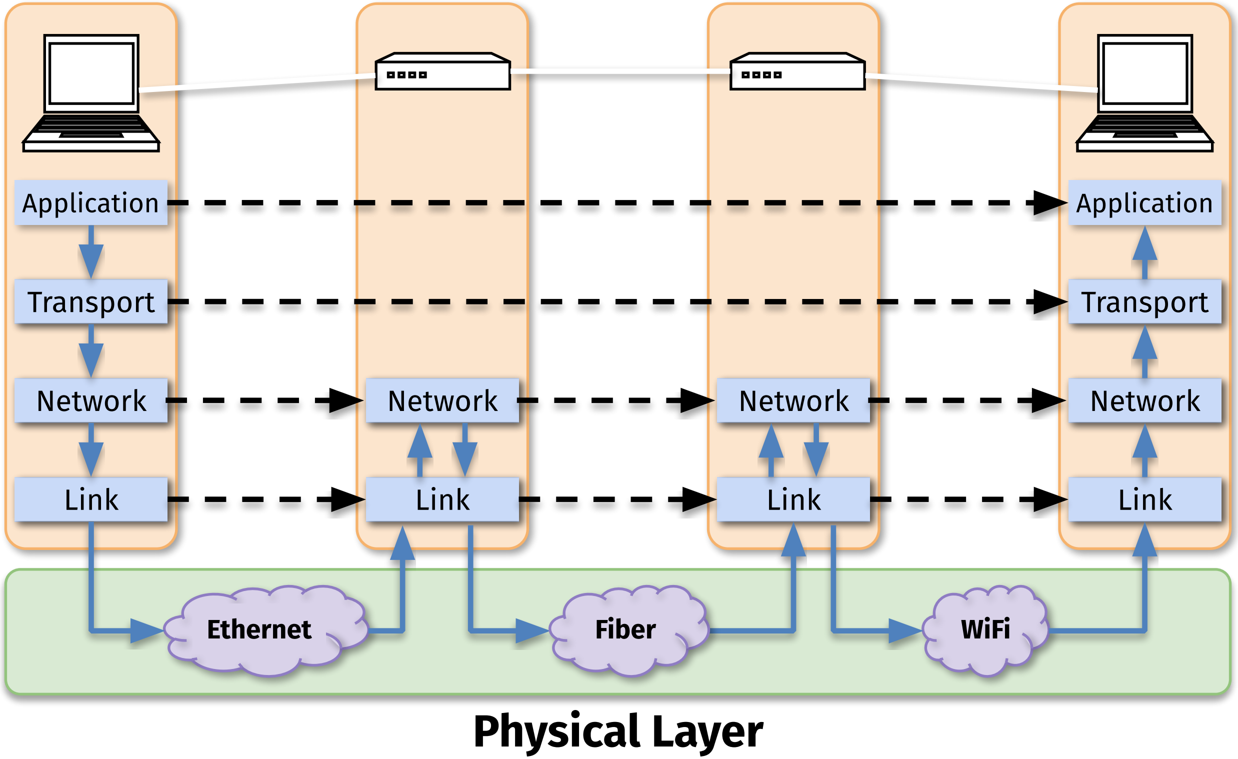
Baggage (**claim**)

Gate (**unload**)

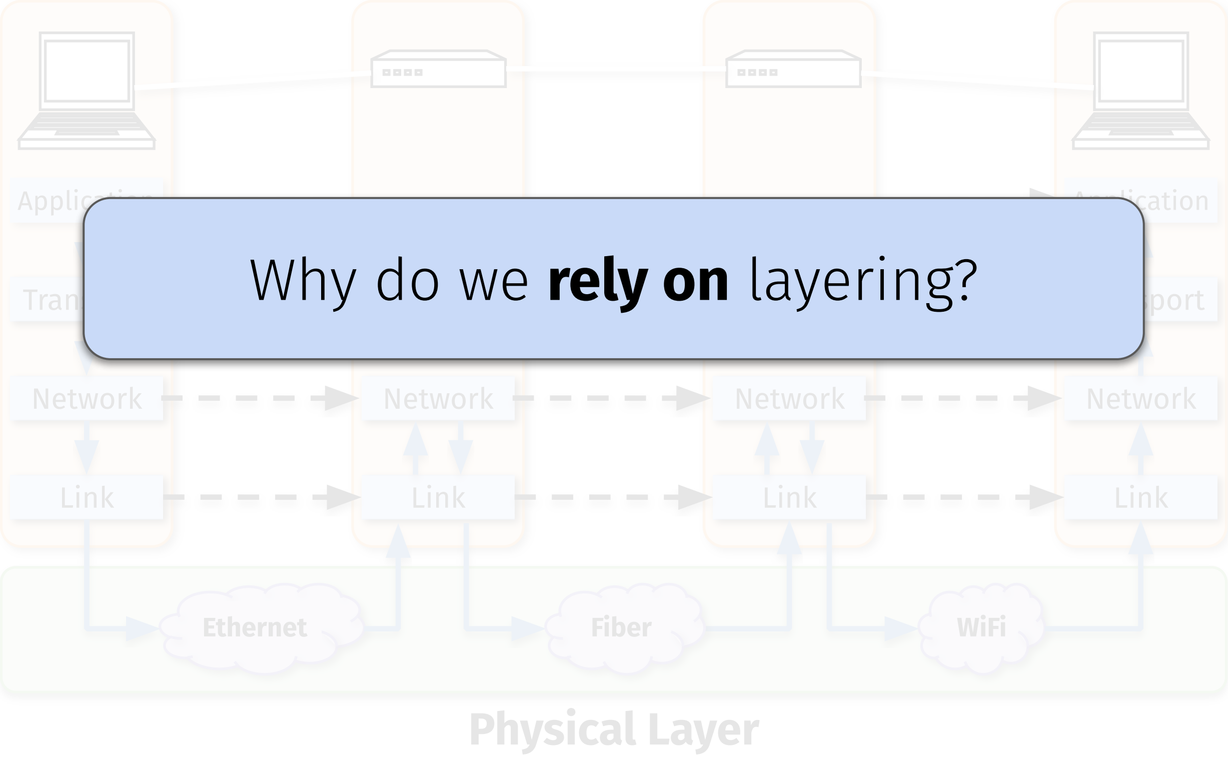
Runway (**land**)

Routing (**flying**)

The 5-layer Internet

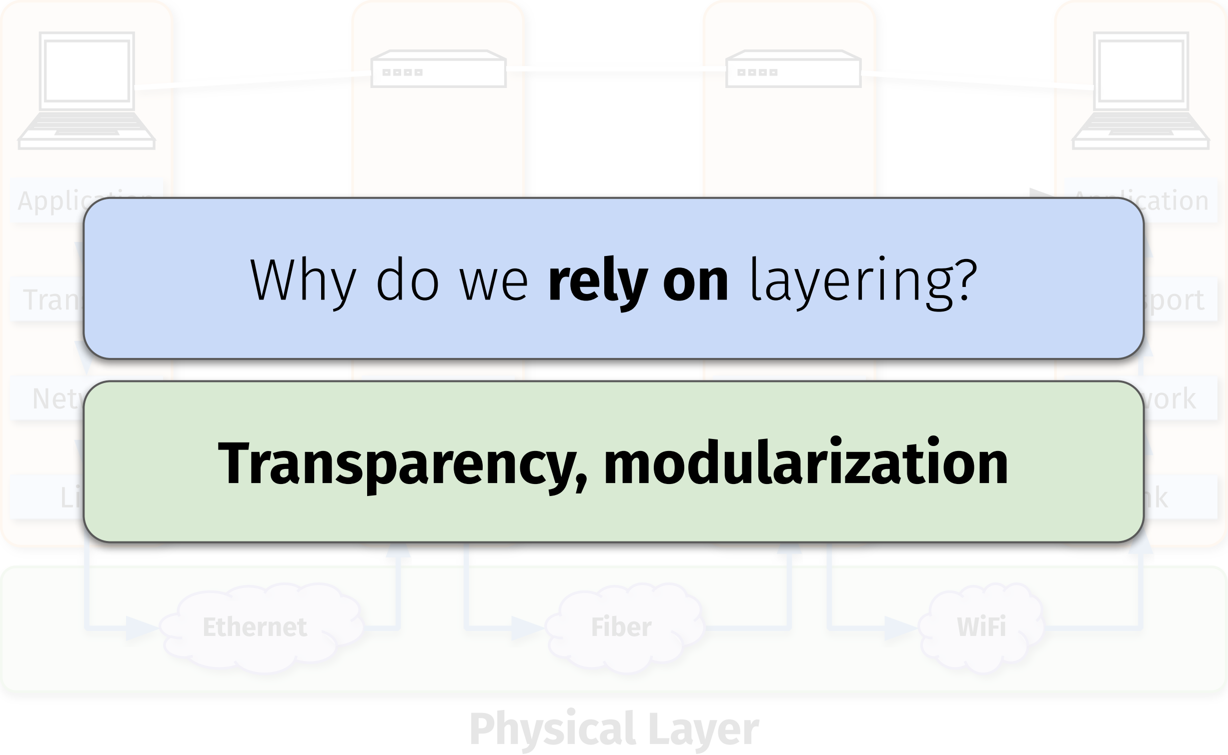


The 5-layer Internet



Why do we **rely on** layering?

The 5-layer Internet

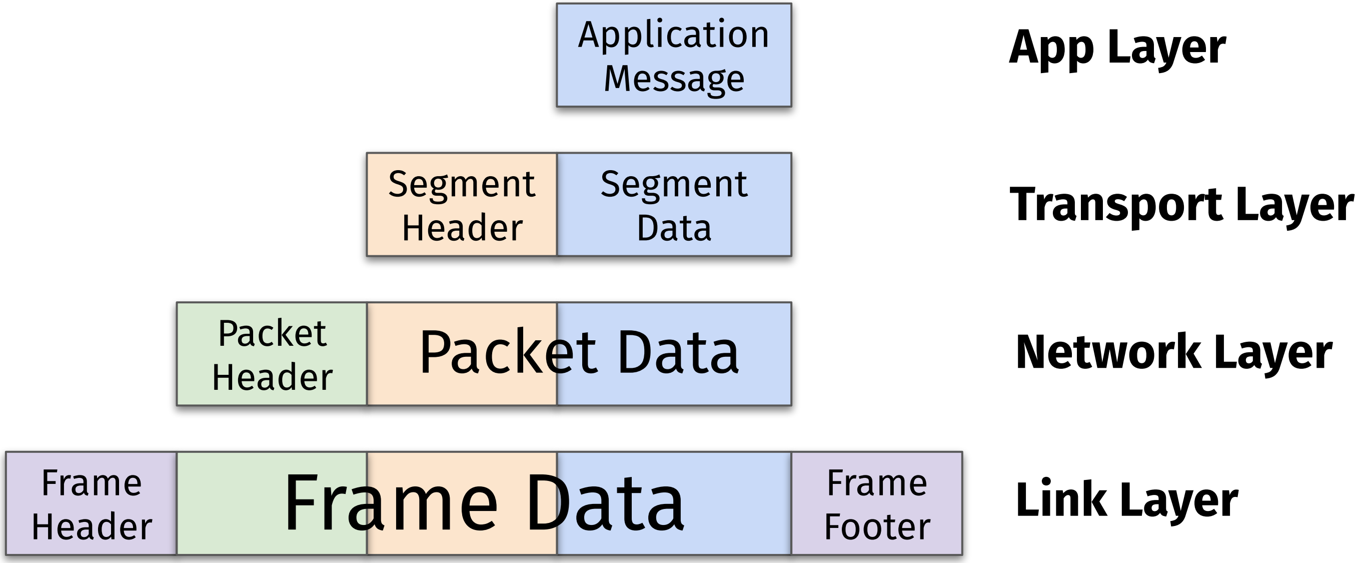


Why do we **rely on** layering?

Transparency, modularization

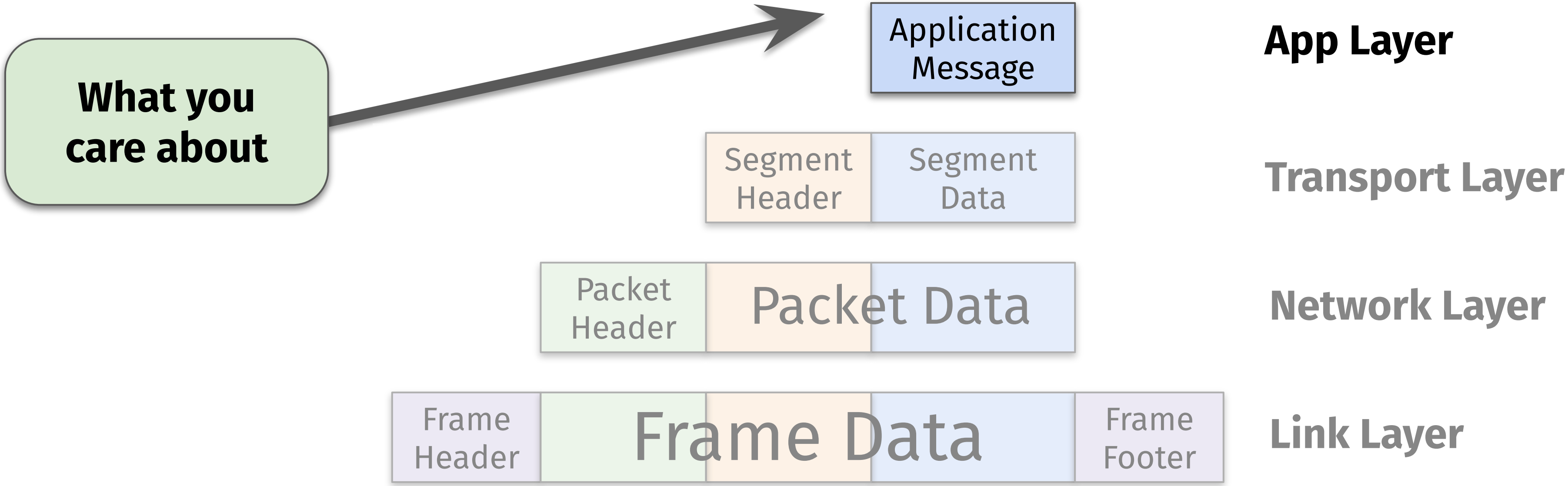
Internet Packet Encapsulation

- How packets are generated and sent



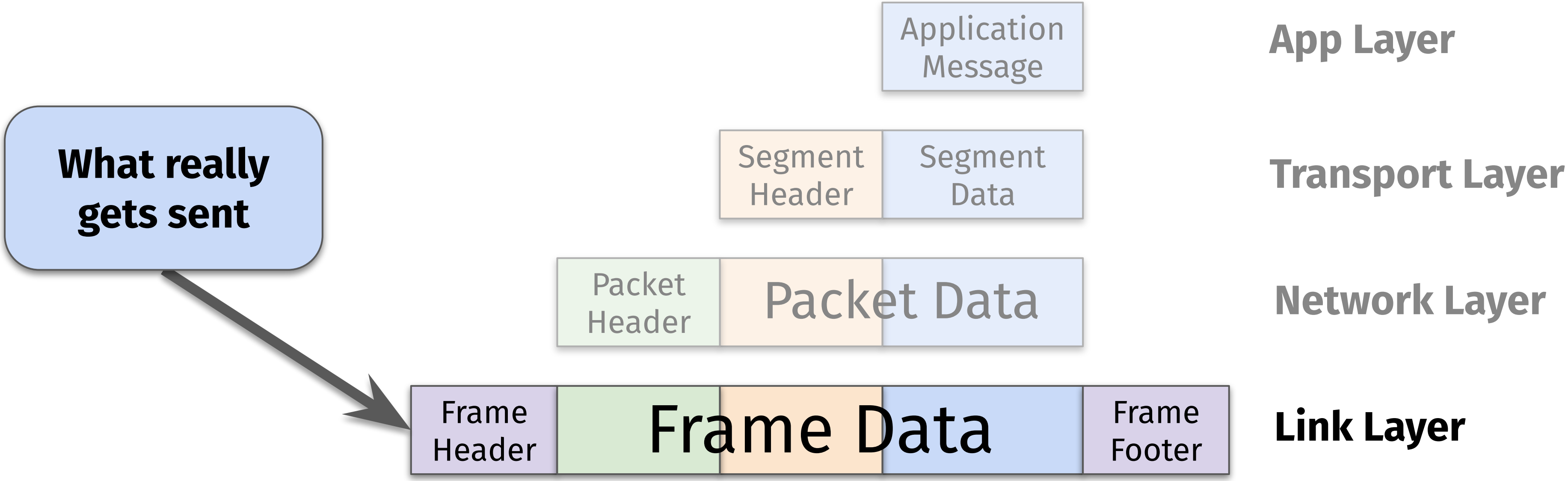
Internet Packet Encapsulation

- How packets are generated and sent



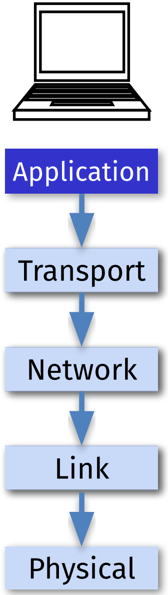
Internet Packet Encapsulation

- How packets are generated and sent



The Application Layer

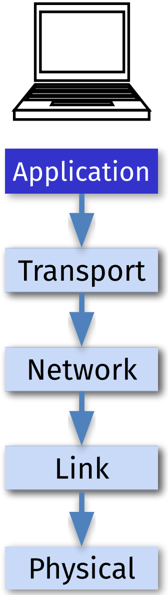
- **What is it?**
  - ???



The Application Layer

- **What is it?**
  - The **top-most layer** in the 5-layer network model
  - Where applications send and receive messages

- What does it **define**? **Application protocols**
  - **???**

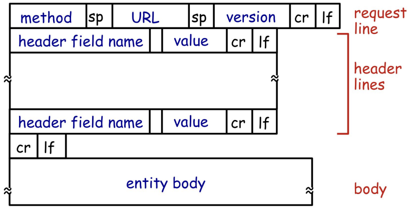


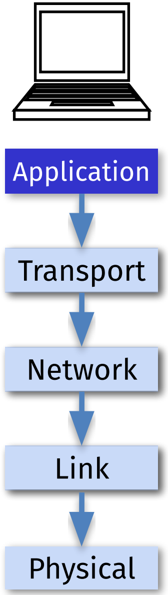
The Application Layer

- **What is it?**
  - The **top-most layer** in the 5-layer network model
  - Where applications send and receive messages

- What does it **define**? **Application protocols**
  - Message **types**
    - What is the purpose of this message?
  - Message **syntax**
    - How should this message be structured?
  - Message **semantics**
    - What does each message field really mean?
  - Rules for **sending/receiving** messages
    - When/how should this application respond?

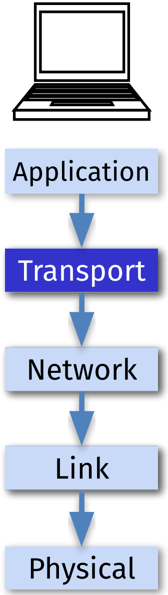
http:// FTP
SMTP
McAfee
zoom




Application
Transport
Network
Link
Physical

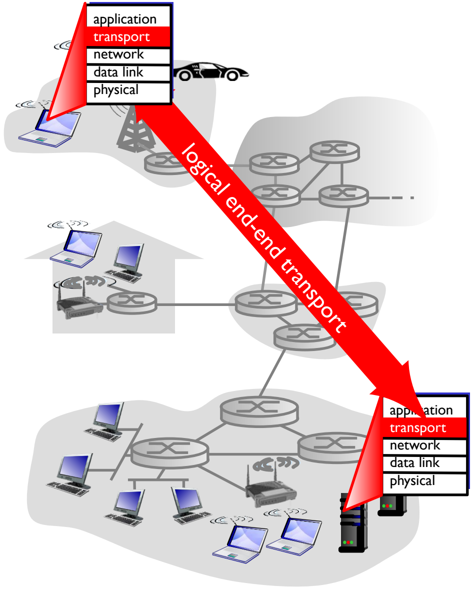
The Transport Layer

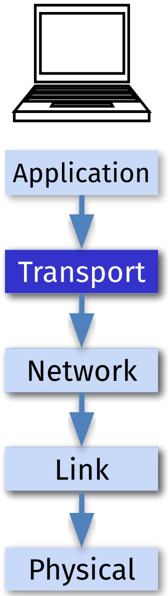
- **What is it?**
  - **???**



The Transport Layer

- **What is it?**
  - The **second layer** in the 5-layer network model
  - Communication between apps on different hosts
- What are its two main **protocols**?
  - ???




Application

Transport

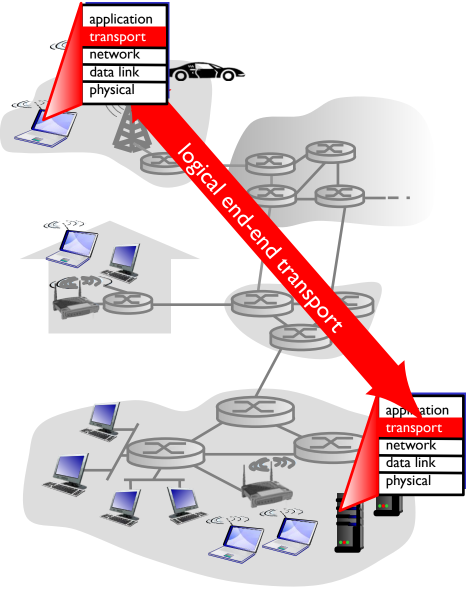
Network

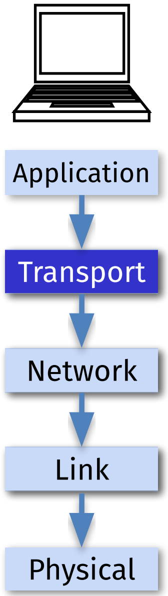
Link

Physical

The Transport Layer

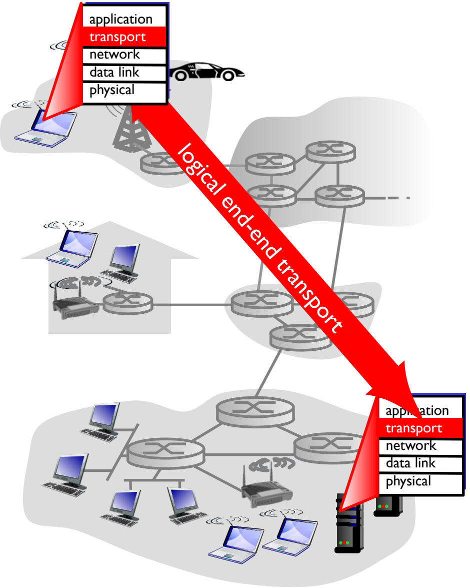
- **What is it?**
  - The **second layer** in the 5-layer network model
  - Communication between apps on different hosts
- What are its two main **protocols**? **TCP**, **UDP**
  - **TCP**—Transmission Control Protocol
    - **Characteristics: slow/complex** but **reliable**
  - **UDP**—User Datagram Protocol
    - **Characteristics: fast/simple** but **unreliable**
- What are **ideal use cases** for TCP and UDP?
  - **???**

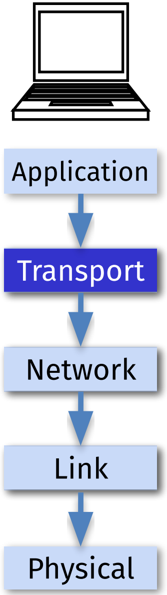




The Transport Layer

- **What is it?**
  - The **second layer** in the 5-layer network model
  - Communication between apps on different hosts
- What are its two main **protocols**? TCP, UDP
  - TCP—Transmission Control Protocol
    - **Characteristics: slow/complex** but **reliable**
  - UDP—User Datagram Protocol
    - **Characteristics: fast/simple** but **unreliable**
- What are **ideal use cases** for TCP and UDP?
  - TCP: reliability matters (file transfer, SSH, e-mail)
  - UDP: speed matters (video calls, gaming, livestream)




Application

Transport

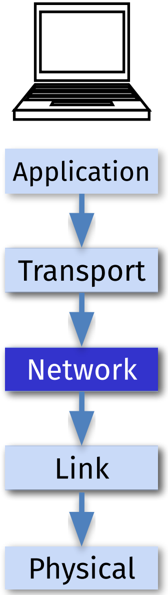
Network

Link

Physical

The Network Layer

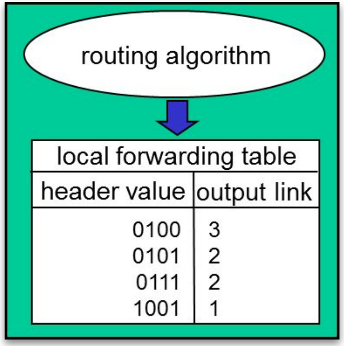
- **What is it?**
  - **???**



The Network Layer

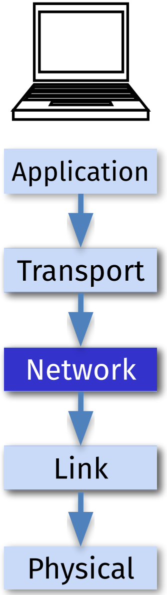
- **What is it?**
  - The **third layer** in the 5-layer network model
  - Sends data from host on one network to another

- What are its **two functions**?
  - ???



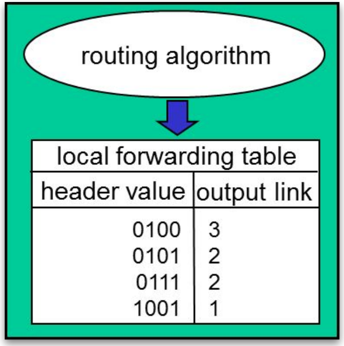
local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

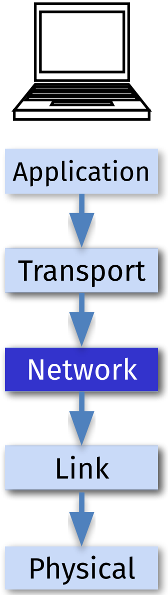


The Network Layer

- **What is it?**
  - The **third layer** in the 5-layer network model
  - Sends data from host on one network to another
- What are its **two functions**? **Routing**, **forwarding**
  - **Routing:** find shortest possible path to send a packet
  - **Forwarding:** sending packets on to the next hop
- What is its **addressing** based on?
  - **???**

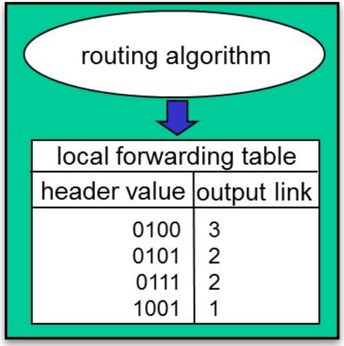


local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1



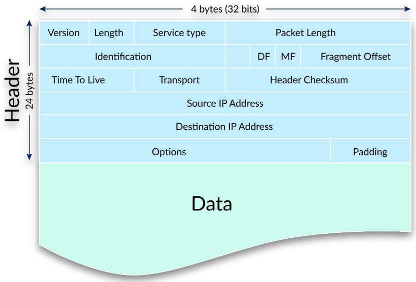
The Network Layer

- **What is it?**
  - The **third layer** in the 5-layer network model
  - Sends data from host on one network to another
- What are its **two functions**? **Routing**, **forwarding**
  - **Routing:** find shortest possible path to send a packet
  - **Forwarding:** sending packets on to the next hop
- What is its **addressing** based on?
  - IP addresses—a logical address
    - Network-internal IP assigned by your router
    - Public IP assigned by Internet Service Provider

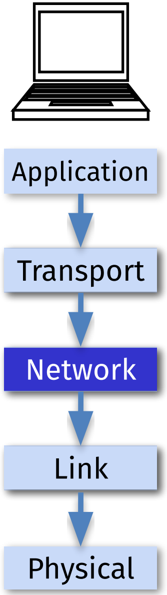


local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

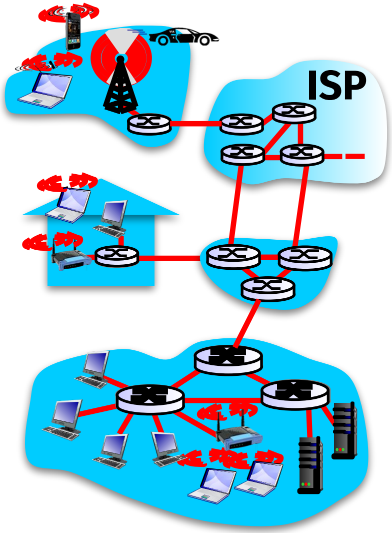


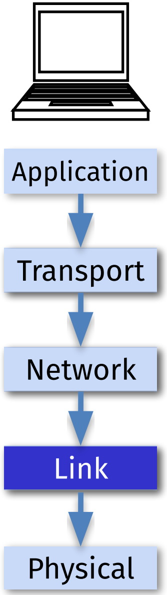
Version	Length	Service type	Packet Length		
Identification			DF	MF	Fragment Offset
Time To Live		Transport	Header Checksum		
Source IP Address					
Destination IP Address					
Options					Padding



The Data Link Layer

- **What is it?**
  - ???

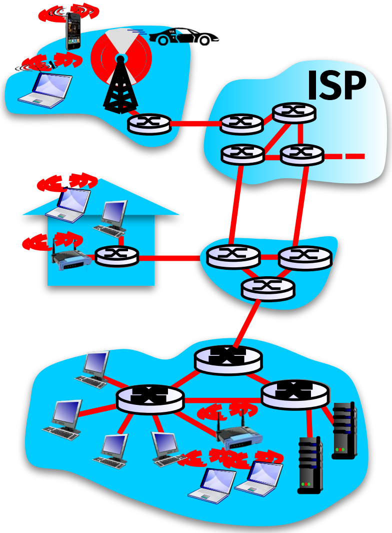


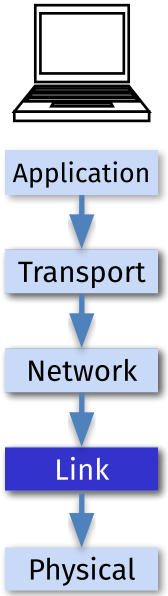


The Data Link Layer

- **What is it?**
  - The **fourth layer** in the 5-layer network model
  - Responsible for the node-to-node delivery of data

- What are **“nodes”**?
  - **???**



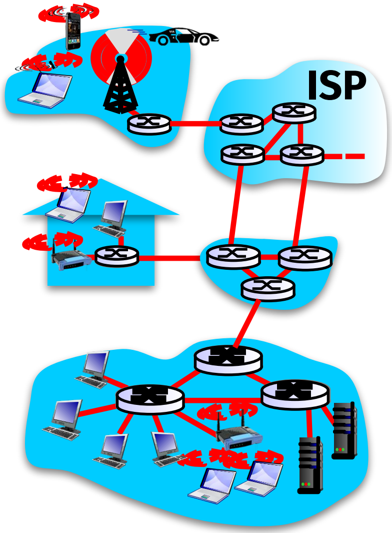


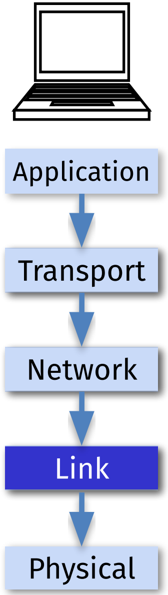
The Data Link Layer

- **What is it?**
  - The **fourth layer** in the 5-layer network model
  - Responsible for the node-to-node delivery of data

- What are “**nodes**”? **Hosts**, **switches**
  - **Hosts:** the physical devices within a network
  - **Switches:** interface to all hosts on the network

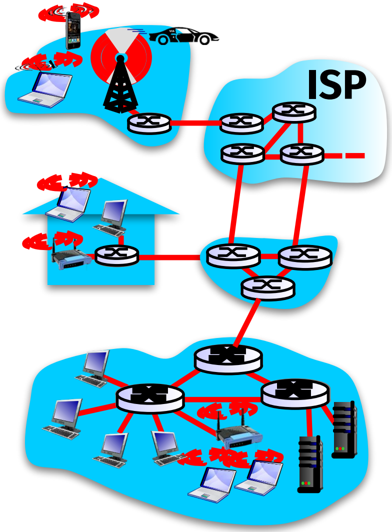
- What is its **addressing** based on?
  - ???

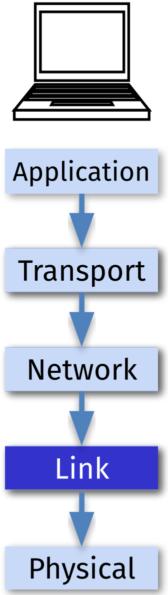




The Data Link Layer

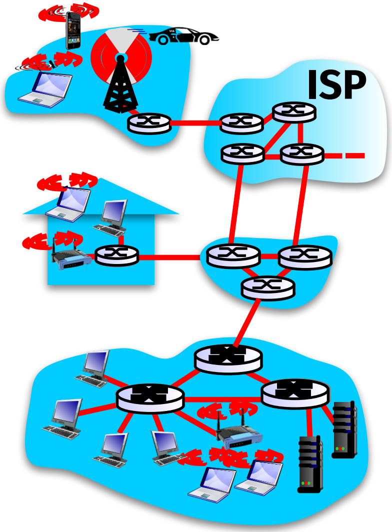
- **What is it?**
  - The **fourth layer** in the 5-layer network model
  - Responsible for the node-to-node delivery of data
- What are **“nodes”**? **Hosts**, **switches**
  - **Hosts:** the physical devices within a network
  - **Switches:** interface to all hosts on the network
- What is its **addressing** based on?
  - MAC addresses—a physical identifier for hardware
- Do MAC addresses guarantee **authenticity**?
  - **???**

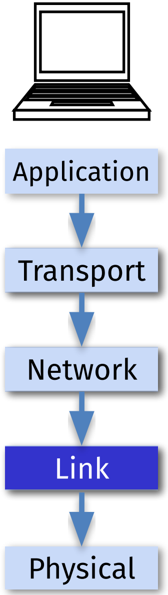




The Data Link Layer

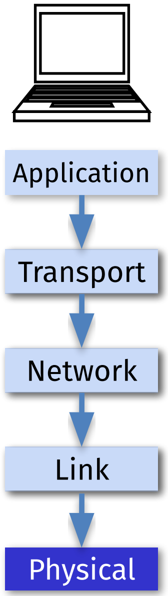
- **What is it?**
  - The **fourth layer** in the 5-layer network model
  - Responsible for the node-to-node delivery of data
- What are **“nodes”**? **Hosts**, **switches**
  - **Hosts:** the physical devices within a network
  - **Switches:** interface to all hosts on the network
- What is its **addressing** based on?
  - MAC addresses—a physical identifier for hardware
- Do MAC addresses guarantee **authenticity**?
  - Reconfigurable via network interface
  - **Attacker-spoofable**





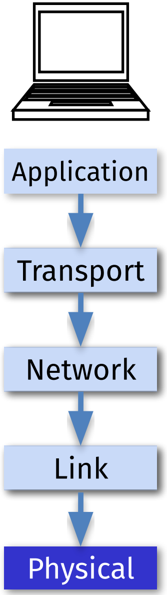
The Physical Layer

- **What is it?**
  - **???**



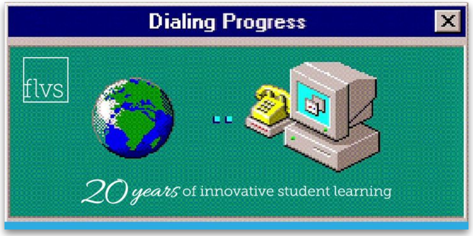
The Physical Layer

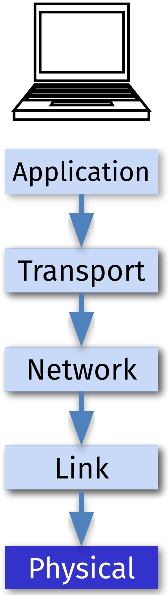
- **What is it?**
  - The **last layer** in the 5-layer network model
  - The physical means of sending/receiving data
- **Examples** of physical layers?
  - ???



The Physical Layer

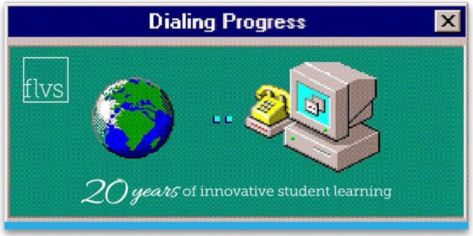
- **What is it?**
  - The **last layer** in the 5-layer network model
  - The physical means of sending/receiving data
- **Examples** of physical layers?
  - Radio waves
  - Telephone lines
  - Fiber optic cables
  - Undersea submarine cables
- Does physical layer guarantee **availability**?
  - ???

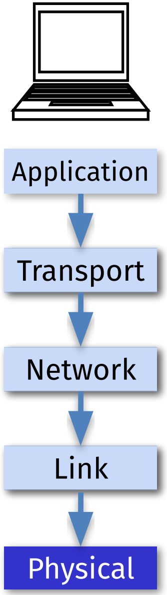




The Physical Layer

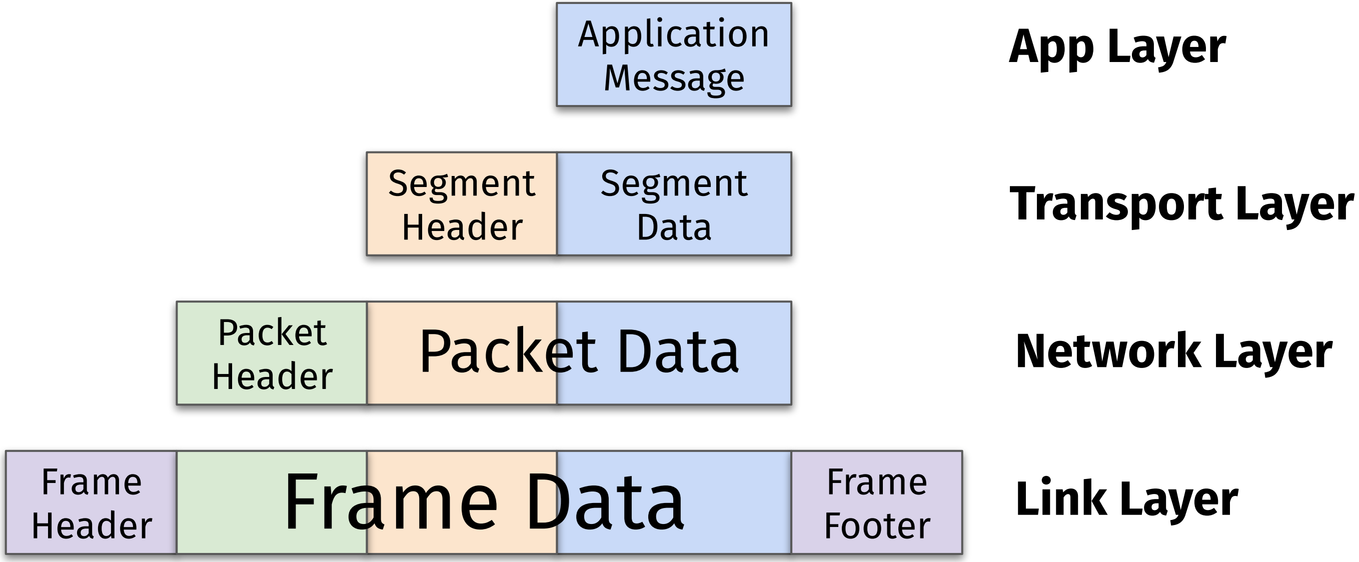
- **What is it?**
  - The **last layer** in the 5-layer network model
  - The physical means of sending/receiving data
- **Examples** of physical layers?
  - Radio waves
  - Telephone lines
  - Fiber optic cables
  - Undersea submarine cables
- Does physical layer guarantee **availability**?
  - **No—tamperable by third parties!**





Food for Thought

- Are any of the five network layers susceptible to **attacks**? If so, **which ones**?



Which network layers are susceptible to attack?

Physical	0%
Link	0%
Network	0%
Transport	0%
Application	0%
None of the above	0%
All of the above	0%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Food for Thought

- Are any hich ones?

Message

er

Packet

Packet Data

Network Layer

Fra
Hea

Every network layer is susceptible to **attack**

Today's focus: **Application Layer** attacks

Thursday's focus: attacking the **other layers**

Questions?

This time on CS 4440…

Application Layer Attacks

HTTP Content Injection

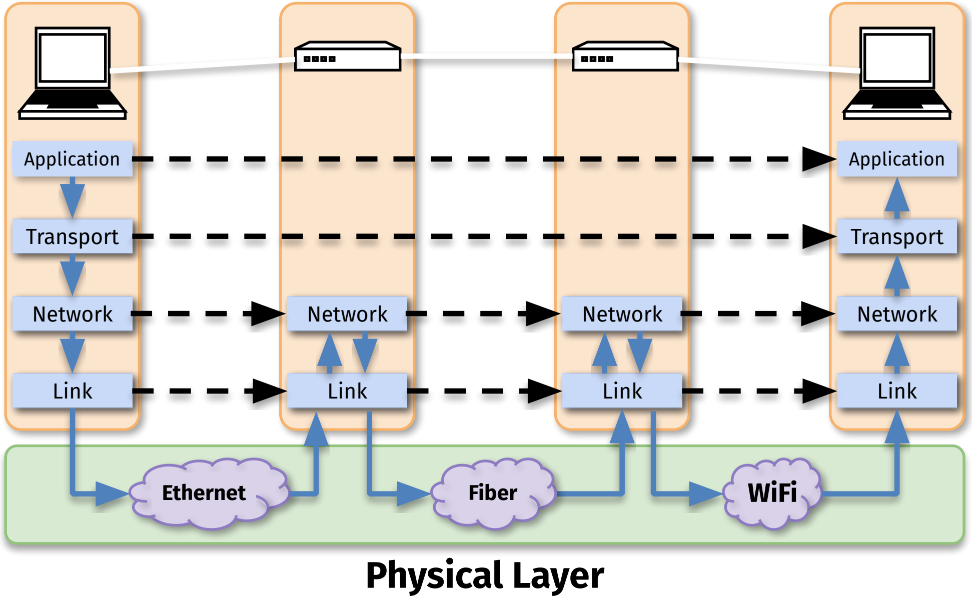
SMTP Header Spoofing

DNS Hijacking

Network Packet Analysis

Recap: The 5-Layer Internet

- **Application Layer:**
  - Sends/receives app messages
- **Transport Layer:**
  - Communication between apps
- **Network Layer:**
  - Communication between hosts
- **Data Link Layer:**
  - Node-to-node delivery of data
- **Physical Layer:**
  - Send/receive the physical signals

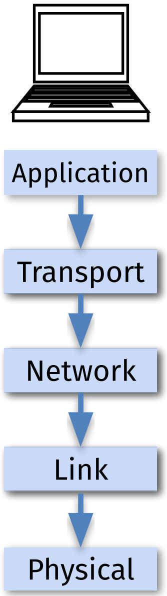


Where is each layer implemented?

- **Application Layer:** **???**
- **Transport Layer:** **???**
- **Network Layer:** **???**
- **Data Link Layer:** **???**

Quiz:

1. **Host Device** (e.g., your laptop)
2. **Other Devices** (e.g., switch, router)
3. **Both!**

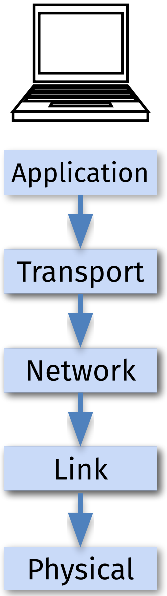


Where is each layer implemented?

- **Application Layer:** **Host**
  - E.g., browser, socket-based app
- **Transport Layer:** **Host**
  - OS library and necessary drivers
- **Network Layer:** **Both**
  - Within every host, router, switch
- **Data Link Layer:** **Host**
  - Network interface card, ethernet

Quiz:

1. **Host Device** (e.g., your laptop)
2. **Other Devices** (e.g., switch, router)
3. **Both!**



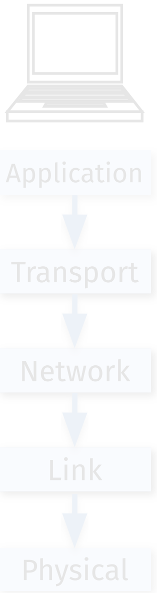
Where is each layer implemented?

- **Application Layer:** **Host**
  - E.g., browser, socket-based app
- **Transport L**
  - OS library
- **Network Layer:** **Both**
  - Within every host, router, switch
- **Data Link Layer:** **Host**
  - Network interface card, ethernet

Quiz:

2. **Non-host** (e.g., switch, router)
3. **Both!**

Why implement so much in **hosts**?



Where is each layer implemented?

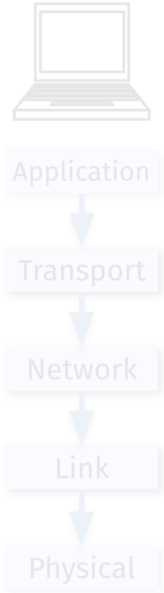
- **Application Layer:** **Host**
  - E.g., browser, socket-based app
- **Transport L**
  - OS library
- **Network La**
  - Within ev
- **Data Link Layer:** **Host**
  - Network interface card, ethernet

Quiz:

? **Non-host**

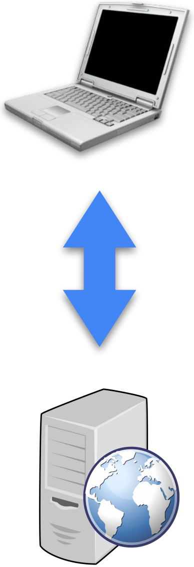
Why implement so much in **hosts**?

The End-to-End Principle



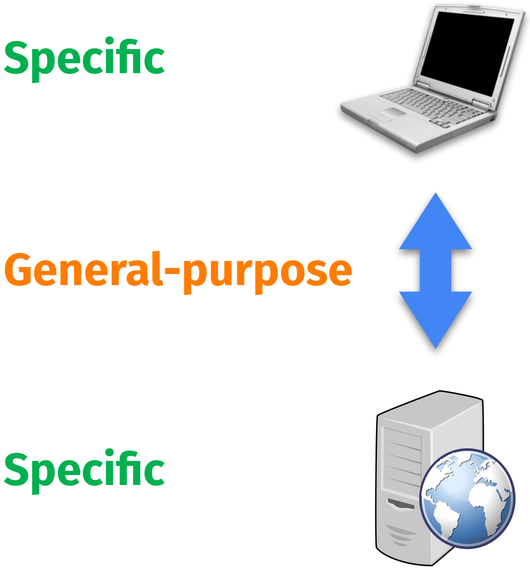
The End-to-End Principle

- Based on Paul Baran's 1960's work "reliability from unreliable parts"
- **Key idea:**
  - **Application-specific functions** ought to reside in the **end hosts** of a network
  - Rather than in **intermediary nodes**
  - All this assumes that the end host can implement it "completely and correctly"



The End-to-End Principle

- Based on Paul Baran's 1960's work “reliability from unreliable parts”
- **Key idea:**
    - **Application-specific functions** ought to reside in the **end hosts** of a network
    - Rather than in **intermediary nodes**
    - All this assumes that the end host can implement it “completely and correctly”



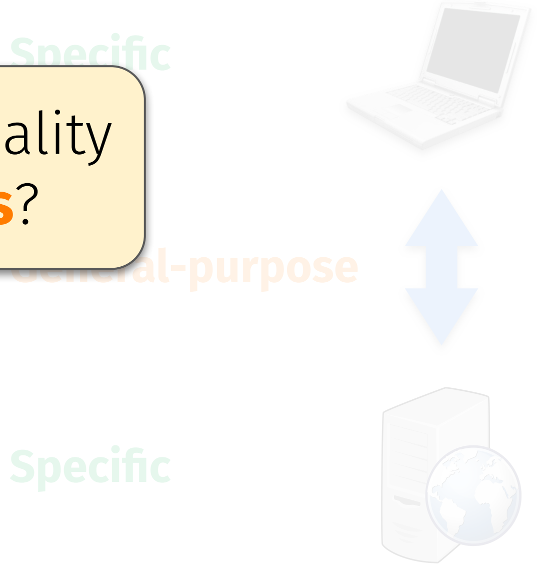
Specific

General-purpose

Specific

The End-to-End Principle

- Based on Paul Baran's 1960's work “reliability from unreliable parts”
- **Key idea:**
  - **Application-s** reside in the **end hosts** of a network
  - Rather than in **intermediary nodes**
  - All this assumes that the end host can implement it “completely and correctly”

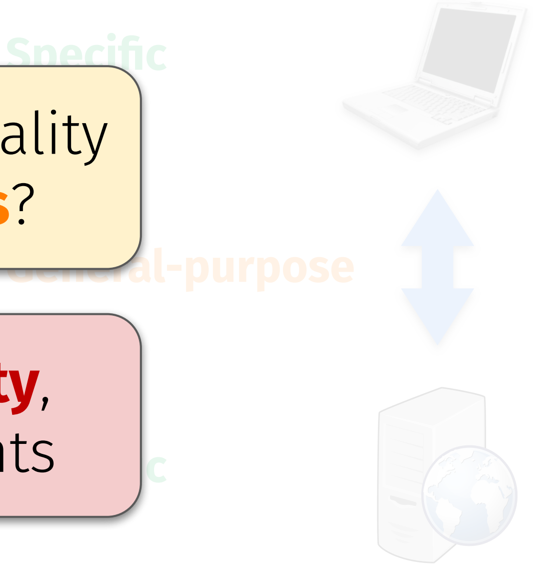


Why not move more functionality to the **intermediate nodes**?

The End-to-End Principle

- Based on Paul Baran's 1960's work “reliability from unreliable parts”

- **Key idea:**
  - **Application-s** reside in the end hosts of a network
  - Rather than i
  - All this assu implement it



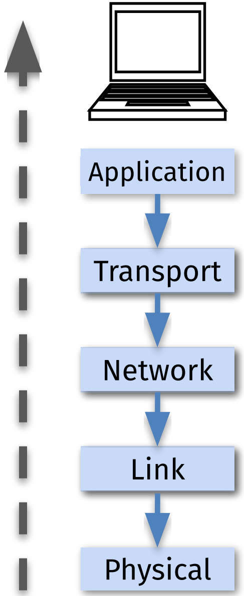
Why not move more functionality to the **intermediate nodes**?

More **latency**, less **flexibility**, higher-**complexity** midpoints

Implications of The E2E Principle

- **Nothing** guaranteed by default
  - No default **integrity**
  - No default **confidentiality**
  - No default **availability**
  - No default **authentication**

Complexity,
Specificity

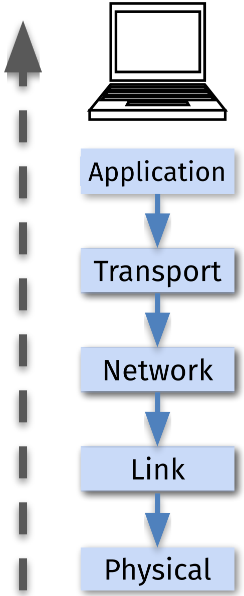


Implications of The E2E Principle

- **Nothing** guaranteed by default
  - No default **integrity**
  - No default **confidentiality**
  - No default **availability**
  - No default **authentication**

- **Solution: bolt-on** more security!
  - **IPsec** (Network layer)
    - Integrity, Confidentiality, Authentication
  - **DNSSec** (App layer)
    - Integrity, Authenticity
  - **TLS** (Session layer)
    - Integrity, Confidentiality, Authentication

Complexity, Specificity



Application Layer Network Attacks

Application Layer Attacks

- **Application Layer:** where **network-facing apps** send/receive message
  - Application-specific protocols (message semantics, structure, processing rules, etc.)

- **Attacking the application layer:**
  - **???**

Application Layer Attacks

- **Application Layer:** where **network-facing apps** send/receive message
  - Application-specific protocols (message semantics, structure, processing rules, etc.)

- **Attacking the application layer:**
  - **Command Injection**
    - SQL injection, CSRF, XSS
  - **Denial of Service**
    - Crash a remote application
    - Prevent others from using it
  - **Message Tampering / Sniffing**
    - Injecting data into messages
    - Capturing unencrypted data

Application Layer Attacks

- **Application Layer:** where **network-facing apps** send/receive message
  - Application-specific protocols (message semantics, structure, processing rules, etc.)

- **Attacking the application layer:**
  - **Command Injection**
    - SQL injection, CSRF, XSS
  - **Denial of Service**
    - Crash a remote application
    - Prevent others from using it
  - **Message Tampering / Sniffing**
    - Injecting data into messages
    - Capturing unencrypted data
  - **Other protocol-specific attacks**

http:// FTP

SMTP

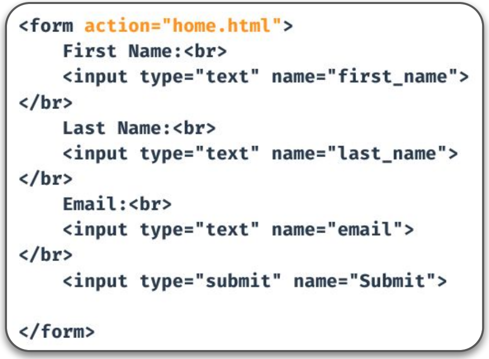
McAfee

zoom

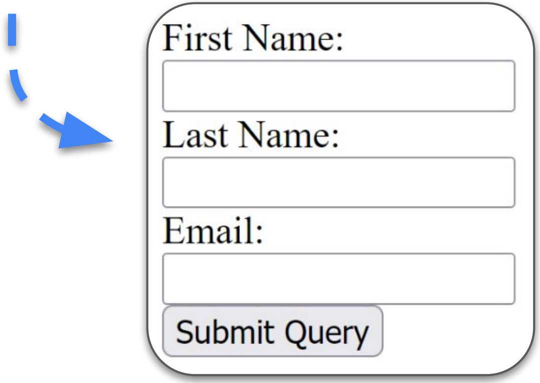
Application Layer Attacks
HTTP Content Injection

Recap: HyperText Markup Language (HTML)

- **What is HTML?**

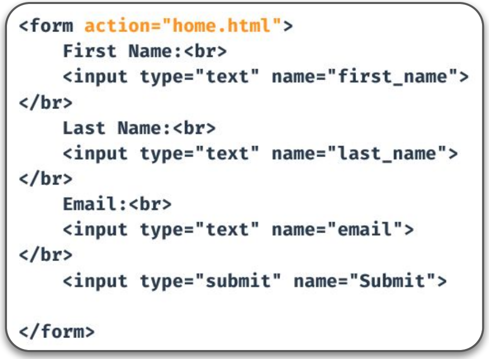


```
<form action="home.html">
    First Name:<br>
    <input type="text" name="first_name">
</br>
    Last Name:<br>
    <input type="text" name="last_name">
</br>
    Email:<br>
    <input type="text" name="email">
</br>
    <input type="submit" name="Submit">

</form>
```

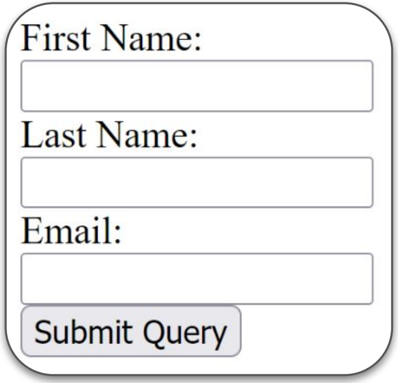


Recap: HyperText Markup Language (HTML)

- **What is HTML?**
  - Describes **content** and **formatting** of web pages
  - Rendered within browser window
- **HTML features**
  - **Static** document description language
  - Links to external pages, images by **reference**
  - User input sent to server via **forms**
- **HTML extensions**
  - Additional media (e.g., PDF, videos) via **plugins**
  - Embedding **programs** in other languages (e.g., **Java**) provides **dynamic content** that can:
    - Interacts with the user
    - Modify the browser user interface
    - Access the client computer environment

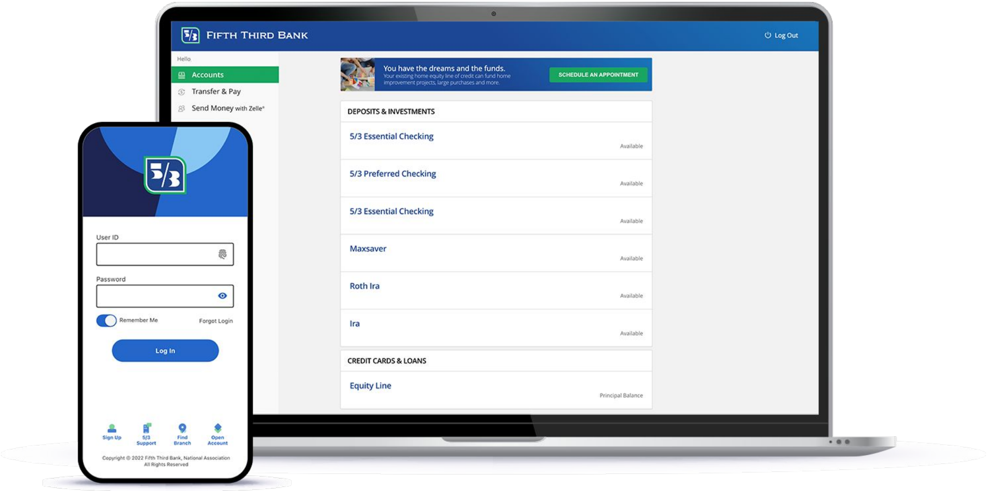


```
<form action="home.html">
    First Name:<br>
    <input type="text" name="first_name">
</br>
    Last Name:<br>
    <input type="text" name="last_name">
</br>
    Email:<br>
    <input type="text" name="email">
</br>
    <input type="submit" name="Submit">

</form>
```



Recap: HyperText Transfer Protocol (HTTP)

- **What is HTTP?**



What is HTTP?

How we transmit hyper-media objects over the web!

0%

An unencrypted, stateless protocol for transmitting information from server to client (and back).

0%

Halloween Toblerone transfer protocol 🎃🍫

0%

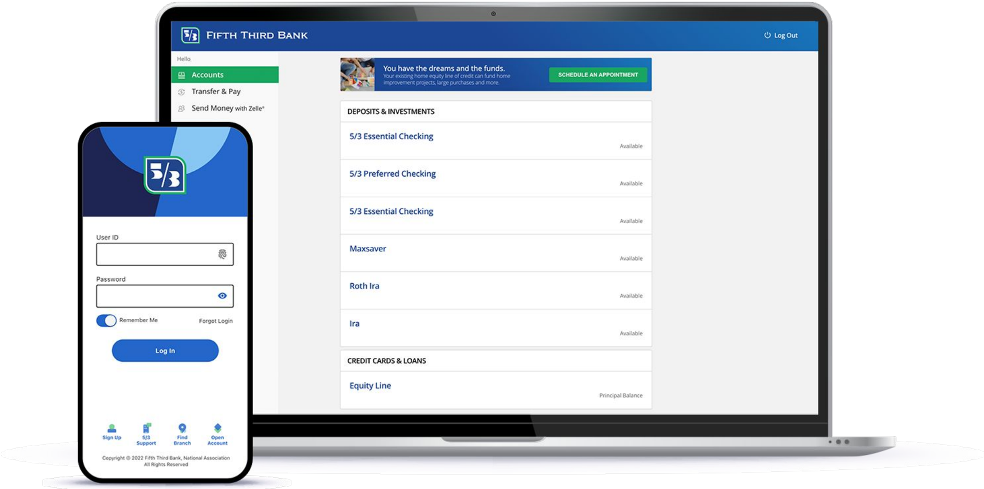
None of the above

0%

Recap: HyperText Transfer Protocol (HTTP)

- **What is HTTP?**
  - Protocol for transmitting **hypermedia documents** (e.g., web pages)

- **HTTP's Characteristics:**
  - Widely used
  - **Simple**
  - **Unencrypted**



You have the dreams and the funds.
Your existing home equity line of credit can fund home improvement projects, large purchases and more.

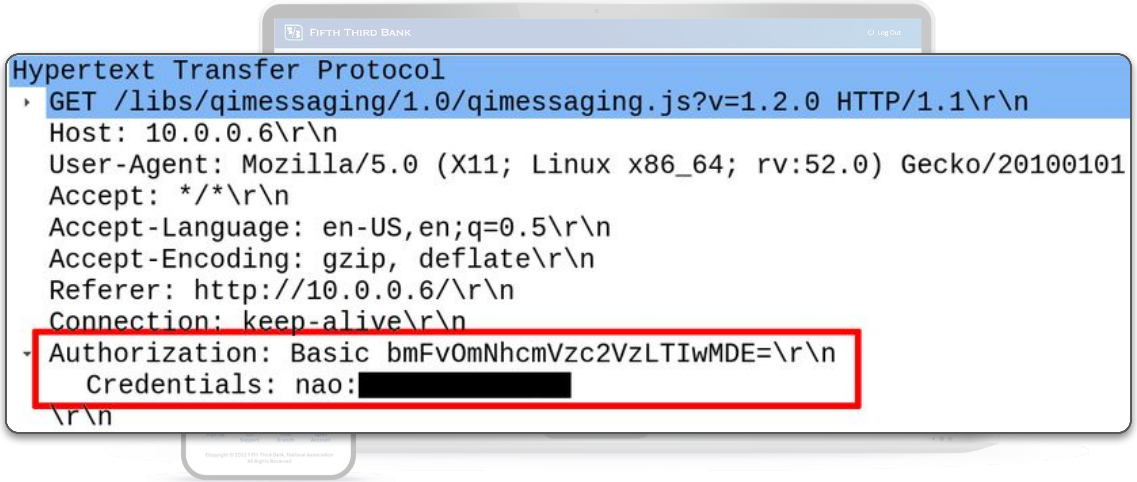
Copyright © 2022 Fifth Third Bank, National Association
All Rights Reserved

Recap: HyperText Transfer Protocol (HTTP)

- **What is HTTP?**
  - Protocol for transmitting **hypermedia documents** (e.g., web pages)

- **HTTP's Characteristics:**
  - Widely used
  - **Simple**
  - **Unencrypted**

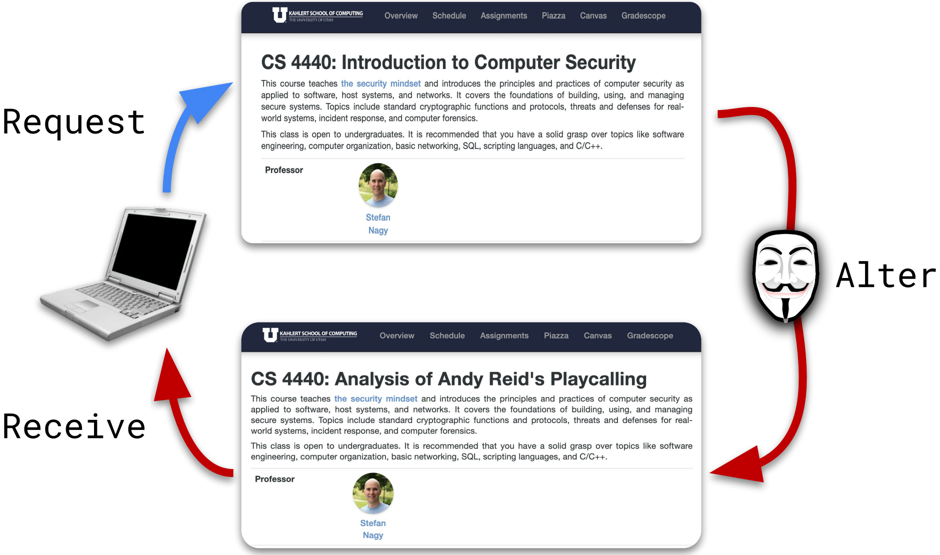
Problem: no way of keeping data hidden from **prying eyes**!



```
Hypertext Transfer Protocol
 ▸ GET /libs/qimessaging/1.0/qimessaging.js?v=1.2.0 HTTP/1.1\r\n
   Host: 10.0.0.6\r\n
   User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
   Accept: */*\r\n
   Accept-Language: en-US,en;q=0.5\r\n
   Accept-Encoding: gzip, deflate\r\n
   Referer: http://10.0.0.6/\r\n
   Connection: keep-alive\r\n
 ▾ Authorization: Basic bmFvOmNhcmVzc2VzLTIwMDE=\r\n
     Credentials: nao:[redacted]
   \r\n
```

Tampering with HTTP-transmitted HTML

- Capitalizes on **HTTP's** insecurity
    - **Nothing is encrypted!**
- **Attacker intercepts** requested webpage and **modifies it**
    - User receives modified webpage
- **Attacker capabilities?**
    - **???**



CS 4440: Introduction to Computer Security

This course teaches the security mindset and introduces the principles and practices of computer security as applied to software, host systems, and networks. It covers the foundations of building, using, and managing secure systems. Topics include standard cryptographic functions and protocols, threats and defenses for real-world systems, incident response, and computer forensics.

This class is open to undergraduates. It is recommended that you have a solid grasp over topics like software engineering, computer organization, basic networking, SQL, scripting languages, and C/C++.

Professor

Stefan Nagy

CS 4440: Analysis of Andy Reid's Playcalling

This course teaches the security mindset and introduces the principles and practices of computer security as applied to software, host systems, and networks. It covers the foundations of building, using, and managing secure systems. Topics include standard cryptographic functions and protocols, threats and defenses for real-world systems, incident response, and computer forensics.

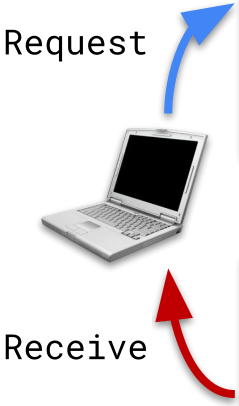
This class is open to undergraduates. It is recommended that you have a solid grasp over topics like software engineering, computer organization, basic networking, SQL, scripting languages, and C/C++.

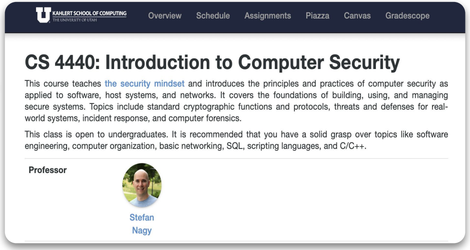
Professor

Stefan Nagy

Tampering with HTTP-transmitted HTML

- Capitalizes on **HTTP's** insecurity
  - **Nothing is encrypted!**
- **Attacker intercepts** requested webpage and **modifies it**
  - User receives modified webpage
- **Attacker capabilities?**
  - Inject malicious **content**
  - Inject malicious **code**





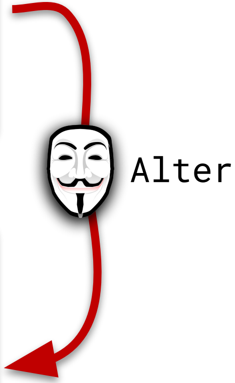
KAHLERT SCHOOL OF COMPUTING THE UNIVERSITY OF UTAH Overview Schedule Assignments Piazza Canvas Gradescope

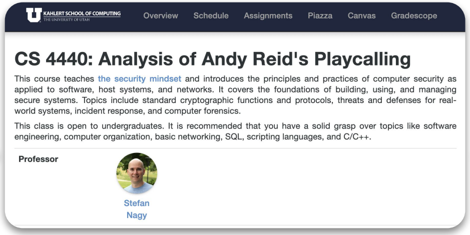
CS 4440: Introduction to Computer Security

This course teaches the security mindset and introduces the principles and practices of computer security as applied to software, host systems, and networks. It covers the foundations of building, using, and managing secure systems. Topics include standard cryptographic functions and protocols, threats and defenses for real-world systems, incident response, and computer forensics.

This class is open to undergraduates. It is recommended that you have a solid grasp over topics like software engineering, computer organization, basic networking, SQL, scripting languages, and C/C++.

Professor Stefan Nagy





KAHLERT SCHOOL OF COMPUTING THE UNIVERSITY OF UTAH Overview Schedule Assignments Piazza Canvas Gradescope

CS 4440: Analysis of Andy Reid's Playcalling

This course teaches the security mindset and introduces the principles and practices of computer security as applied to software, host systems, and networks. It covers the foundations of building, using, and managing secure systems. Topics include standard cryptographic functions and protocols, threats and defenses for real-world systems, incident response, and computer forensics.

This class is open to undergraduates. It is recommended that you have a solid grasp over topics like software engineering, computer organization, basic networking, SQL, scripting languages, and C/C++.

Professor Stefan Nagy

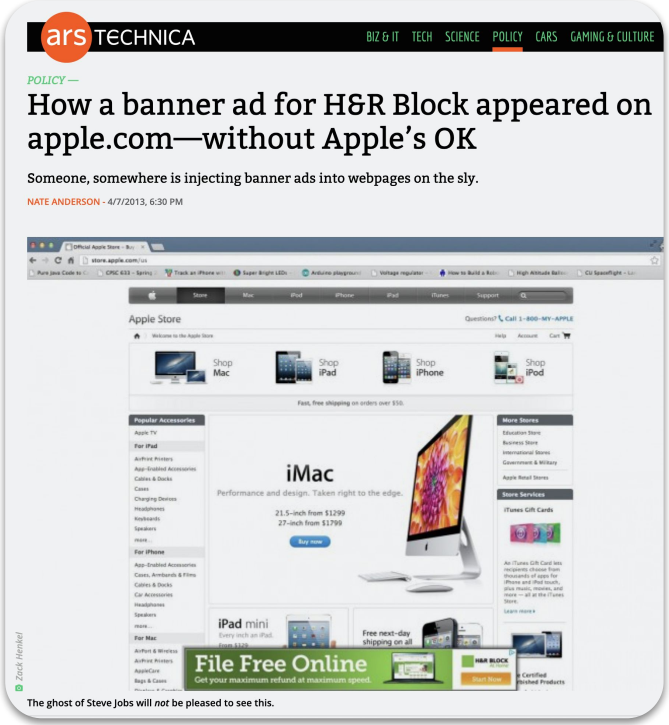
Case Study: Content Injection

- Do you trust your **ISP**?
  - Could they tamper with data?

Case Study: Content Injection

- Do you trust your **ISP**?
  - Could they tamper with data?

- **Attack:** ISP intercepts HTML pages transmitted via **HTTP**
  - Injects advertisements
    - They make commission



POLICY —

How a banner ad for H&R Block appeared on apple.com—without Apple's OK

Someone, somewhere is injecting banner ads into webpages on the sly.

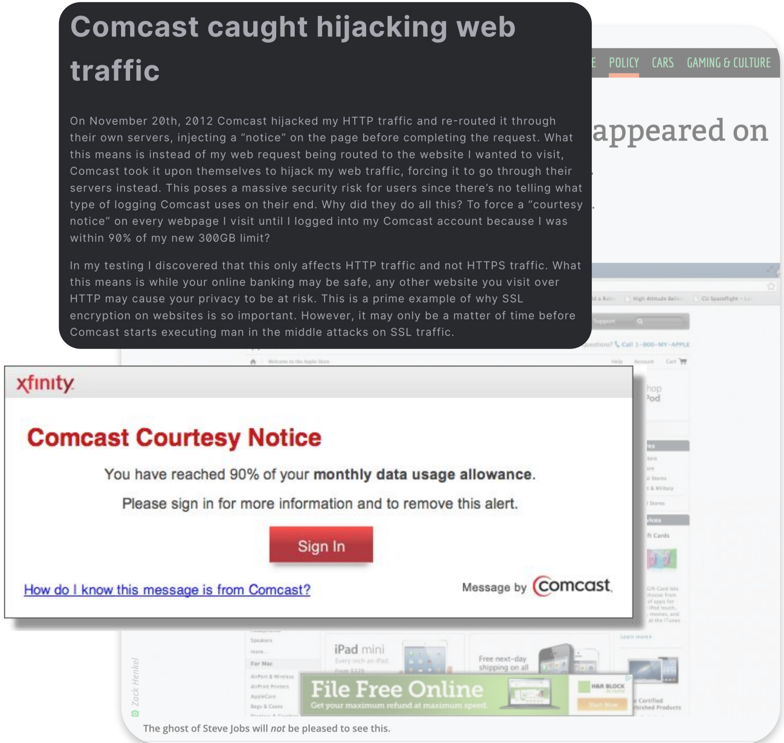
NATE ANDERSON - 4/7/2013, 6:30 PM

The ghost of Steve Jobs will *not* be pleased to see this.

Case Study: Content Injection

- Do you trust your **ISP**?
  - Could they tamper with data?

- **Attack:** ISP intercepts HTML pages transmitted via **HTTP**
  - Injects advertisements
    - They make commission
  - Injects pop-up messages
    - E.g., monthly data usage



Comcast caught hijacking web traffic

On November 20th, 2012 Comcast hijacked my HTTP traffic and re-routed it through their own servers, injecting a "notice" on the page before completing the request. What this means is instead of my web request being routed to the website I wanted to visit, Comcast took it upon themselves to hijack my web traffic, forcing it to go through their servers instead. This poses a massive security risk for users since there's no telling what type of logging Comcast uses on their end. Why did they do all this? To force a "courtesy notice" on every webpage I visit until I logged into my Comcast account because I was within 90% of my new 300GB limit?

In my testing I discovered that this only affects HTTP traffic and not HTTPS traffic. What this means is while your online banking may be safe, any other website you visit over HTTP may cause your privacy to be at risk. This is a prime example of why SSL encryption on websites is so important. However, it may only be a matter of time before Comcast starts executing man in the middle attacks on SSL traffic.

xfinity

Comcast Courtesy Notice

You have reached 90% of your **monthly data usage allowance**.

Please sign in for more information and to remove this alert.

Sign In

How do I know this message is from Comcast?

Message by Comcast

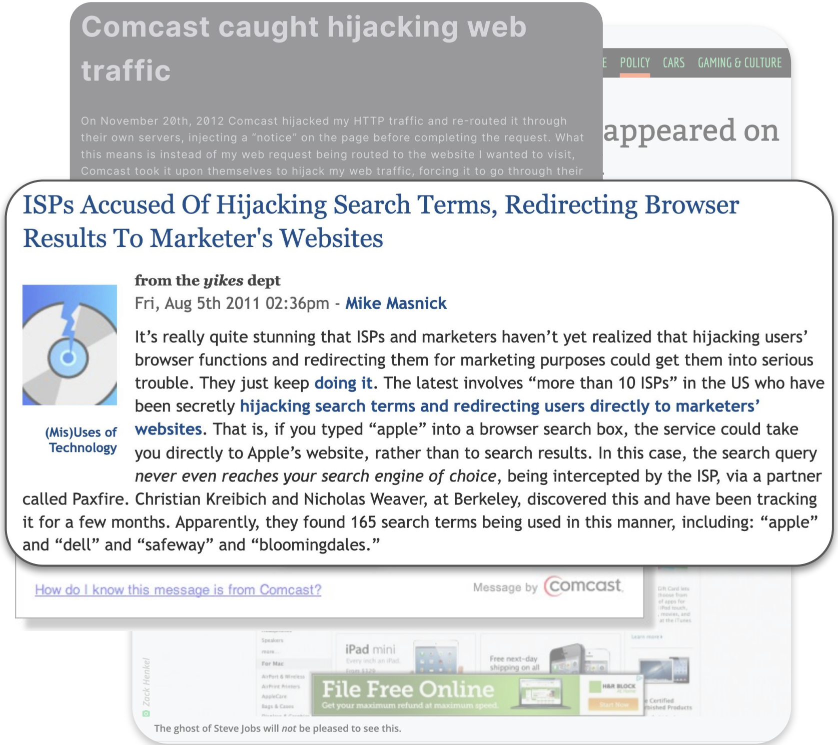
The ghost of Steve Jobs will *not* be pleased to see this.

E POLICY CARS GAMING & CULTURE
appeared on
iPad mini
Free next-day shipping on all
File Free Online
Get your maximum refund at maximum speed.
H&R BLOCK
Zack Henkel

Case Study: Content Injection

- Do you trust your **ISP**?
  - Could they tamper with data?

- **Attack:** ISP intercepts HTML pages transmitted via **HTTP**
  - Injects advertisements
    - They make commission
  - Injects pop-up messages
    - E.g., monthly data usage
  - Redirect search engine results
    - More commission!



Comcast caught hijacking web traffic

On November 20th, 2012 Comcast hijacked my HTTP traffic and re-routed it through their own servers, injecting a "notice" on the page before completing the request. What this means is instead of my web request being routed to the website I wanted to visit, Comcast took it upon themselves to hijack my web traffic, forcing it to go through their

ISPs Accused Of Hijacking Search Terms, Redirecting Browser Results To Marketer's Websites

from the *yikes* dept

Fri, Aug 5th 2011 02:36pm - Mike Masnick

It's really quite stunning that ISPs and marketers haven't yet realized that hijacking users' browser functions and redirecting them for marketing purposes could get them into serious trouble. They just keep doing it. The latest involves "more than 10 ISPs" in the US who have been secretly hijacking search terms and redirecting users directly to marketers' websites. That is, if you typed "apple" into a browser search box, the service could take you directly to Apple's website, rather than to search results. In this case, the search query *never even reaches your search engine of choice*, being intercepted by the ISP, via a partner called Paxfire. Christian Kreibich and Nicholas Weaver, at Berkeley, discovered this and have been tracking it for a few months. Apparently, they found 165 search terms being used in this manner, including: "apple" and "dell" and "safeway" and "bloomingdales."

How do I know this message is from Comcast?

Message by Comcast

iPad mini
Free next-day shipping on all
File Free Online
Get your maximum refund at maximum speed.
H&R BLOCK
Zack Henkel

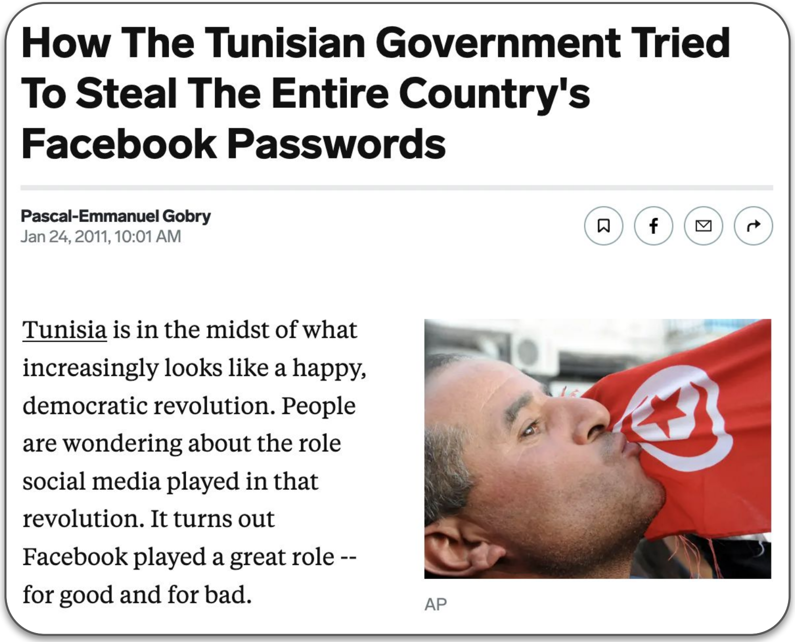
The ghost of Steve Jobs will *not* be pleased to see this.

Case Study: Code Injection

- Do you trust your **government**?
    - Could they tamper with data?

Case Study: Code Injection

- Do you trust your **government**?
  - Could they tamper with data?
- **Attack:** government forces ISPs to inject code into HTTP content
  - Steal HTTP-transmitted **passwords**
    - E.g., Facebook, GMail, Twitter



How The Tunisian Government Tried To Steal The Entire Country's Facebook Passwords

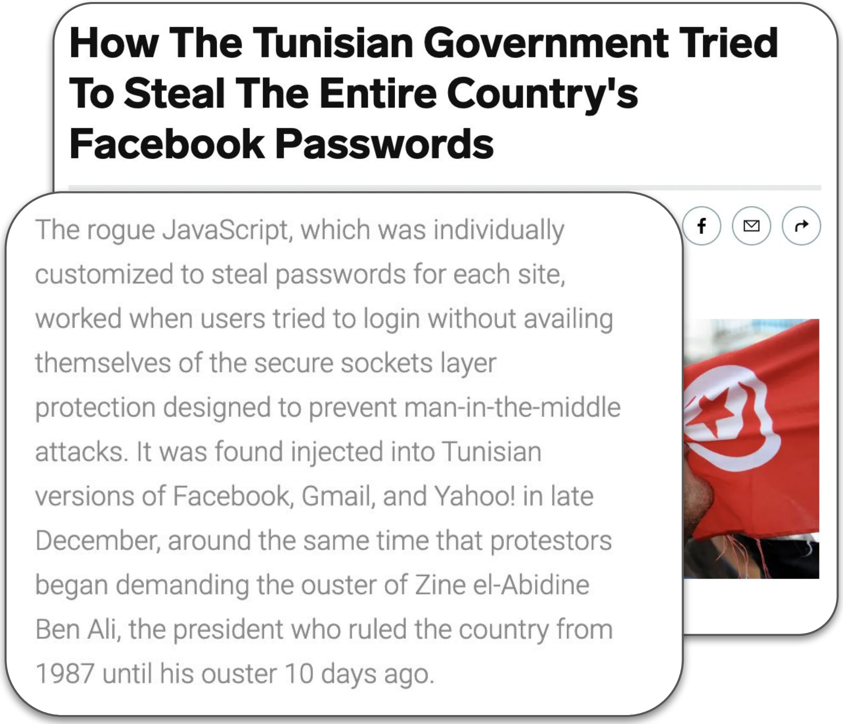
Pascal-Emmanuel Gobry
Jan 24, 2011, 10:01 AM

Tunisia is in the midst of what increasingly looks like a happy, democratic revolution. People are wondering about the role social media played in that revolution. It turns out Facebook played a great role -- for good and for bad.

AP

Case Study: Code Injection

- Do you trust your **government**?
  - Could they tamper with data?
- **Attack:** government forces ISPs to **inject code** into HTTP content
  - Steal HTTP-transmitted **passwords**
    - E.g., Facebook, GMail, Twitter
  - **Result: persistent XSS** attack
    - Passes **Same-origin Policy**!



How The Tunisian Government Tried To Steal The Entire Country's Facebook Passwords

The rogue JavaScript, which was individually customized to steal passwords for each site, worked when users tried to login without availing themselves of the secure sockets layer protection designed to prevent man-in-the-middle attacks. It was found injected into Tunisian versions of Facebook, Gmail, and Yahoo! in late December, around the same time that protestors began demanding the ouster of Zine el-Abidine Ben Ali, the president who ruled the country from 1987 until his ouster 10 days ago.

Thwarting HTTP Injection

- How do we prevent code and content injection in HTTP-transmitted data?

Thwarting HTTP Injection

- How do we prevent code and content injection in HTTP-transmitted data?

Answer: *completely* ditch HTTP!

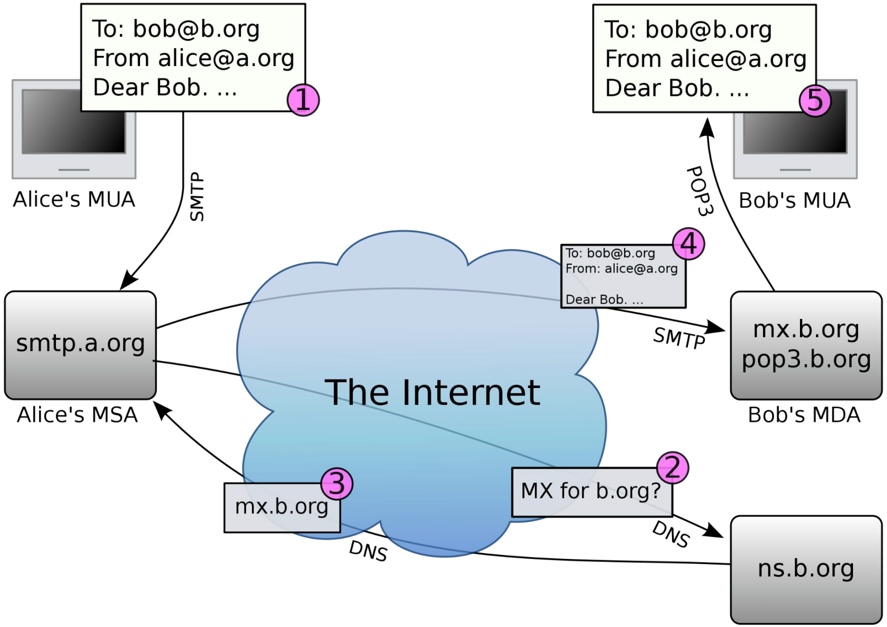
- As web and app developers, enforce strict **HTTPS** compliance
  - Necessary to prevent **HTTPS→HTTP downgrade attacks**

Questions?

Application Layer Attacks
SMTP Header Spoofing

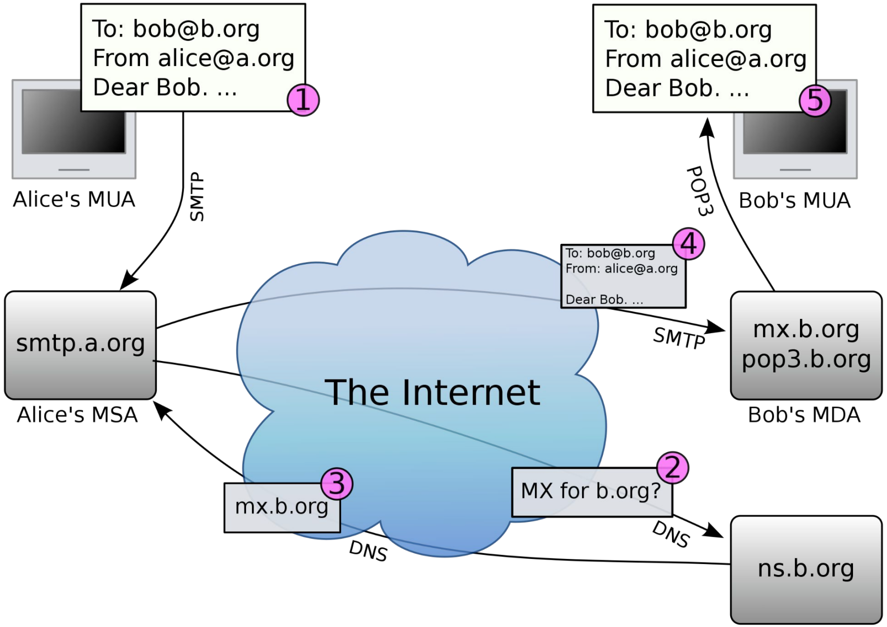
How does sending E-mail work?

- Nigerian Prince writes me a great investment opportunity
- **Sends it!**



How does sending E-mail work?

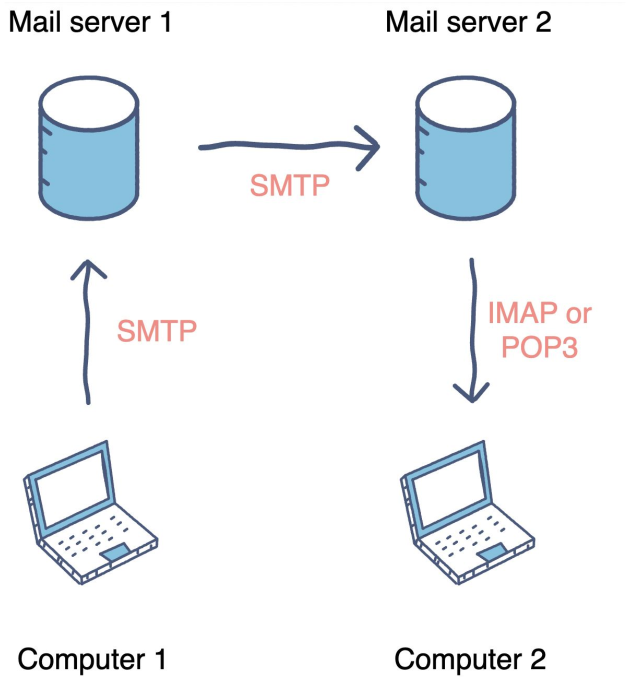
- Nigerian Prince writes me a great investment opportunity
- **Sends it!**
  - Protocol **SMTP** (aka the **Simple Mail Transfer Protocol**)
  - **Sender's SMTP server** breaks up the message into body/receiver
  - **Sender's SMTP server** queries DNS to find **receiver's server IP**
  - **Receiver's SMTP server** gets msg, then queries **its POP3 server** to find the **correct user mailbox**



SMTP Protocol

- **SMTP:** Simple Mail Transfer Protocol
  - Implemented in the **application** layer

- **Characteristics:**
  - Text-based
  - Connection-oriented
  - Uses TCP ports 25/587

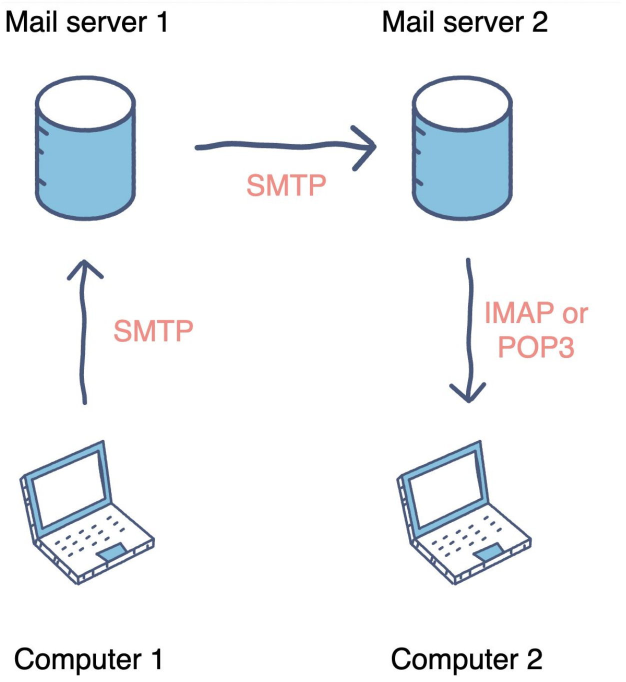


SMTP Protocol

- **SMTP:** Simple Mail Transfer Protocol
  - Implemented in the **application** layer

- **Characteristics:**
  - Text-based
  - Connection-oriented
  - Uses TCP ports 25/587

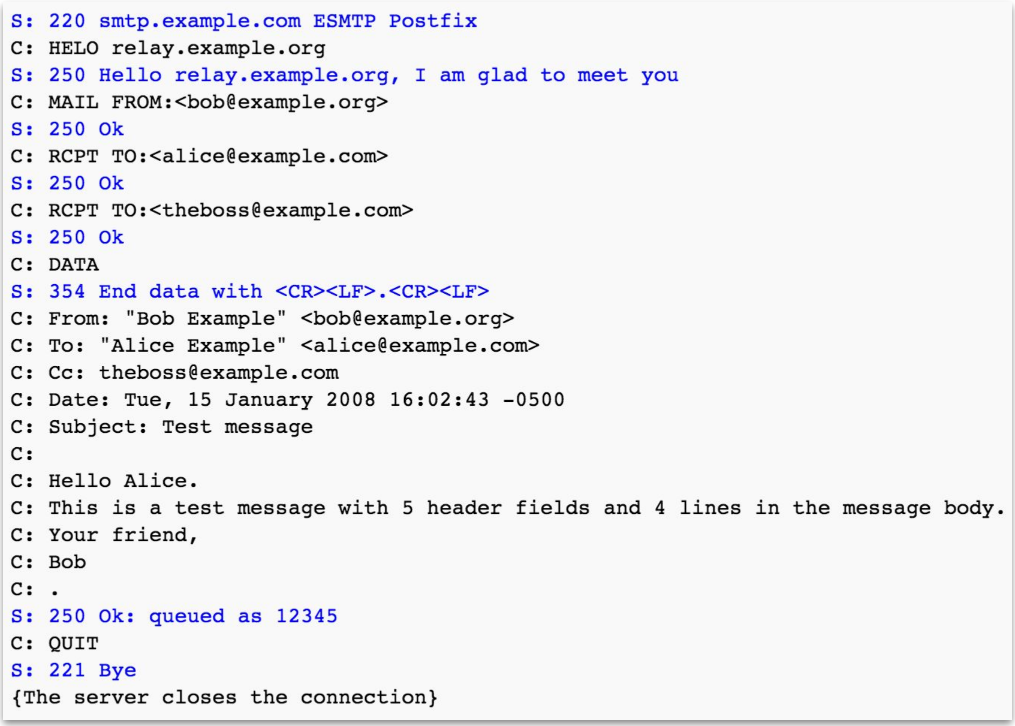
- **Security guarantees:**
  - Message integrity—**no!**
  - Confidentiality—**no!**
  - Authentication—**no!**



Example SMTP Connection

- **Plain SMTP:**
  - No encryption whatsoever

- **Key Protocol fields:**
  - HELO: setup sender’s server
  - MAIL FROM: sender address
  - RCPT TO: recipient address
  - DATA: subject, body, files



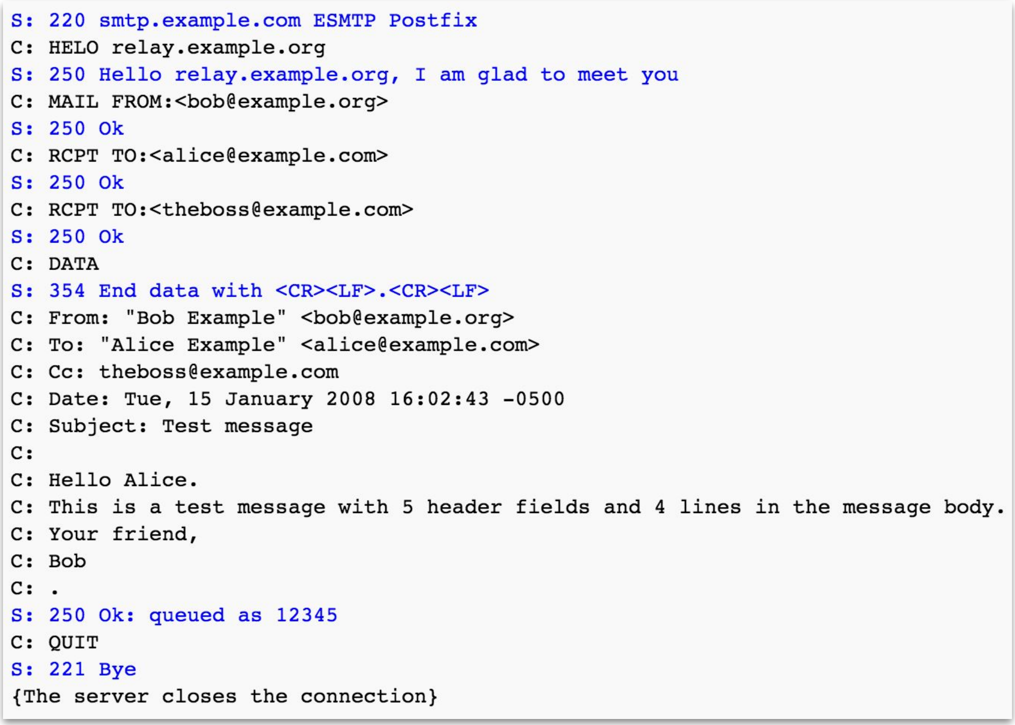
```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

Example SMTP Connection

- **Plain SMTP:**
  - No encryption whatsoever

- **Key Protocol fields:**
  - HELO: setup sender's server
  - MAIL FROM: sender address
  - RCPT TO: recipient address
  - DATA: subject, body, files

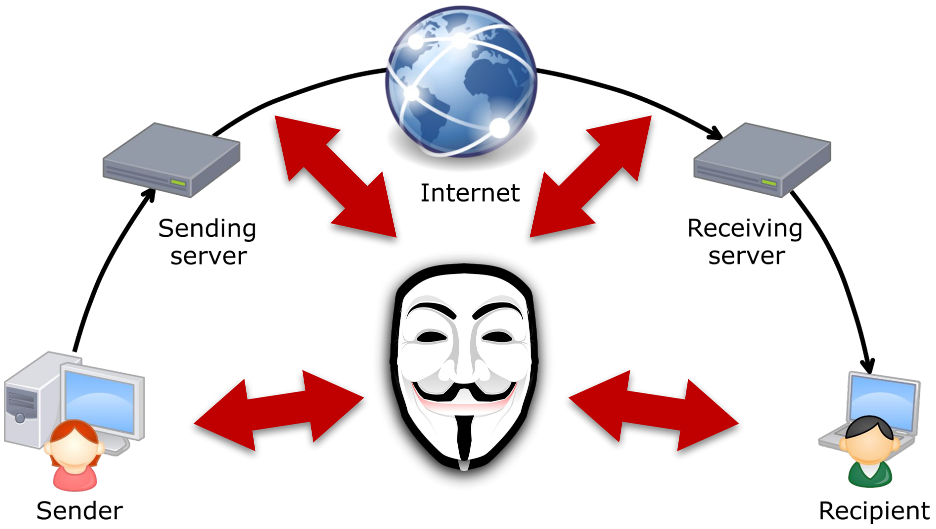
What could an attacker do?



```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

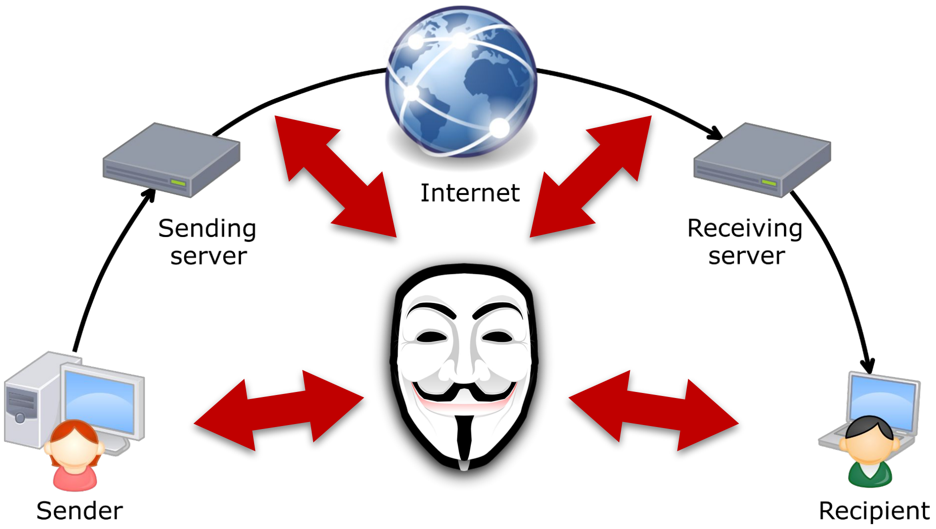
SMTP Attacks

- No **message integrity**
  - Tamper with messages
  - Block messages



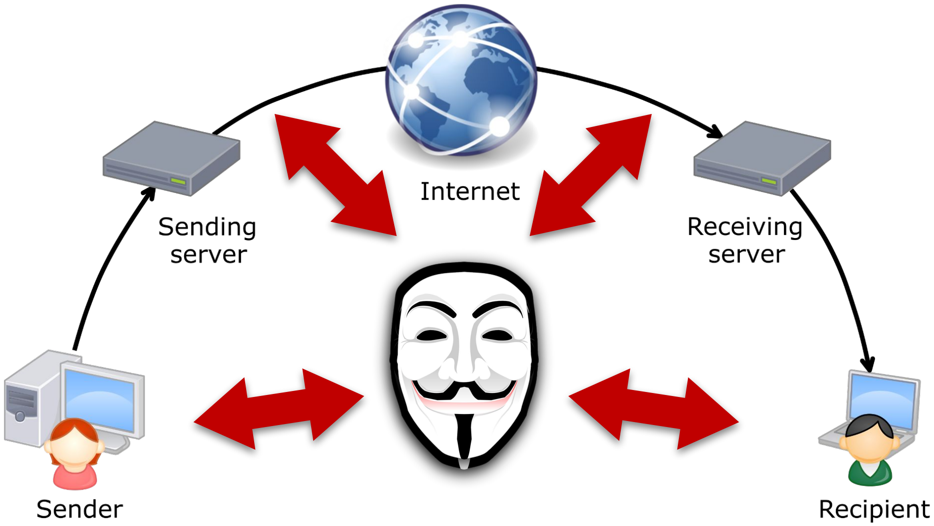
SMTP Attacks

- No **message integrity**
  - Tamper with messages
  - Block messages
- No **confidentiality**
  - Find sender/recipient
  - Read message contents



SMTP Attacks

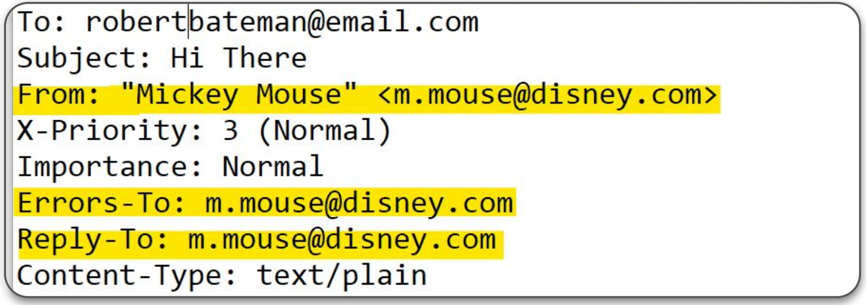
- No **message integrity**
  - Tamper with messages
  - Block messages
- No **confidentiality**
  - Find sender/recipient
  - Read message contents
- No **authentication**
  - Spoof sender identity



Case Study: Email Header Spoofing

- **Attack:** spoof email header to **mislead recipient** about sender of the email

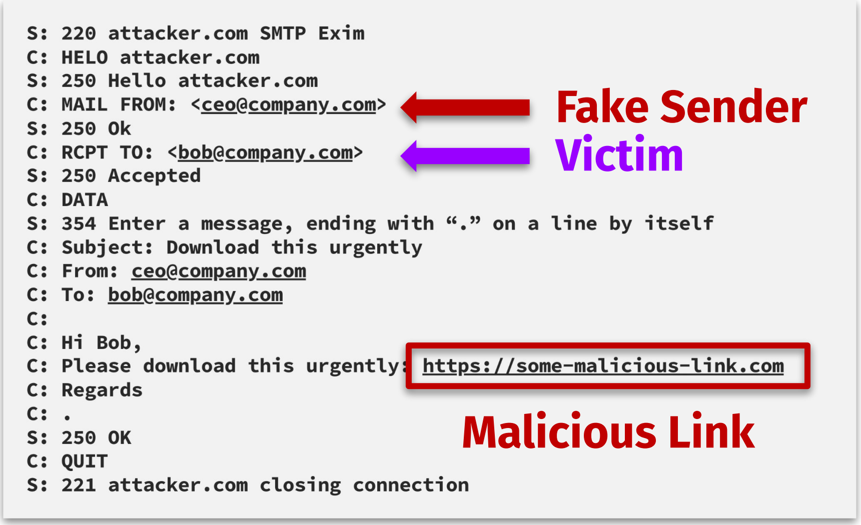
```
S: 220 attacker.com SMTP Exim
C: HELO attacker.com
S: 250 Hello attacker.com
C: MAIL FROM: <ceo@company.com>
S: 250 Ok
C: RCPT TO: <bob@company.com>
S: 250 Accepted
C: DATA
S: 354 Enter a message, ending with "." on a line by itself
C: Subject: Download this urgently
C: From: ceo@company.com
C: To: bob@company.com
C:
C: Hi Bob,
C: Please download this urgently: https://some-malicious-link.com
C: Regards
C: .
S: 250 OK
C: QUIT
S: 221 attacker.com closing connection
```



```
To: robertbateman@email.com
Subject: Hi There
From: "Mickey Mouse" <m.mouse@disney.com>
X-Priority: 3 (Normal)
Importance: Normal
Errors-To: m.mouse@disney.com
Reply-To: m.mouse@disney.com
Content-Type: text/plain
```

Case Study: Email Header Spoofing

- **Attack:** spoof email header to **mislead recipient** about sender of the email

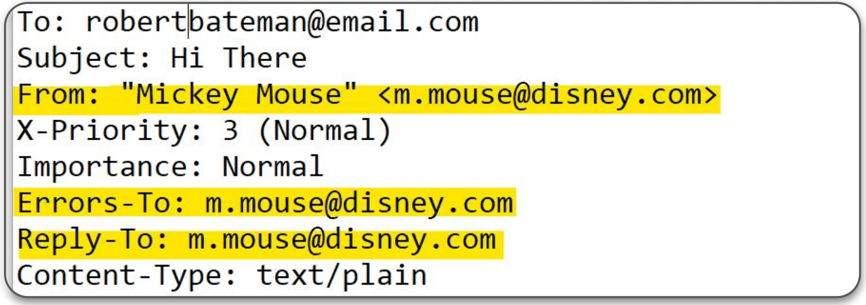


```
S: 220 attacker.com SMTP Exim
C: HELO attacker.com
S: 250 Hello attacker.com
C: MAIL FROM: <ceo@company.com>
S: 250 Ok
C: RCPT TO: <bob@company.com>
S: 250 Accepted
C: DATA
S: 354 Enter a message, ending with “.” on a line by itself
C: Subject: Download this urgently
C: From: ceo@company.com
C: To: bob@company.com
C:
C: Hi Bob,
C: Please download this urgently: https://some-malicious-link.com
C: Regards
C: .
S: 250 OK
C: QUIT
S: 221 attacker.com closing connection
```

Fake Sender

Victim

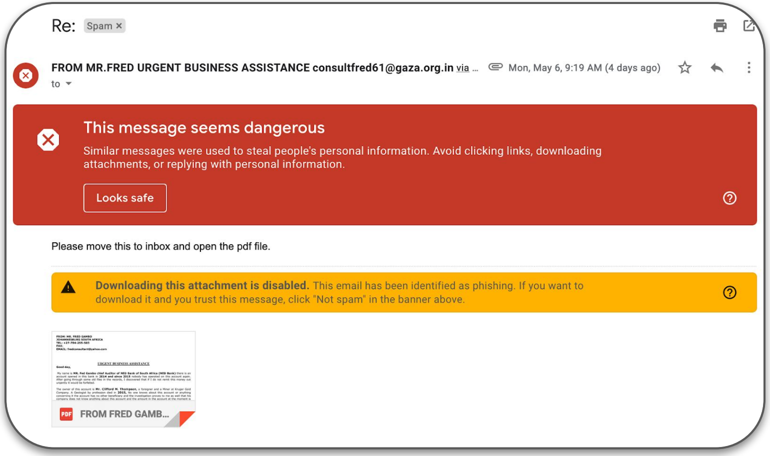
Malicious Link



```
To: robertbateman@email.com
Subject: Hi There
From: "Mickey Mouse" <m.mouse@disney.com>
X-Priority: 3 (Normal)
Importance: Normal
Errors-To: m.mouse@disney.com
Reply-To: m.mouse@disney.com
Content-Type: text/plain
```

Thwarting Email Spoofing

- **Checking email bodies**
  - Included links
  - Attached files
  - Text analysis (e.g., known spam campaigns)



Re: Spam ×

FROM MR.FRED URGENT BUSINESS ASSISTANCE consultfred61@gaza.org.in via ... Mon, May 6, 9:19 AM (4 days ago)

to

This message seems dangerous

Similar messages were used to steal people's personal information. Avoid clicking links, downloading attachments, or replying with personal information.

Looks safe

Please move this to inbox and open the pdf file.

Downloading this attachment is disabled. This email has been identified as phishing. If you want to download it and you trust this message, click "Not spam" in the banner above.

FROM: MR. FRED GAMBO
JOHANNESBURG SOUTH AFRICA
TEL: +27-784-255-503
FAX:
EMAIL: fredconsultant@yahoo.com

URGENT BUSINESS ASSISTANCE

Good day,

My name is **MR. Fed Gambo chief Auditor of NED Bank of South Africa (NED Bank)** there is an account opened in this bank in **2014 and since 2015** nobody has operated on this account again. After going through some old files in the records, I discovered that if I do not remit this money out urgently it would be forfeited.

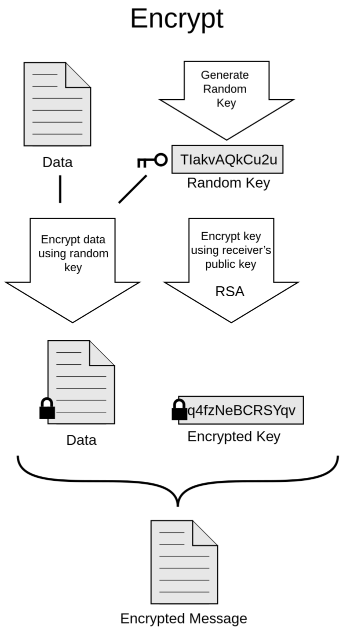
The owner of this account is **Mr. Clifford H. Thompson,** a foreigner and a Miner at Kruger Gold Company. A Geologist by profession died in **2015,** No one knows about this account or anything concerning it the account has no other beneficiary and the investigation proves to me as well that his company does not know anything about this account and the amount in the account at the moment is

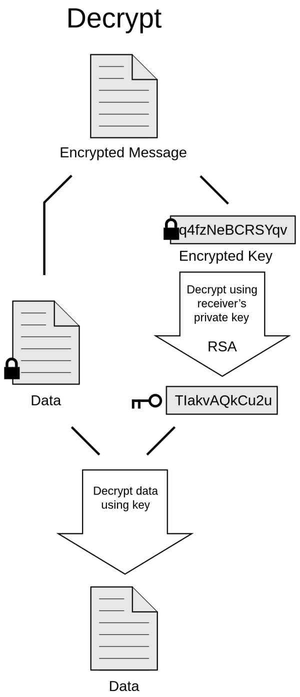
PDF FROM FRED GAMB...

Thwarting Email Spoofing

- **Checking email bodies**
  - Included links
  - Attached files
  - Text analysis (e.g., known spam campaigns)

- **Checking email headers**
  - Egress server domain registration
    - Check that sender is who it says it is
  - Pretty Good Privacy (PGP)
    - Sender and Receiver authentication
    - Confidentiality
    - Integrity





Questions?

Application Layer Attacks
DNS Hijacking

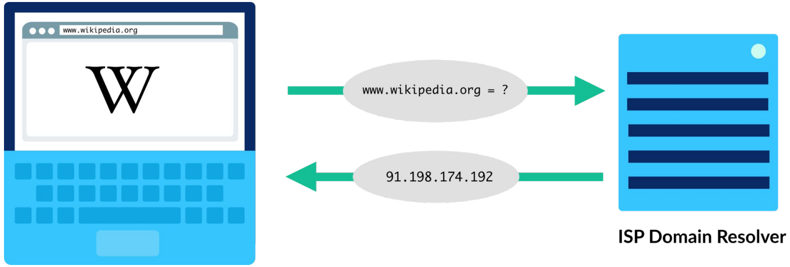
Identification on the Web

- How do we identify **people**?
  - **???**

Identification on the Web

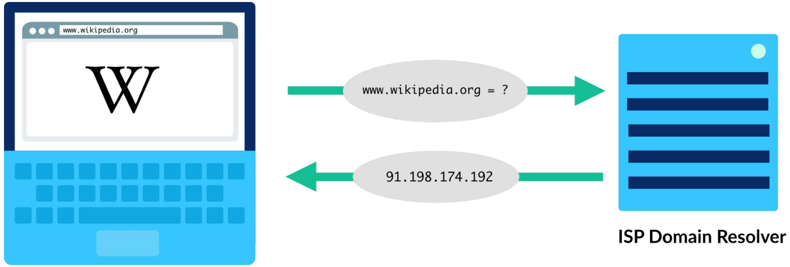
- How do we identify **people**?
  - Social security numbers
  - Passports, drivers licenses
  - Their unique fingerprints

- How can we identify **internet hosts**?
  - **Network layer:** location via **IP addresses**
    - A logical addressing system
    - 32-bit (IPV4) addressing datagrams



Identification on the Web

- How do we identify **people**?
  - Social security numbers
  - Passports, drivers licenses
  - Their unique fingerprints
- How can we identify **internet hosts**?
  - **Network layer:** location via **IP addresses**
    - A logical addressing system
    - 32-bit (IPV4) addressing datagrams
  - **What you care about:** the domain name
    - E.g., `www.wikipedia.org`

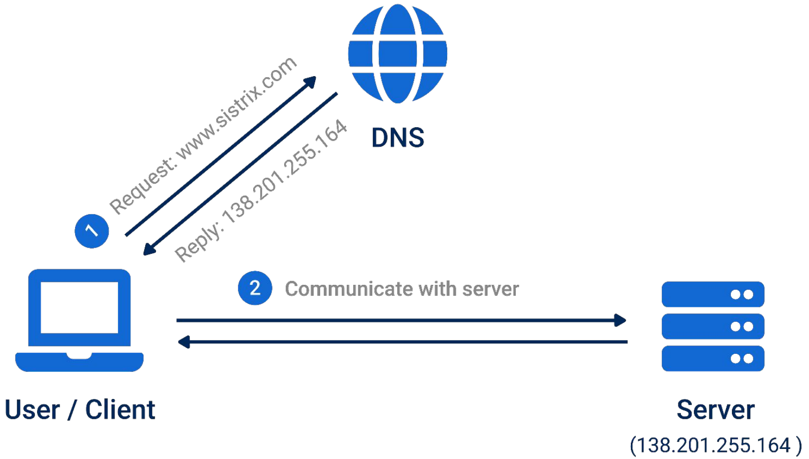


The Domain Name System

- **Distributed database** implemented in hierarchy of many name servers

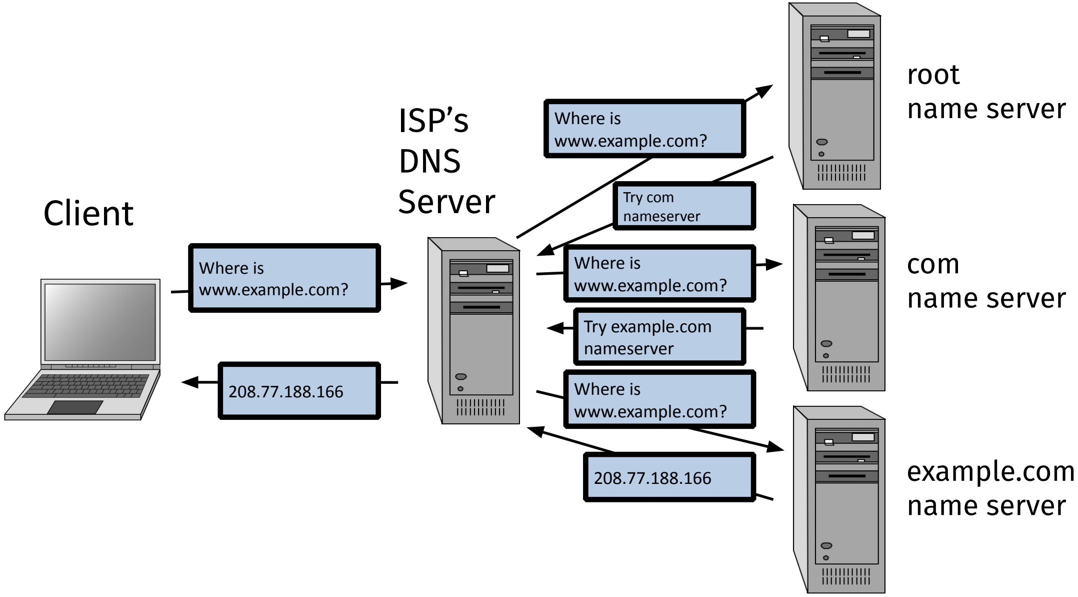
- **Application-layer protocol:**
  - Hosts and domain name servers communicate to resolve **domain names**
    - Address–name translation

- **Result:** user requests **domain name**
  - But their host really gets its **IP address**
  - Convenient!



DNS Name Resolution

- **Zone:** collection of connected nodes with the same authoritative DNS server
- Resolution method when answer **not in cache:**



Reducing Resolution Latency

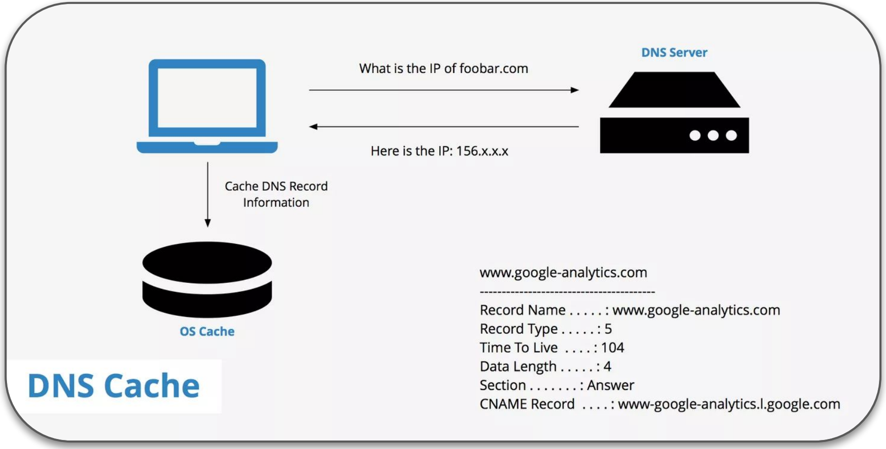
- **How can we optimize DNS resolution?**
  - **???**

Reducing Resolution Latency

- **How can we optimize DNS resolution?**
  - **Cache look-ups** to **amortize initial look-up**, reduce system load

- Temporal locality of requests:
  - `www.espn.com/page1`
  - `www.espn.com/page2`

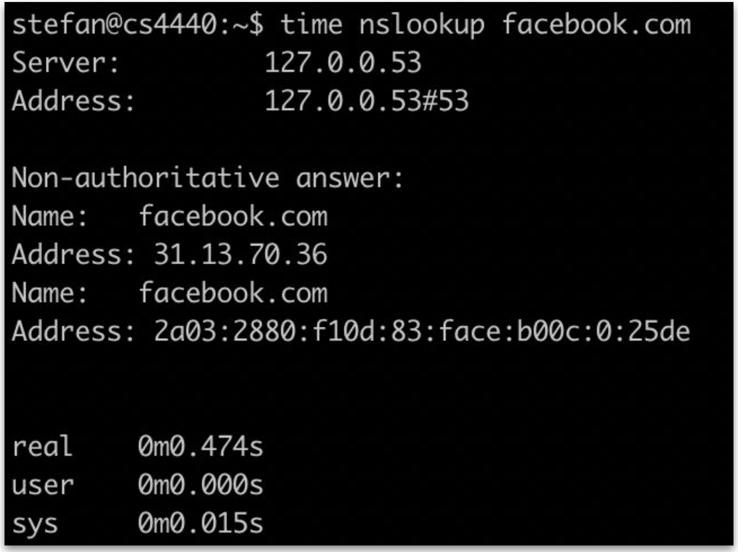
- Popular destinations:
  - `google.com`
  - `Facebook.com`



DNS Server
What is the IP of foobar.com
Here is the IP: 156.x.x.x
Cache DNS Record Information
OS Cache
DNS Cache

```
www.google-analytics.com
----------------------------------------
Record Name . . . . . : www.google-analytics.com
Record Type . . . . . : 5
Time To Live  . . . . : 104
Data Length . . . . . : 4
Section . . . . . . . : Answer
CNAME Record  . . . . : www-google-analytics.l.google.com
```

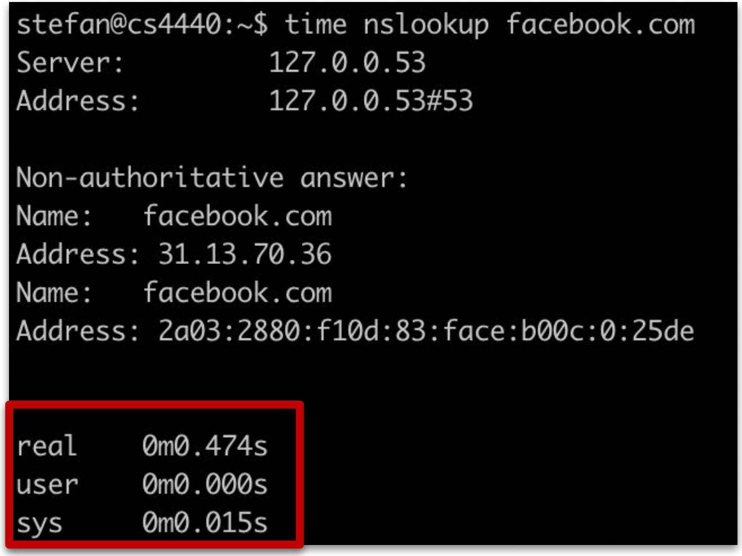
Reducing Resolution Latency



```
stefan@cs4440:~$ time nslookup facebook.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   facebook.com
Address: 31.13.70.36
Name:   facebook.com
Address: 2a03:2880:f10d:83:face:b00c:0:25de


real    0m0.474s
user    0m0.000s
sys     0m0.015s
```

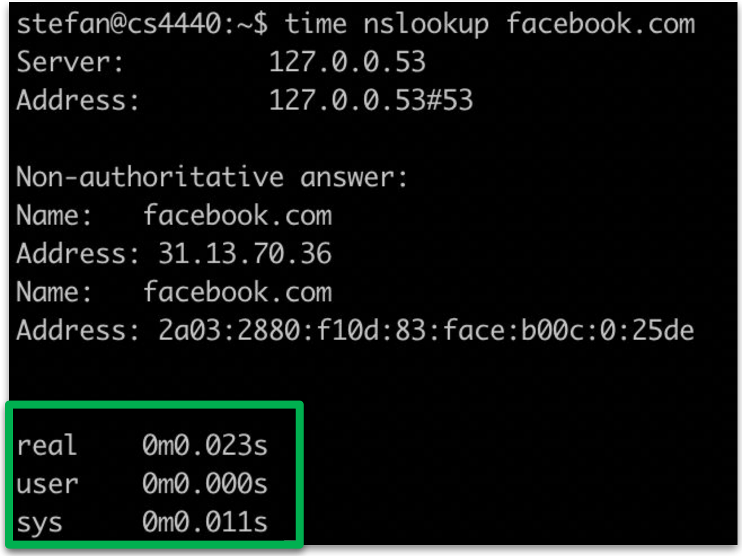
First Lookup (**non-cached**)

Reducing Resolution Latency



```
stefan@cs4440:~$ time nslookup facebook.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   facebook.com
Address: 31.13.70.36
Name:   facebook.com
Address: 2a03:2880:f10d:83:face:b00c:0:25de


real    0m0.474s
user    0m0.000s
sys     0m0.015s
```

First Lookup (**non-cached**)

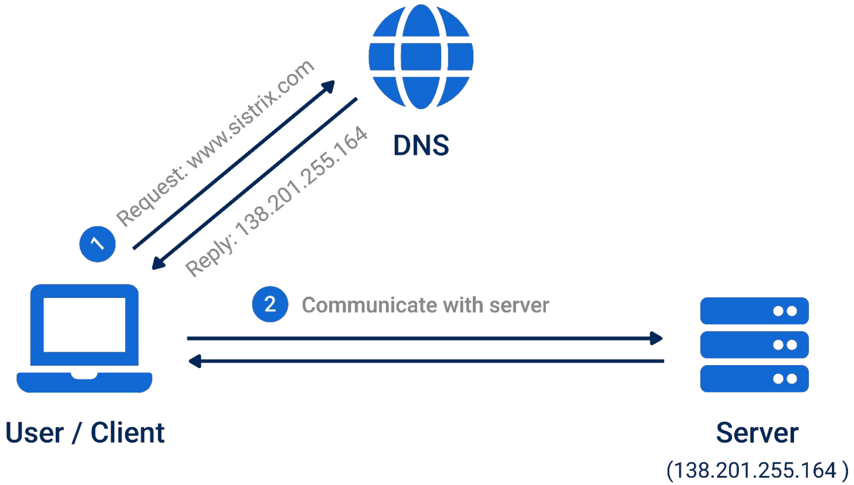


```
stefan@cs4440:~$ time nslookup facebook.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   facebook.com
Address: 31.13.70.36
Name:   facebook.com
Address: 2a03:2880:f10d:83:face:b00c:0:25de


real    0m0.023s
user    0m0.000s
sys     0m0.011s
```

Second Lookup (**cached**)

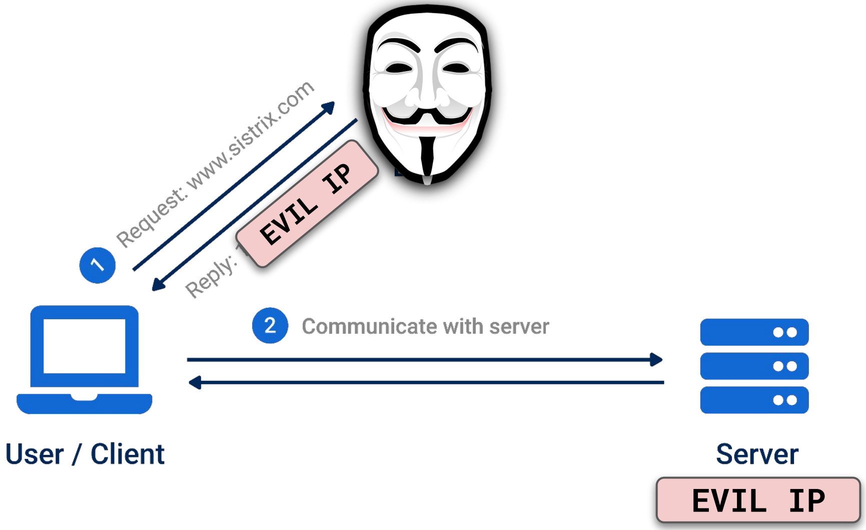
Attacking DNS

- What can an attacker do if they **control a DNS server**?



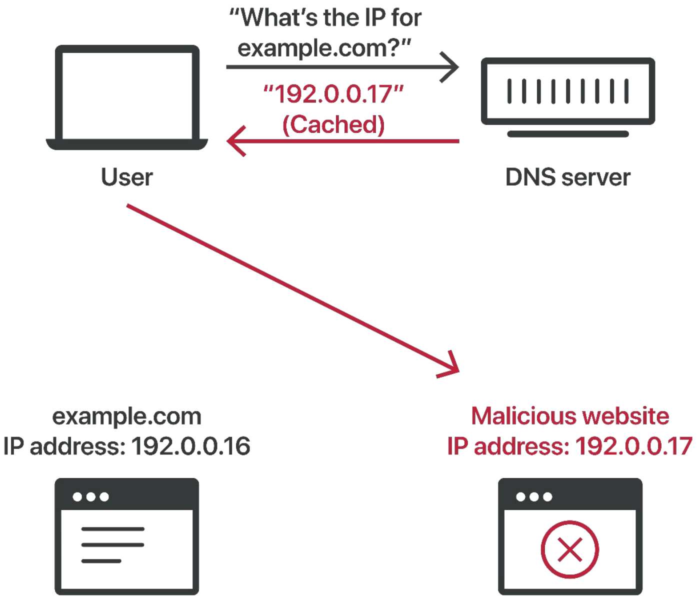
Attacking DNS

- What can an attacker do if they **control a DNS server**?
  - **Control how users of that DNS server view the internet!**
    - Assuming they use domain names



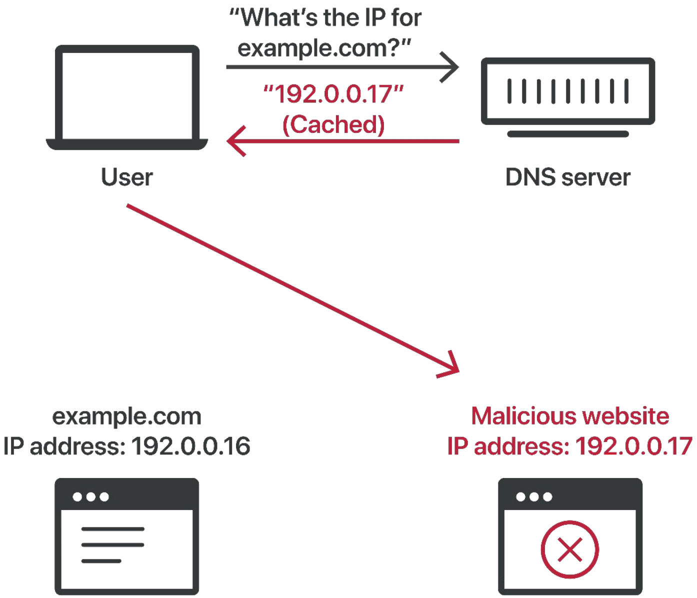
Case Study: DNS Cache Poisoning

- **Attack:** pre-empt DNS lookup by **injecting malicious cache** contents
  - Exploits DNS lookup optimization!



Case Study: DNS Cache Poisoning

- **Attack:** pre-empt DNS lookup by **injecting malicious cache** contents
  - Exploits DNS lookup optimization!
- Victim performs cache lookup, instead gets malicious domain IP
  - Attacker can **redirect** the victim's browser to the malicious website

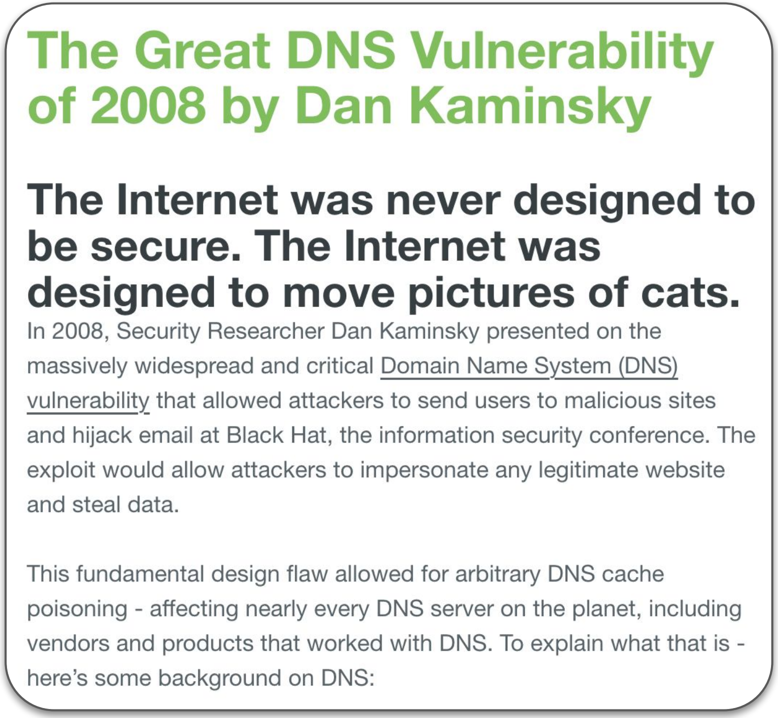


Case Study: DNS Cache Poisoning

- **Attack:** pre-empt DNS lookup by **injecting malicious cache** contents
  - Exploits DNS lookup optimization!

- Victim performs cache lookup, instead gets malicious domain IP
  - Attacker can **redirect** the victim's browser to the malicious website

- **A massive vulnerability in 2008!**



The Great DNS Vulnerability of 2008 by Dan Kaminsky

The Internet was never designed to be secure. The Internet was designed to move pictures of cats.

In 2008, Security Researcher Dan Kaminsky presented on the massively widespread and critical Domain Name System (DNS) vulnerability that allowed attackers to send users to malicious sites and hijack email at Black Hat, the information security conference. The exploit would allow attackers to impersonate any legitimate website and steal data.

This fundamental design flaw allowed for arbitrary DNS cache poisoning - affecting nearly every DNS server on the planet, including vendors and products that worked with DNS. To explain what that is - here's some background on DNS:

Thwarting DNS Hijacking

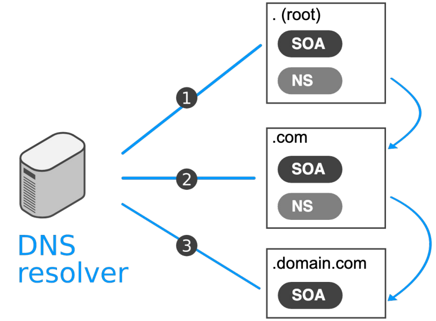
- **Attack points:**
  - Local host
  - Router
  - ISP

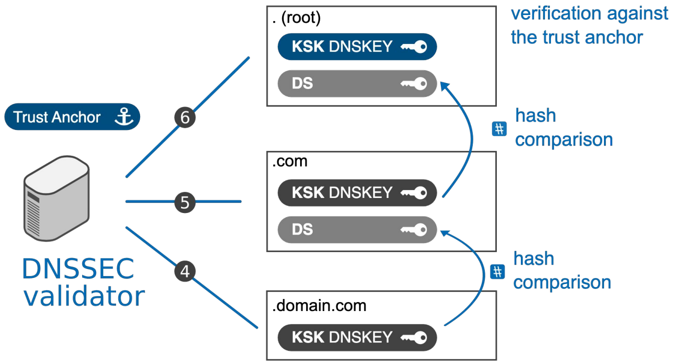
Thwarting DNS Hijacking

- **Attack points:**
  - Local host
  - Router
  - ISP

- **DNS-level** authentication
  - DNSSec
  - Public-key crypto to “sign” DNS records

- **Endpoint** authentication
  - Certify that what I am seeing really is bank.com
  - Transport Layer Security (TLS)



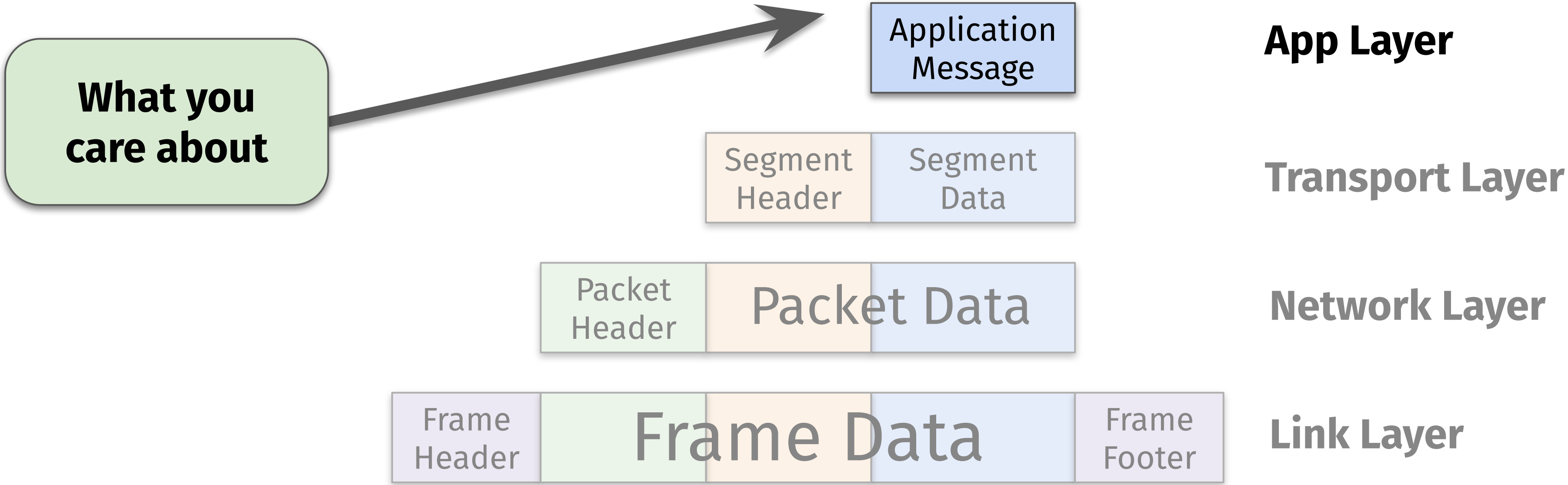


Questions?

Analyzing Network Packets

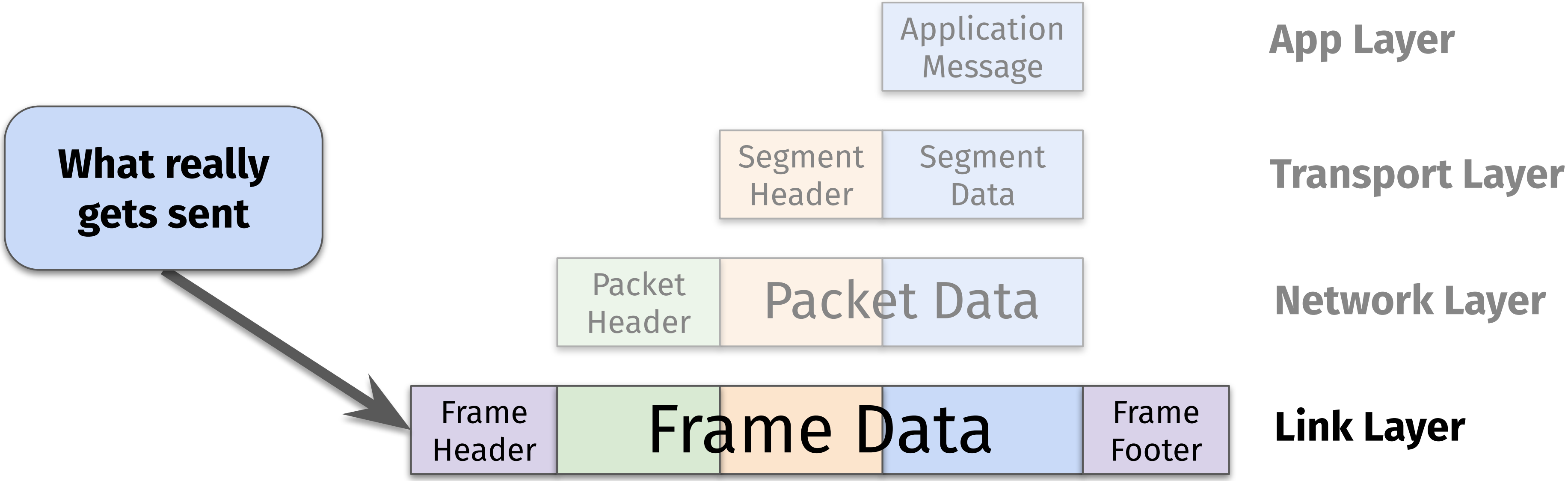
Recap: Internet Packet Encapsulation

- How packets are generated and sent



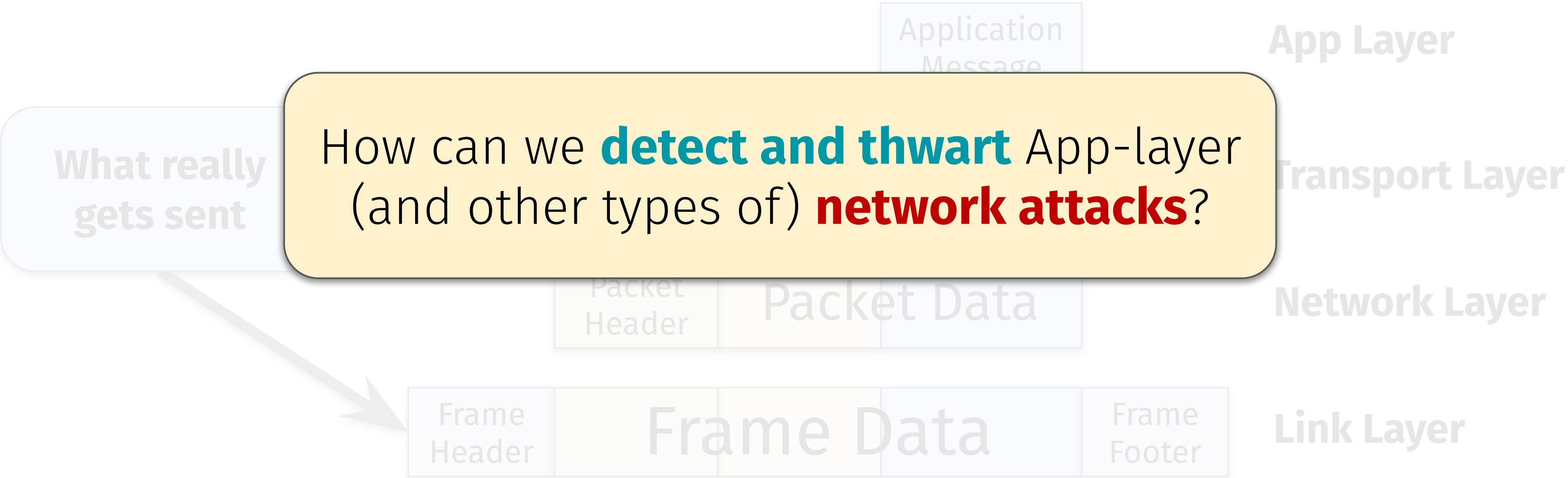
Recap: Internet Packet Encapsulation

- How packets are generated and sent



Recap: Internet Packet Encapsulation

- How packets are generated and sent



How can we **detect and thwart** App-layer (and other types of) **network attacks**?

Tools of the Trade

- **Packet Analyzers:**
  - Tools for dissecting network packets
- Packet Analyzers allow you to:
  - Identify unusual packets
  - Characterize network activity
  - Pinpoint **malicious traffic**
- The basis of modern-day network security (e.g., firewalls, antivirus)

Familiarity with packet analysis tools?

I eat NetSec CTF challenges like a kid eats candy on Halloween. 🧙

0%

Some (e.g., Wireshark, DPKT, Scapy, or something else)

0%

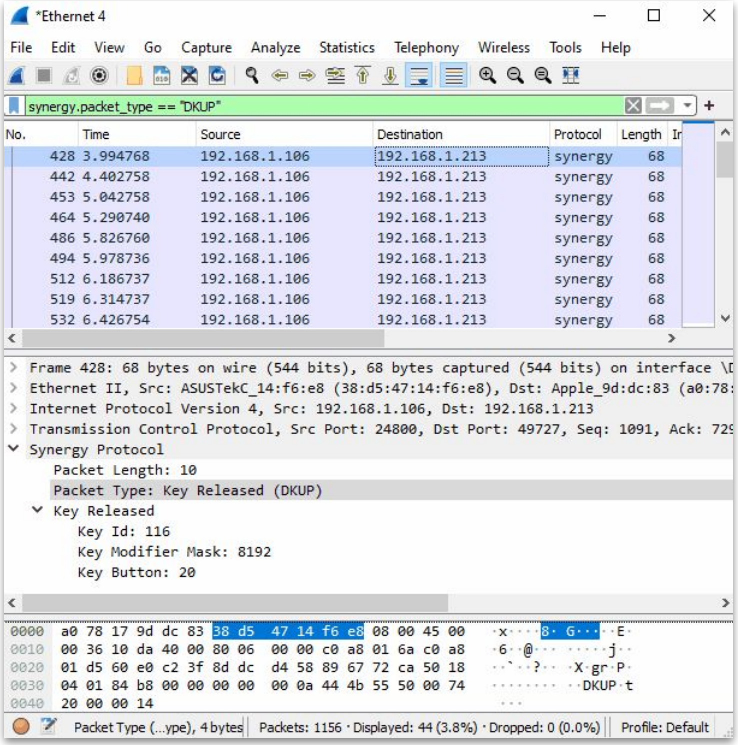
None (but that's totally okay!)

0%

Tools of the Trade: Wireshark

- A "graphical interface" for manual packet analysis
  - Completely **open-source and free**

- General workflow:
  - Load up a PCAP (packet capture)
  - Wireshark will display each packet
  - Inspect particular fields of interest



synergy.packet_type == "DKUP"

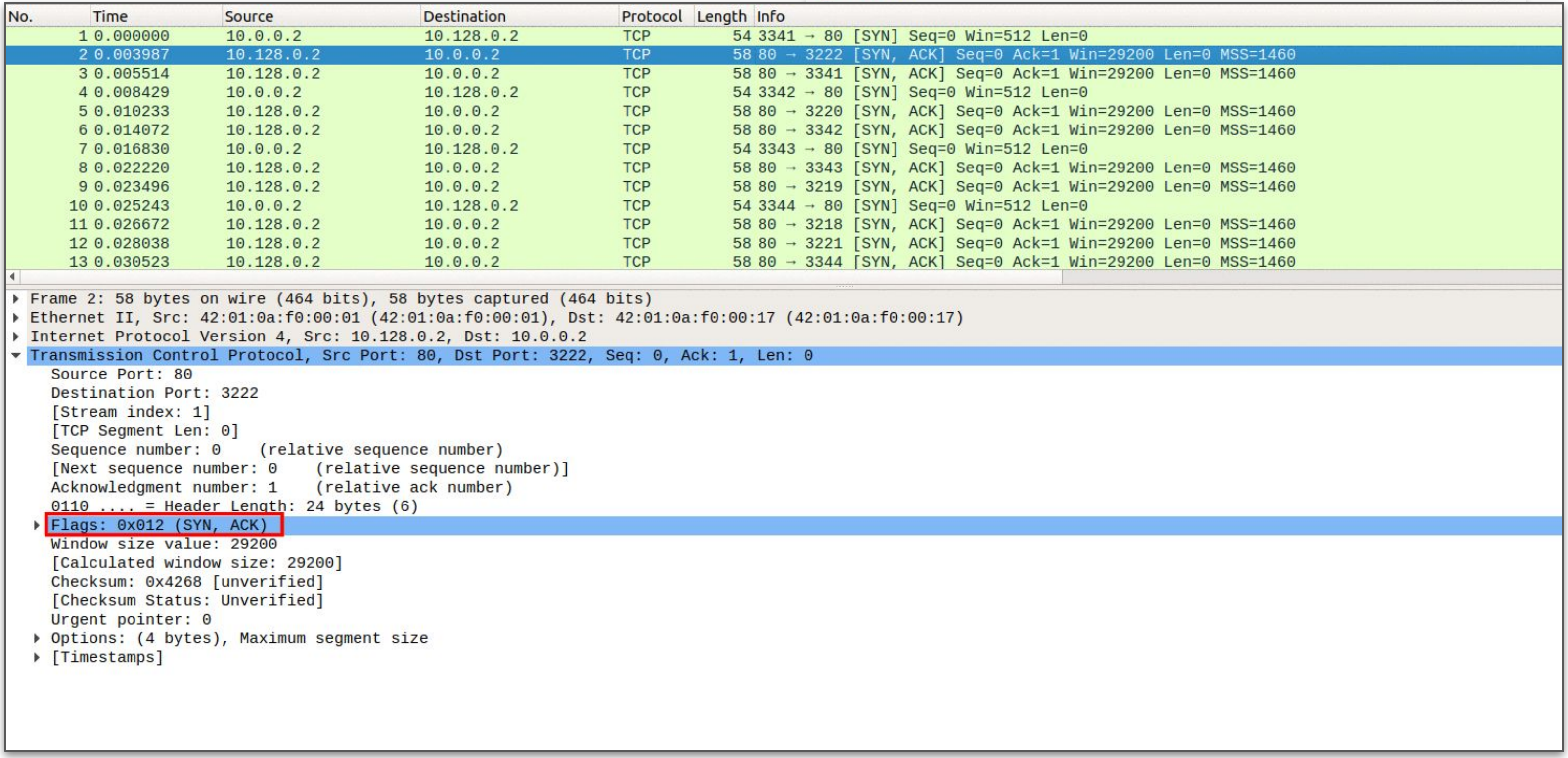
No.	Time	Source	Destination	Protocol	Length	Ir
428	3.994768	192.168.1.106	192.168.1.213	synergy	68	
442	4.402758	192.168.1.106	192.168.1.213	synergy	68	
453	5.042758	192.168.1.106	192.168.1.213	synergy	68	
464	5.290740	192.168.1.106	192.168.1.213	synergy	68	
486	5.826760	192.168.1.106	192.168.1.213	synergy	68	
494	5.978736	192.168.1.106	192.168.1.213	synergy	68	
512	6.186737	192.168.1.106	192.168.1.213	synergy	68	
519	6.314737	192.168.1.106	192.168.1.213	synergy	68	
532	6.426754	192.168.1.106	192.168.1.213	synergy	68	

```
> Frame 428: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface \D
> Ethernet II, Src: ASUSTekC_14:f6:e8 (38:d5:47:14:f6:e8), Dst: Apple_9d:dc:83 (a0:78:
> Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.213
> Transmission Control Protocol, Src Port: 24800, Dst Port: 49727, Seq: 1091, Ack: 729
v Synergy Protocol
    Packet Length: 10
    Packet Type: Key Released (DKUP)
  v Key Released
      Key Id: 116
      Key Modifier Mask: 8192
      Key Button: 20
```

```
0000  a0 78 17 9d dc 83 38 d5  47 14 f6 e8 08 00 45 00   ·x····8· G·····E·
0010  00 36 10 da 40 00 80 06  00 00 c0 a8 01 6a c0 a8   ·6··@··· ·····j··
0020  01 d5 60 e0 c2 3f 8d dc  d4 58 89 67 72 ca 50 18   ··`··?·· ·X·gr·P·
0030  04 01 84 b8 00 00 00 00  00 0a 44 4b 55 50 00 74   ········ ··DKUP·t
0040  20 00 00 14                                         ···
```

Packet Type (...ype), 4 bytes | Packets: 1156 · Displayed: 44 (3.8%) · Dropped: 0 (0.0%) | Profile: Default

Tools of the Trade: Wireshark

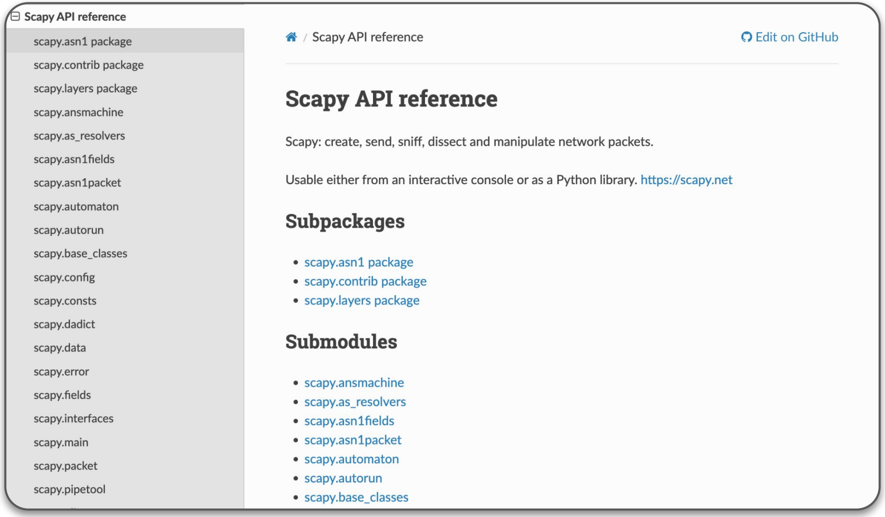


```
No.   Time      Source       Destination  Protocol Length Info
    1 0.000000  10.0.0.2     10.128.0.2   TCP          54 3341 → 80 [SYN] Seq=0 Win=512 Len=0
    2 0.003987  10.128.0.2   10.0.0.2     TCP          58 80 → 3222 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
    3 0.005514  10.128.0.2   10.0.0.2     TCP          58 80 → 3341 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
    4 0.008429  10.0.0.2     10.128.0.2   TCP          54 3342 → 80 [SYN] Seq=0 Win=512 Len=0
    5 0.010233  10.128.0.2   10.0.0.2     TCP          58 80 → 3220 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
    6 0.014072  10.128.0.2   10.0.0.2     TCP          58 80 → 3342 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
    7 0.016830  10.0.0.2     10.128.0.2   TCP          54 3343 → 80 [SYN] Seq=0 Win=512 Len=0
    8 0.022220  10.128.0.2   10.0.0.2     TCP          58 80 → 3343 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
    9 0.023496  10.128.0.2   10.0.0.2     TCP          58 80 → 3219 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
   10 0.025243  10.0.0.2     10.128.0.2   TCP          54 3344 → 80 [SYN] Seq=0 Win=512 Len=0
   11 0.026672  10.128.0.2   10.0.0.2     TCP          58 80 → 3218 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
   12 0.028038  10.128.0.2   10.0.0.2     TCP          58 80 → 3221 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
   13 0.030523  10.128.0.2   10.0.0.2     TCP          58 80 → 3344 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
```

```
▸ Frame 2: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)
▸ Ethernet II, Src: 42:01:0a:f0:00:01 (42:01:0a:f0:00:01), Dst: 42:01:0a:f0:00:17 (42:01:0a:f0:00:17)
▸ Internet Protocol Version 4, Src: 10.128.0.2, Dst: 10.0.0.2
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 3222, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 3222
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0110 .... = Header Length: 24 bytes (6)
  ▸ Flags: 0x012 (SYN, ACK)
    Window size value: 29200
    [Calculated window size: 29200]
    Checksum: 0x4268 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (4 bytes), Maximum segment size
  ▸ [Timestamps]
```

Tools of the Trade: Scapy

- Python API for programmatic packet capture and analysis
  - Think of it as “Wireshark in API form”
  - **Project 4:** you will use Scapy to write your own packet analysis scripts



Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

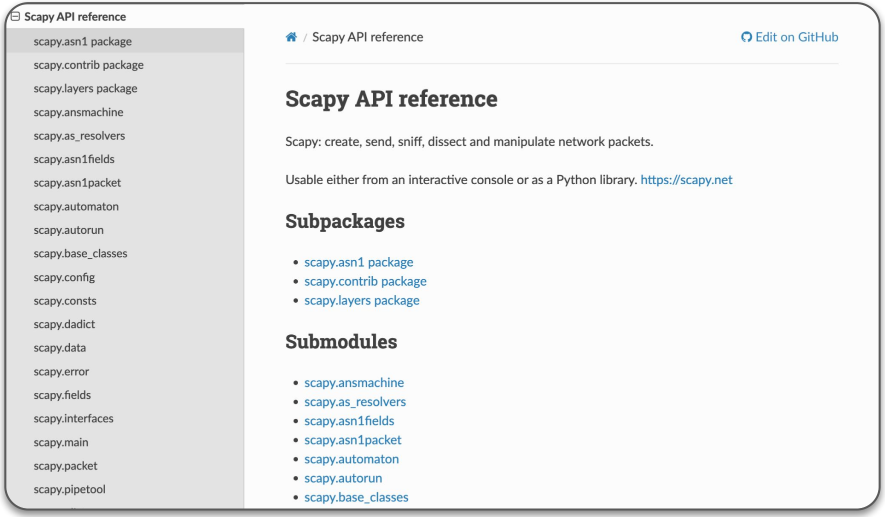
Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Tools of the Trade: Scapy

- Python API for programmatic packet capture and analysis
  - Think of it as “Wireshark in API form”
  - **Project 4:** you will use Scapy to write your own packet analysis scripts

- We’ll provide the PCAP traces…
  - You’ll write code to analyze them!
  - **Examples:**
    - **Detecting attacks** on a network
    - Finding **user credentials**
    - Sniffing a user’s **browsing history**



⊟ Scapy API reference

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package
- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes
- scapy.config
- scapy.consts
- scapy.dadict
- scapy.data
- scapy.error
- scapy.fields
- scapy.interfaces
- scapy.main
- scapy.packet
- scapy.pipetool

/ Scapy API reference

Edit on GitHub

Scapy API reference

Scapy: create, send, sniff, dissect and manipulate network packets.

Usable either from an interactive console or as a Python library. https://scapy.net

Subpackages

- scapy.asn1 package
- scapy.contrib package
- scapy.layers package

Submodules

- scapy.ansmachine
- scapy.as_resolvers
- scapy.asn1fields
- scapy.asn1packet
- scapy.automaton
- scapy.autorun
- scapy.base_classes

Questions?

Next time on CS 4440...

Transport, Network, Link, and Physical Layer Attacks