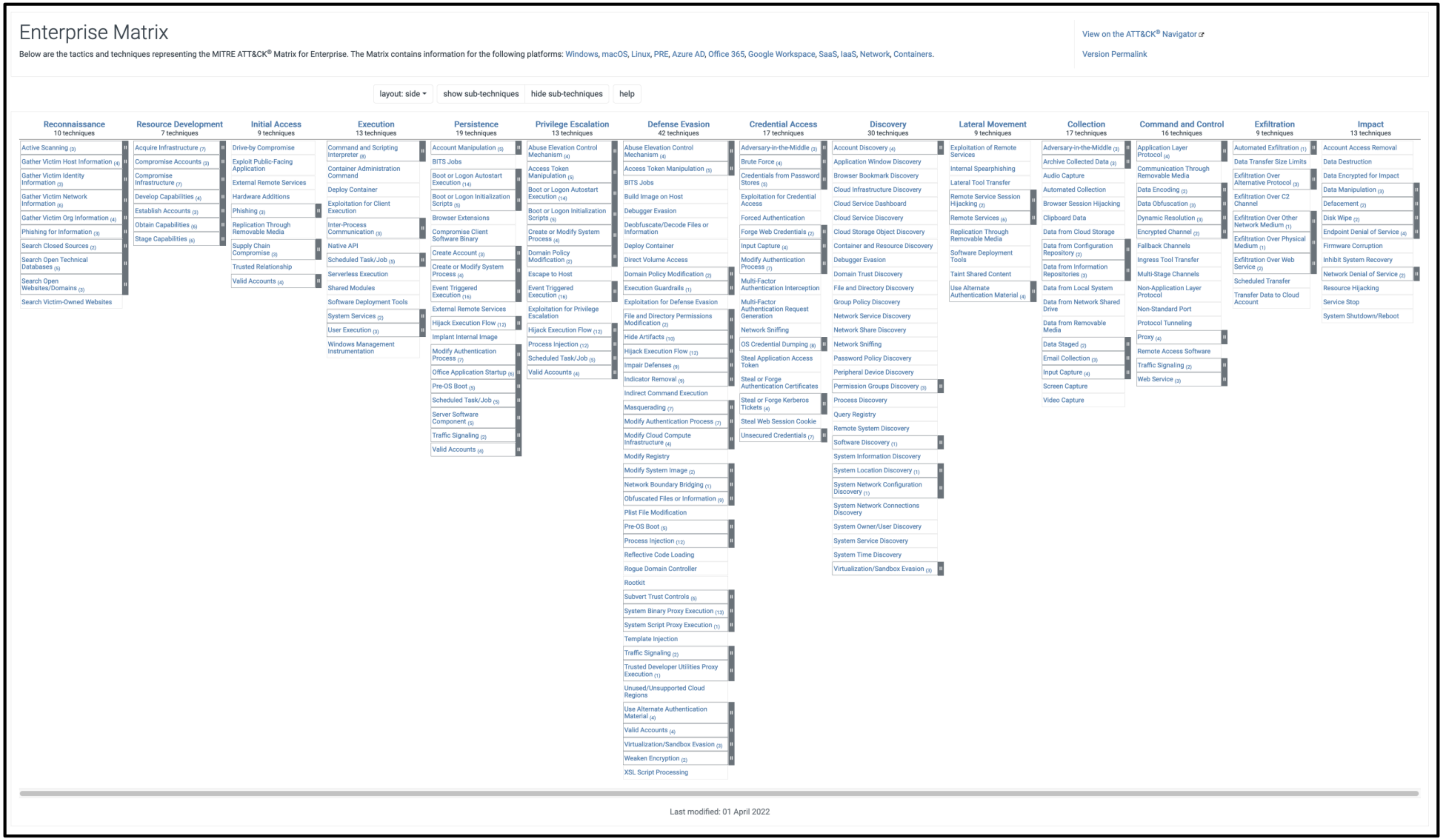
Threats and Countermeasures

Lecture 03: Reconaissance

COMP-5830/-6830
Spring 2025

MITRE ATT&CK Framework



Enterprise Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the following platforms: Windows, macOS, Linux, PRE, Azure AD, Office 365, Google Workspace, SaaS, IaaS, Network, Containers.

View on the ATT&CK® Navigator

Version Permalink

Reconnaissance
10 techniques

- Active Scanning (3)
- Gather Victim Host Information (4)
- Gather Victim Identity Information (3)
- Gather Victim Network Information (6)
- Gather Victim Org Information (4)
- Phishing for Information (3)
- Search Closed Sources (2)
- Search Open Technical Databases (5)
- Search Open Websites/Domains (3)
- Search Victim-Owned Websites

Resource Development
7 techniques

- Acquire Infrastructure (7)
- Compromise Accounts (3)
- Compromise Infrastructure (7)
- Develop Capabilities (4)
- Establish Accounts (3)
- Obtain Capabilities (6)
- Stage Capabilities (6)

Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (3)
- Replication Through Removable Media
- Supply Chain Compromise (3)
- Trusted Relationship
- Valid Accounts (4)

Execution
13 techniques

- Command and Scripting Interpreter (8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (3)
- Native API
- Scheduled Task/Job (5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (2)
- User Execution (3)
- Windows Management Instrumentation

Persistence
19 techniques

- Account Manipulation (5)
- BITS Jobs
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (3)
- Create or Modify System Process (4)
- Event Triggered Execution (16)
- External Remote Services
- Hijack Execution Flow (12)
- Implant Internal Image
- Modify Authentication Process (7)
- Office Application Startup (6)
- Pre-OS Boot (5)
- Scheduled Task/Job (5)
- Server Software Component (5)
- Traffic Signaling (2)
- Valid Accounts (4)

Privilege Escalation
13 techniques

- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Create or Modify System Process (4)
- Domain Policy Modification (2)
- Escape to Host
- Event Triggered Execution (16)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (12)
- Process Injection (12)
- Scheduled Task/Job (5)
- Valid Accounts (4)

Defense Evasion
42 techniques

- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (2)
- Execution Guardrails (1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (2)
- Hide Artifacts (10)
- Hijack Execution Flow (12)
- Impair Defenses (9)
- Indicator Removal (9)
- Indirect Command Execution
- Masquerading (7)
- Modify Authentication Process (7)
- Modify Cloud Compute Infrastructure (4)
- Modify Registry
- Modify System Image (2)
- Network Boundary Bridging (1)
- Obfuscated Files or Information (9)
- Plist File Modification
- Pre-OS Boot (5)
- Process Injection (12)
- Reflective Code Loading
- Rogue Domain Controller
- Rootkit
- Subvert Trust Controls (6)
- System Binary Proxy Execution (13)
- System Script Proxy Execution (1)
- Template Injection
- Traffic Signaling (2)
- Trusted Developer Utilities Proxy Execution (1)
- Unused/Unsupported Cloud Regions
- Use Alternate Authentication Material (4)
- Valid Accounts (4)
- Virtualization/Sandbox Evasion (3)
- Weaken Encryption (2)
- XSL Script Processing

Credential Access
17 techniques

- Adversary-in-the-Middle (3)
- Brute Force (4)
- Credentials from Password Stores (5)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (2)
- Input Capture (4)
- Modify Authentication Process (7)
- Multi-Factor Authentication Interception
- Multi-Factor Authentication Request Generation
- Network Sniffing
- OS Credential Dumping (8)
- Steal Application Access Token
- Steal or Forge Authentication Certificates
- Steal or Forge Kerberos Tickets (4)
- Steal Web Session Cookie
- Unsecured Credentials (7)

Discovery
30 techniques

- Account Discovery (4)
- Application Window Discovery
- Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Network Service Discovery
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (3)
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery (1)
- System Information Discovery
- System Location Discovery (1)
- System Network Configuration Discovery (1)
- System Network Connections Discovery
- System Owner/User Discovery
- System Service Discovery
- System Time Discovery
- Virtualization/Sandbox Evasion (3)

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (2)
- Remote Services (6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (4)

Collection
17 techniques

- Adversary-in-the-Middle (3)
- Archive Collected Data (3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (2)
- Data from Information Repositories (3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (2)
- Email Collection (3)
- Input Capture (4)
- Screen Capture
- Video Capture

Command and Control
16 techniques

- Application Layer Protocol (4)
- Communication Through Removable Media
- Data Encoding (2)
- Data Obfuscation (3)
- Dynamic Resolution (3)
- Encrypted Channel (2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (4)
- Remote Access Software
- Traffic Signaling (2)
- Web Service (3)

Exfiltration
9 techniques

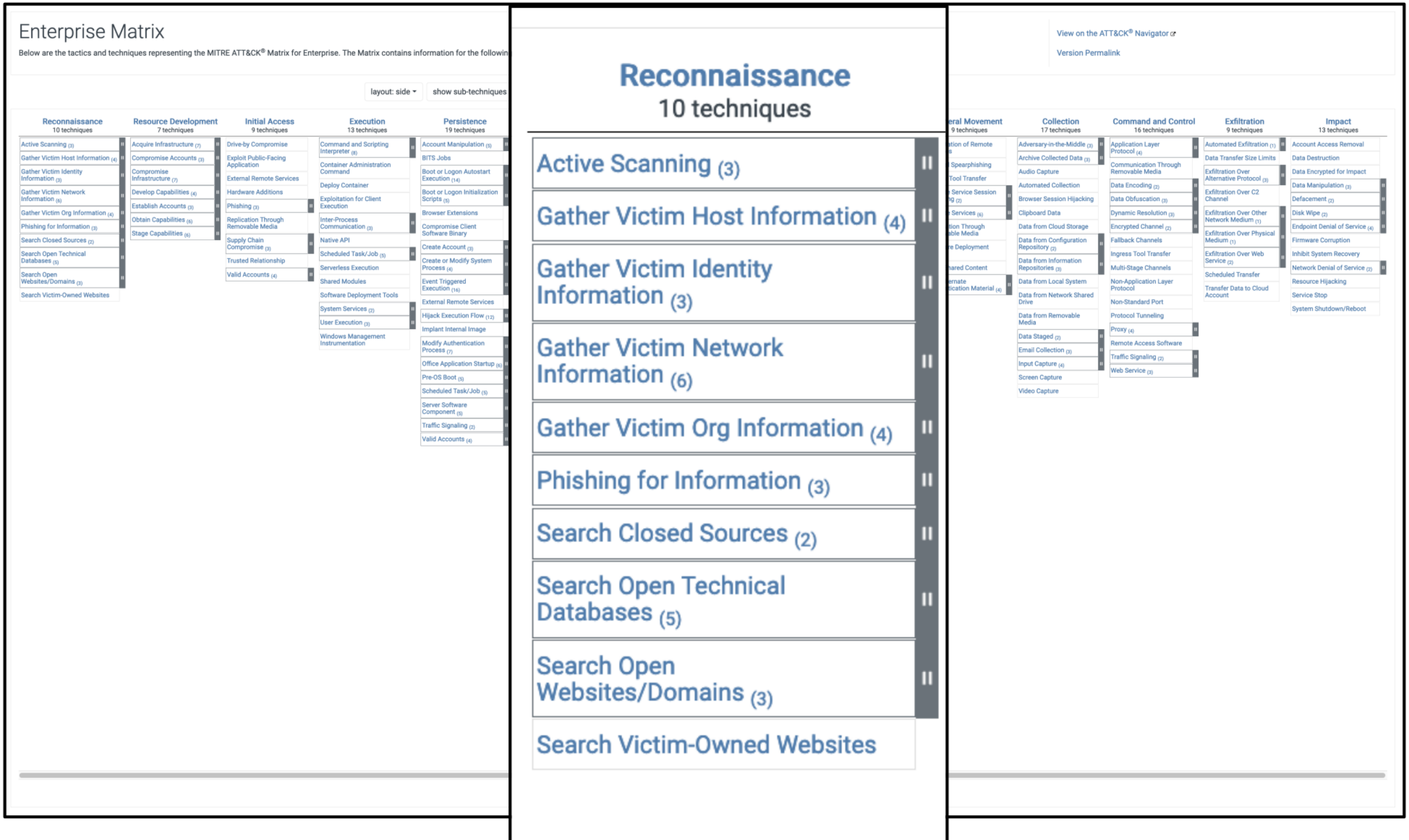
- Automated Exfiltration (1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (1)
- Exfiltration Over Physical Medium (1)
- Exfiltration Over Web Service (2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (3)
- Defacement (2)
- Disk Wipe (2)
- Endpoint Denial of Service (4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

Last modified: 01 April 2022

MITRE ATT&CK Framework



Enterprise Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise. The Matrix contains information for the followin

View on the ATT&CK® Navigator

Version Permalink

layout: side ▾ show sub-techniques

Reconnaissance
10 techniques

- Active Scanning (3)
- Gather Victim Host Information (4)
- Gather Victim Identity Information (3)
- Gather Victim Network Information (6)
- Gather Victim Org Information (4)
- Phishing for Information (3)
- Search Closed Sources (2)
- Search Open Technical Databases (5)
- Search Open Websites/Domains (3)
- Search Victim-Owned Websites

Resource Development
7 techniques

- Acquire Infrastructure (7)
- Compromise Accounts (3)
- Compromise Infrastructure (7)
- Develop Capabilities (4)
- Establish Accounts (3)
- Obtain Capabilities (6)
- Stage Capabilities (6)

Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (3)
- Replication Through Removable Media
- Supply Chain Compromise (3)
- Trusted Relationship
- Valid Accounts (4)

Execution
13 techniques

- Command and Scripting Interpreter (8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (3)
- Native API
- Scheduled Task/Job (5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (2)
- User Execution (3)
- Windows Management Instrumentation

Persistence
19 techniques

- Account Manipulation (5)
- BITS Jobs
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (3)
- Create or Modify System Process (4)
- Event Triggered Execution (16)
- External Remote Services
- Hijack Execution Flow (12)
- Implant Internal Image
- Modify Authentication Process (7)
- Office Application Startup (6)
- Pre-OS Boot (5)
- Scheduled Task/Job (5)
- Server Software Component (5)
- Traffic Signaling (2)
- Valid Accounts (4)

eral Movement
9 techniques

- ation of Remote s
- Spearphishing
- Tool Transfer
- Service Session ng (2)
- Services (6)
- tion Through able Media
- re Deployment
- hared Content
- ernate tication Material (4)

Collection
17 techniques

- Adversary-in-the-Middle (3)
- Archive Collected Data (3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (2)
- Data from Information Repositories (3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (2)
- Email Collection (3)
- Input Capture (4)
- Screen Capture
- Video Capture

Command and Control
16 techniques

- Application Layer Protocol (4)
- Communication Through Removable Media
- Data Encoding (2)
- Data Obfuscation (3)
- Dynamic Resolution (3)
- Encrypted Channel (2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (4)
- Remote Access Software
- Traffic Signaling (2)
- Web Service (3)

Exfiltration
9 techniques

- Automated Exfiltration (1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (1)
- Exfiltration Over Physical Medium (1)
- Exfiltration Over Web Service (2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (3)
- Defacement (2)
- Disk Wipe (2)
- Endpoint Denial of Service (4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

Reconnaissance
10 techniques

- Active Scanning (3)
- Gather Victim Host Information (4)
- Gather Victim Identity Information (3)
- Gather Victim Network Information (6)
- Gather Victim Org Information (4)
- Phishing for Information (3)
- Search Closed Sources (2)
- Search Open Technical Databases (5)
- Search Open Websites/Domains (3)
- Search Victim-Owned Websites

Reconnaissance

Reconnaissance consists of techniques that involve adversaries actively or passively gathering information that can be used to support targeting. Such information may include details of the victim organization, infrastructure, or staff/personnel. This information can be leveraged by the adversary to aid in other phases of the adversary lifecycle, such as using gathered information to plan and execute Initial Access, to scope and prioritize post-compromise objectives, or to drive and lead further Reconnaissance efforts.

Reconnaissance (simplified)

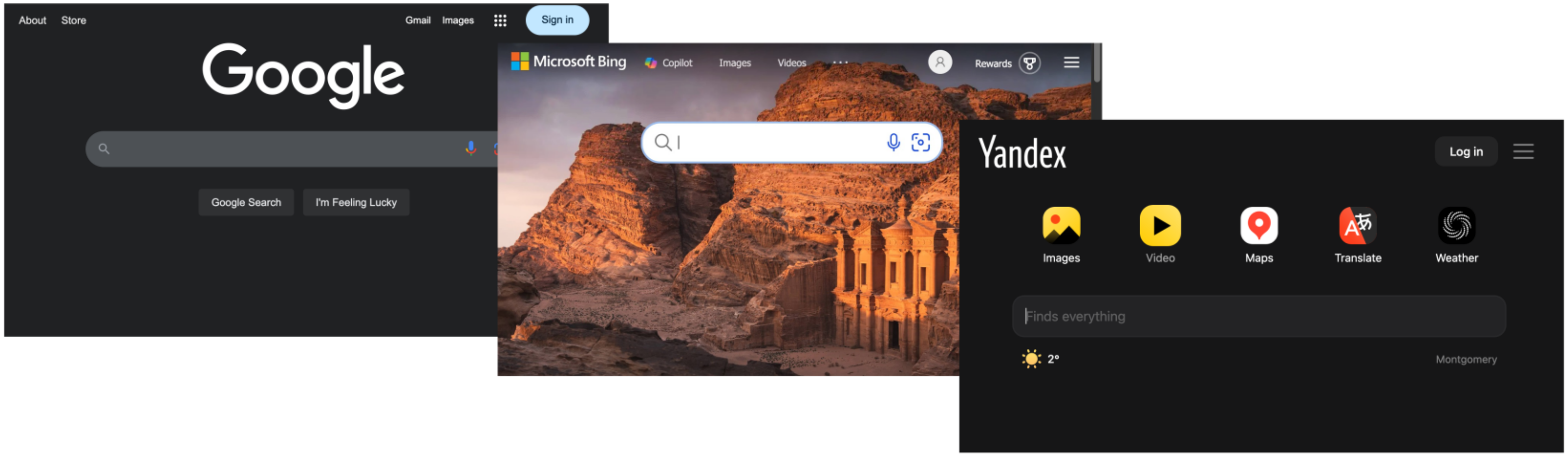
- Actively or passively gathering information on the target
  - Details organization, infrastructure, or staff/personnel
- Informs the conduct of future actions
- Should be highly scoped based on end-objective

Organizational Recon

- Where is the target’s electronic infrastructure?
  - AWS? Google Cloud? On-Prem? PoP?
- Where is the target physically?
  - Foreign country? 900 miles away? 15 miles away?
- Who does the target partner with?
  - Service providers, contractors, suppliers, shipping, etc
- What are their work cycles?
  - When are they busy? What are their holiday plans?
- Who does what?
  - Who is their CEO? Who is their on-call IT person? Who has access to the end-objective?

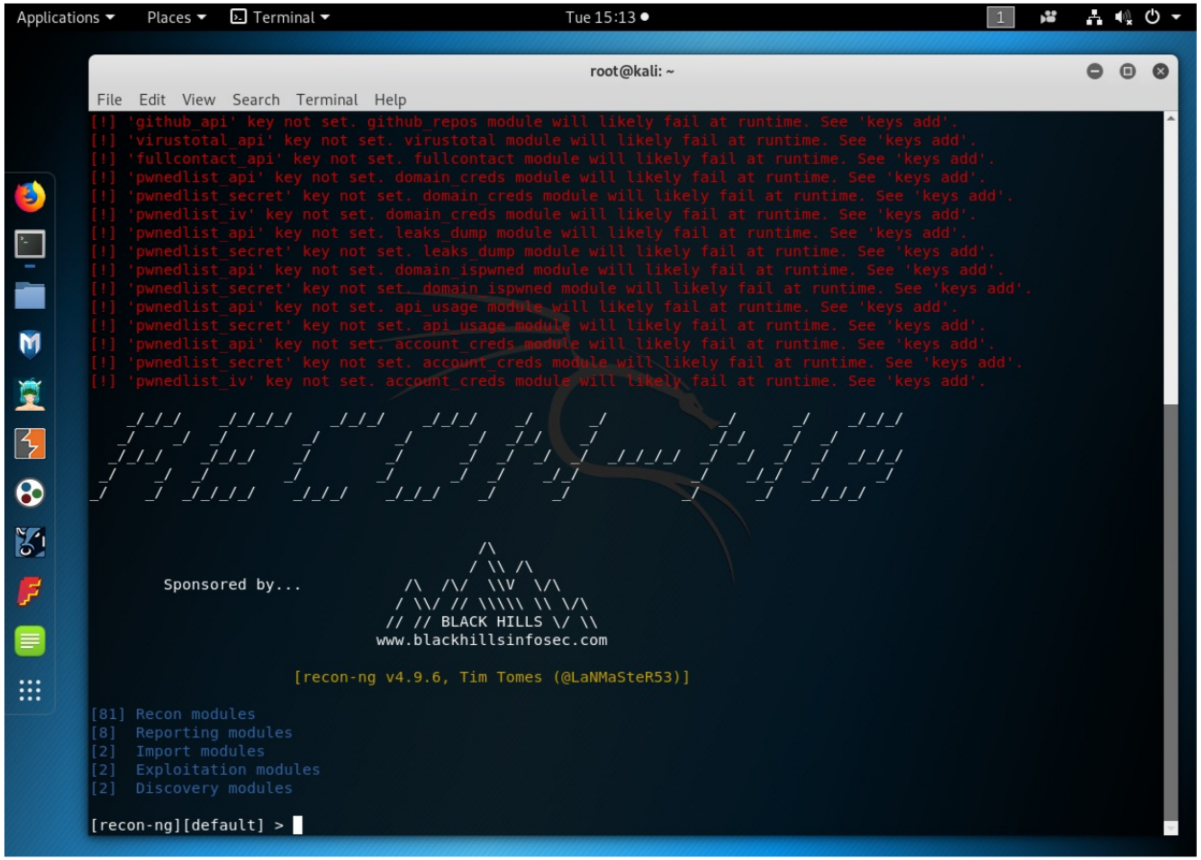
Open-Source Intelligence (OSINT)

- Many, many, many places data can live
  - Search engines
  - Social Media
  - EDGAR Database (SEC filings (not football))



Semi-Automated OSINT

- Some tools are able to aggregate data feeds
- Review and analysis **can not** be automated



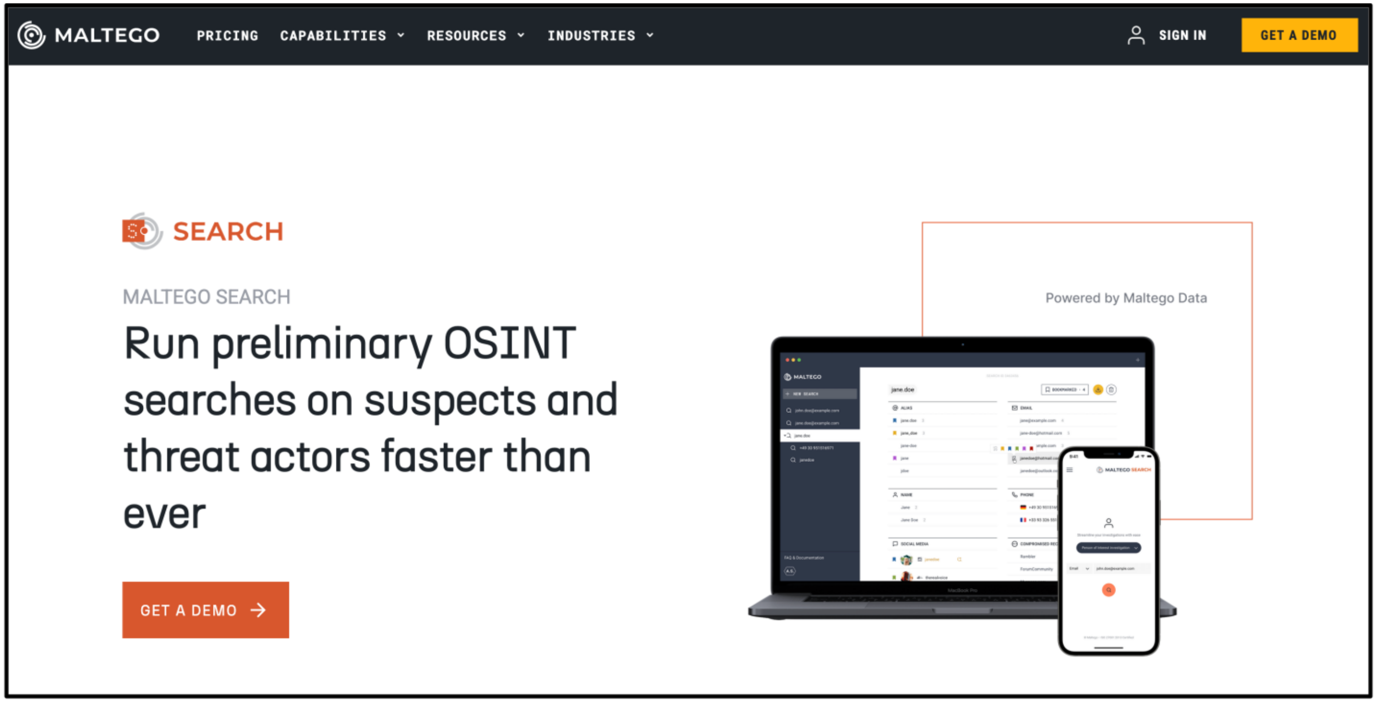
```
[!] 'github_api' key not set. github_repos module will likely fail at runtime. See 'keys add'.
[!] 'virustotal_api' key not set. virustotal module will likely fail at runtime. See 'keys add'.
[!] 'fullcontact_api' key not set. fullcontact module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_api' key not set. domain_creds module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_secret' key not set. domain_creds module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_iv' key not set. domain_creds module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_api' key not set. leaks_dump module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_secret' key not set. leaks_dump module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_api' key not set. domain_ispwned module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_secret' key not set. domain_ispwned module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_api' key not set. api_usage module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_secret' key not set. api_usage module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_api' key not set. account_creds module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_secret' key not set. account_creds module will likely fail at runtime. See 'keys add'.
[!] 'pwnedlist_iv' key not set. account_creds module will likely fail at runtime. See 'keys add'.

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
   _/    _/  _/        _/        _/      _/  _/_/    _/            _/_/    _/  _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
 _/    _/  _/        _/        _/      _/  _/    _/_/            _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/


                                          /\
                                         / \\ /\
    Sponsored by...               /\  /\/  \\V  \/\
                                 / \\/ // \\\\\ \\ \/\
                                // // BLACK HILLS \/ \\
                               www.blackhillsinfosec.com

                      [recon-ng v4.9.6, Tim Tomes (@LaNMaSteR53)]

[81] Recon modules
[8]  Reporting modules
[2]  Import modules
[2]  Exploitation modules
[2]  Discovery modules

[recon-ng][default] >
```

Aggregated OSINT

- Specialized services exist for technical data about targets
  - Shodan
  - Censys
  - PwnedList
  - (many more)

Semi-OSINT

- There are OSINT-specific companies, services, and contractors but most are non-trivially priced.

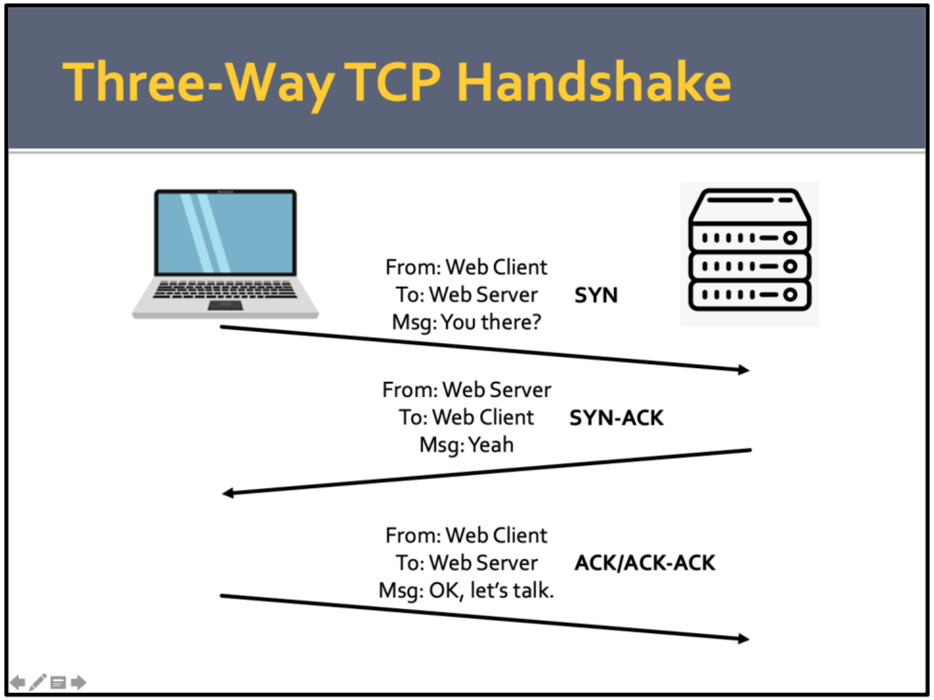


MALTEGO SEARCH

Run preliminary OSINT searches on suspects and threat actors faster than ever

Network Scanning

Network scanning is a reconnaissance technique that is used by attackers to gain information to aid them in their attacks.



Three-Way TCP Handshake

From: Web Client
To: Web Server
Msg: You there?

SYN

From: Web Server
To: Web Client
Msg: Yeah

SYN-ACK

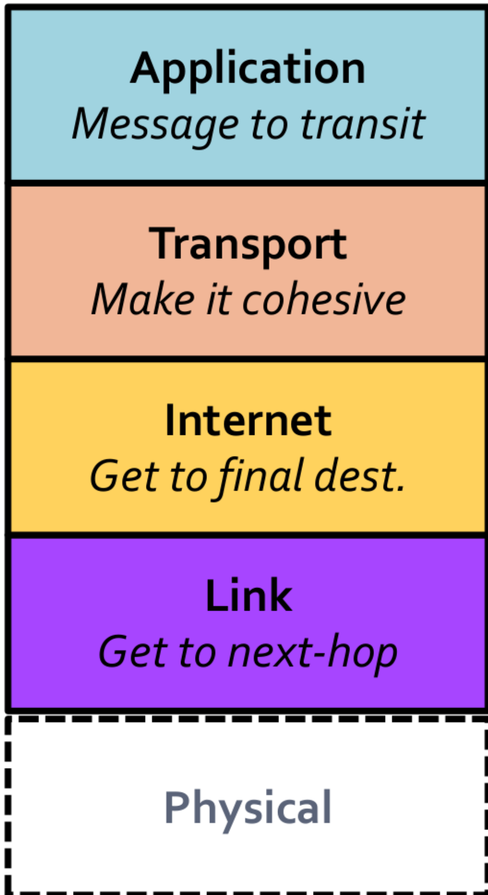
From: Web Client
To: Web Server
Msg: OK, let's talk.

ACK/ACK-ACK

TCP/IP Model

The **TCP/IP Model** is way of thinking about and conceptualizing the various protocols used in network communications.

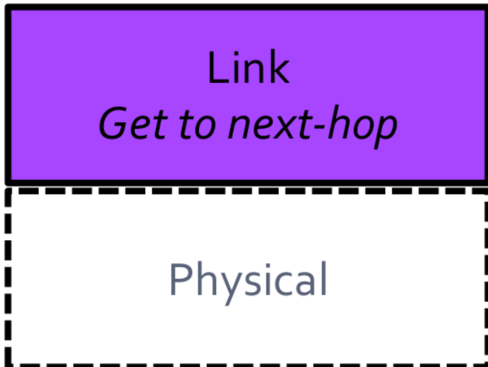
- Reduced “OSI Model”
- Specifics differ greatly based on the source, time, and writer
- Is **NOT** a perfect representation of the real-world



Link Layer

The **link layer** is responsible for addressing and transiting between endpoints on the same Local Area Network (LAN).

- Usually a relatively small physical distance
  - A room, a group of rooms, a floor, etc
- Very useful when bootstrapping to higher-level protocols

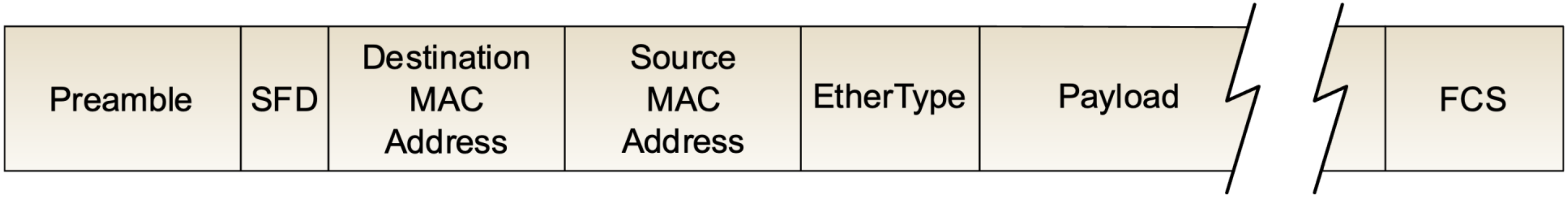


Link
Get to next-hop

Physical

Ethernet Protocol

- Media Access Control (MAC) addresses
  - DE:AD:BE:EF:4D:AD
  - “MAC address” != “Cryptographic MAC”
- Must be “locally unique” addresses
  - 3-byte manufacturer + 3-byte device ID
  - Are **NOT** globally unique



Preamble	SFD	Destination MAC Address	Source MAC Address	EtherType	Payload	FCS

Link Layer Data Flow

Ethernet Hub

- “packet repeater”
- Packet-In HW port 1
- Packet-Out HW ports 2, 3, 4

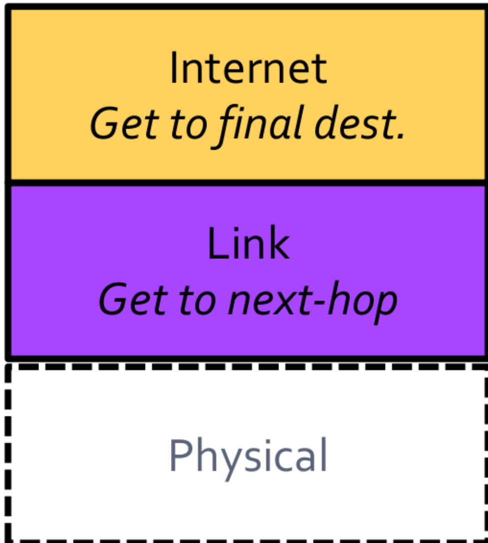
Ethernet Switch

- “packet dispatcher”

Internet Layer

The **Internet layer** is responsible for addressing and transiting between endpoints on *different* LANs connected via a Wide Area Network (WAN).

- Requires a shared addressing and encoding scheme
- Acts as the "Internet Interstate"



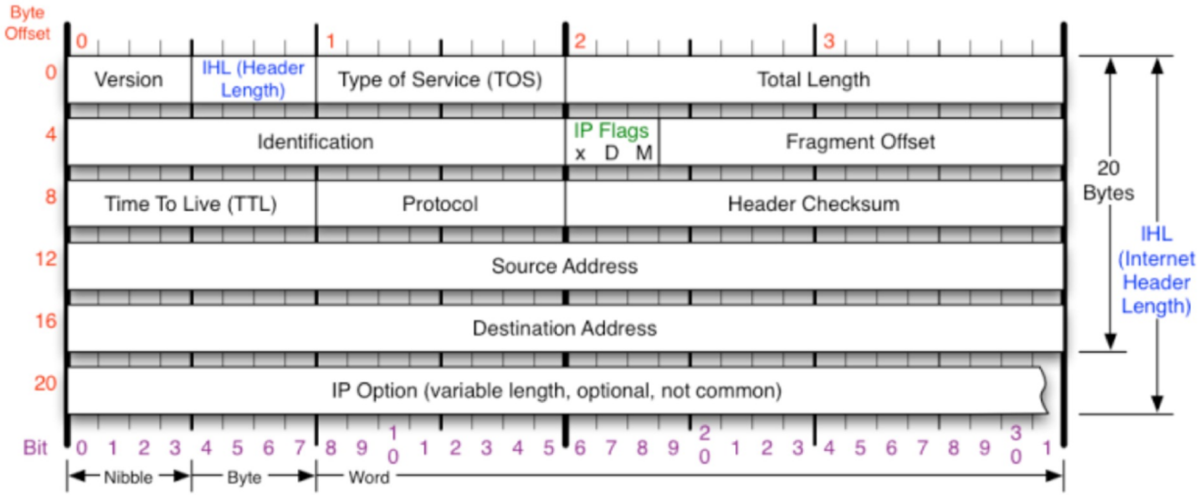
Internet
Get to final dest.

Link
Get to next-hop

Physical

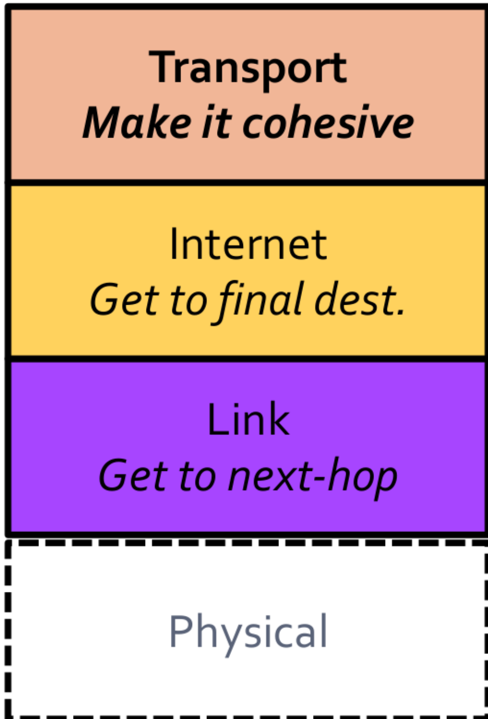
Internet Protocol v4

- Commonly used for client-side addressing
- 4-byte address (~4 billion total)
  - 192.168.1.30, 1.1.1.1, 130.160.0.54, …
- Networks often use “CIDR notation”
  - 1.1.1.0/24 → 1.1.1.0 – 1.1.1.255



Transport Layer

The **Transport layer** is responsible for ensuring that the data is processed in an orderly and complete manner.



Transport
Make it cohesive

Internet
Get to final dest.

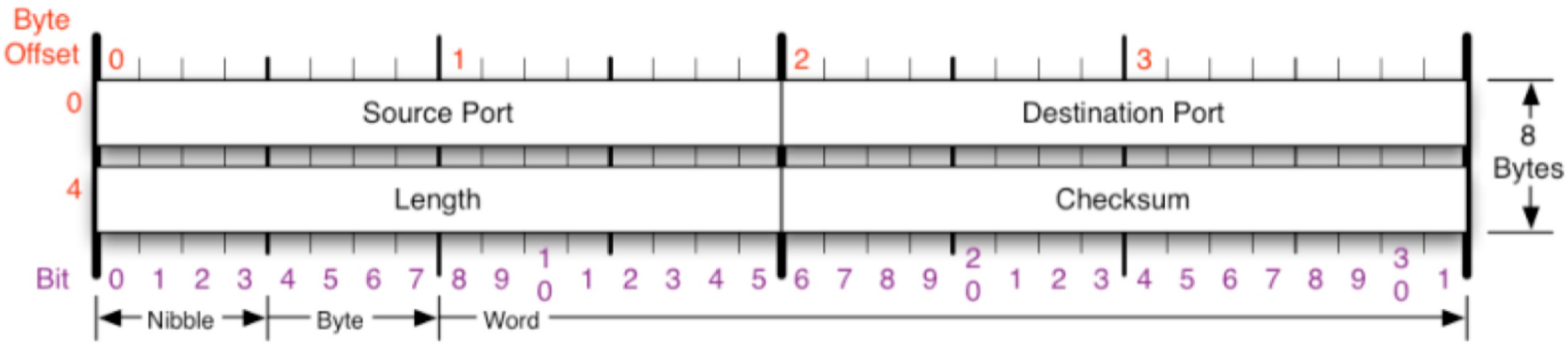
Link
Get to next-hop

Physical

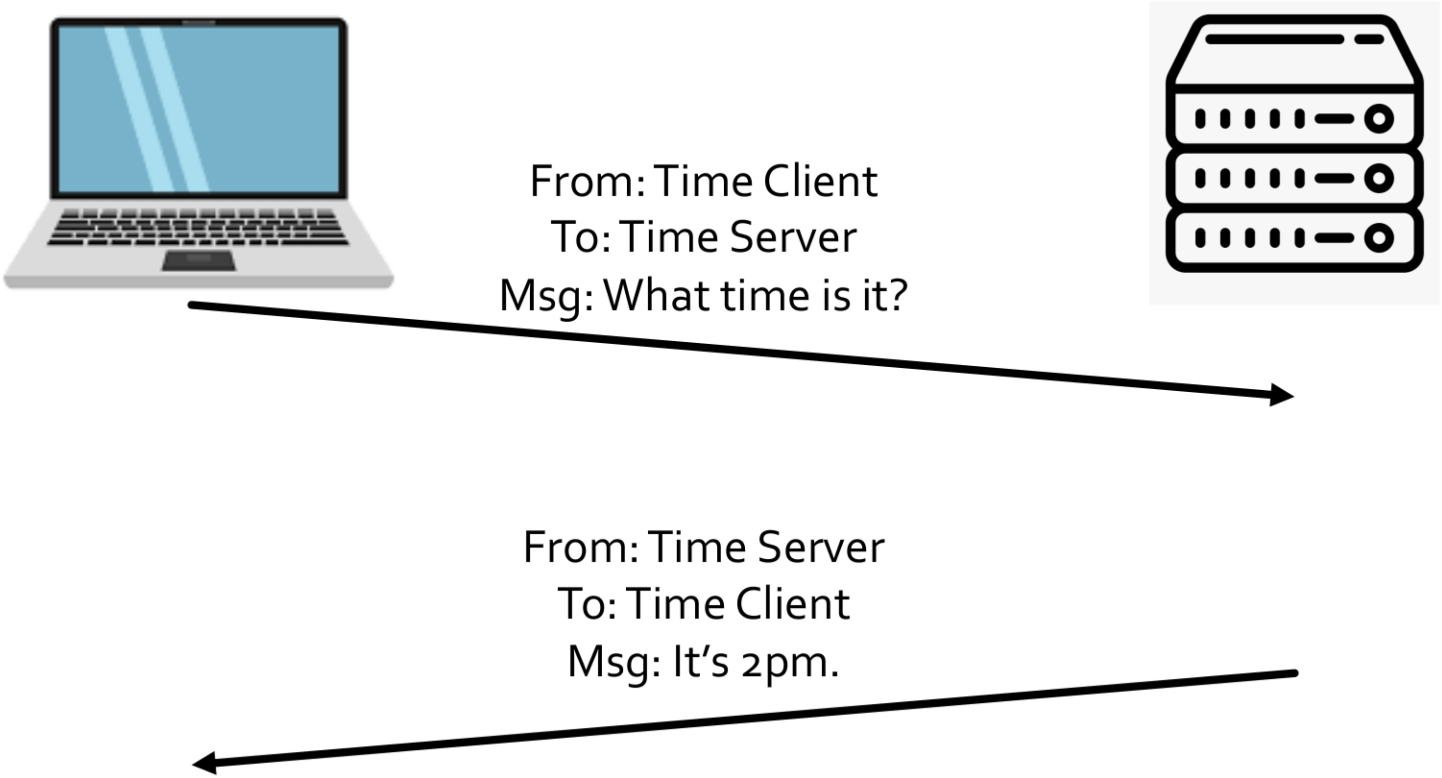
User Datagram Protocol (UDP)

- **Connectionless** protocol

- Used when:
  - Dropped packets are OK or recovery to be handled at the application layer



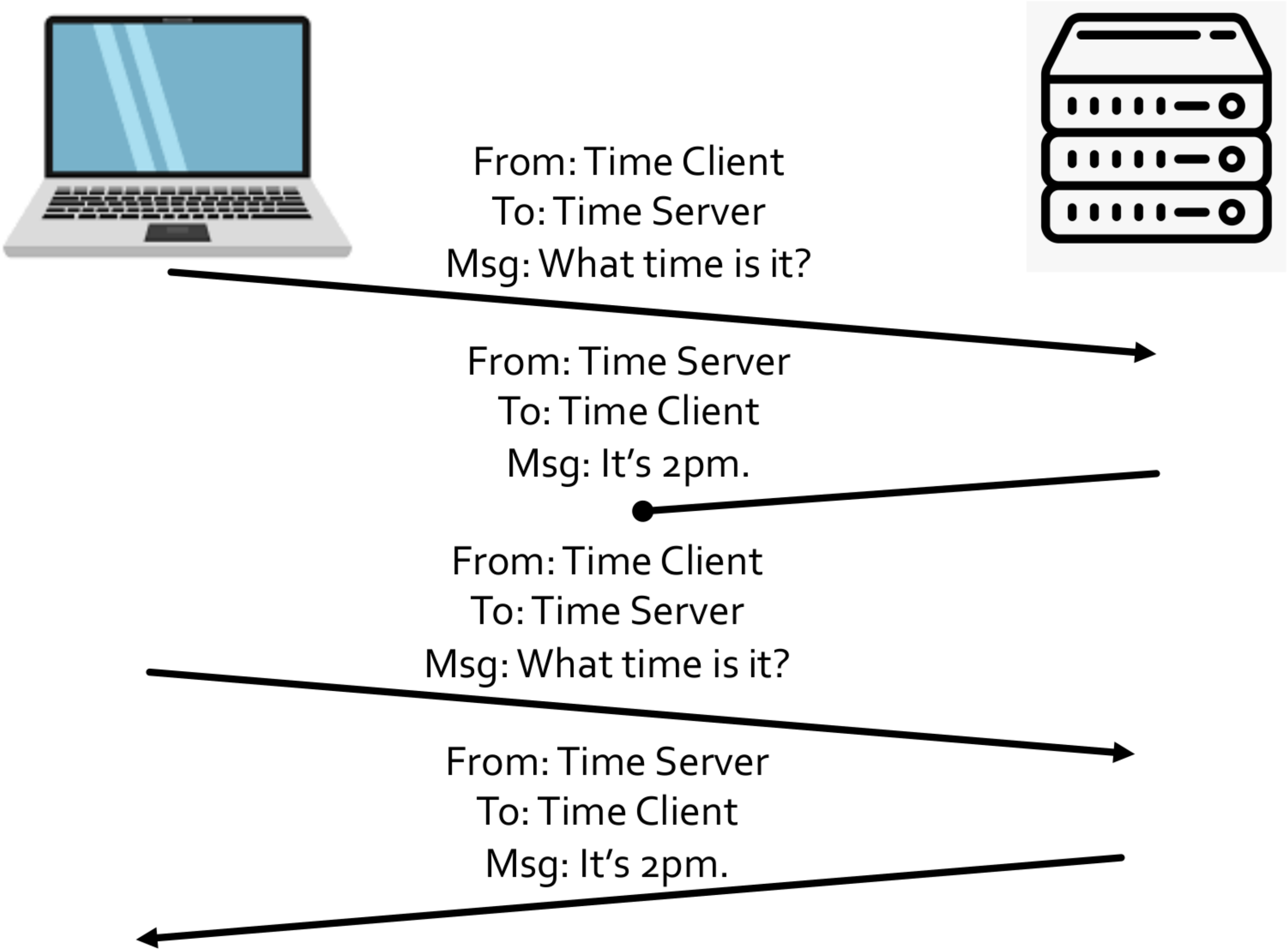
UDP Data Flow



From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It’s 2pm.

UDP Data Flow (packet loss)



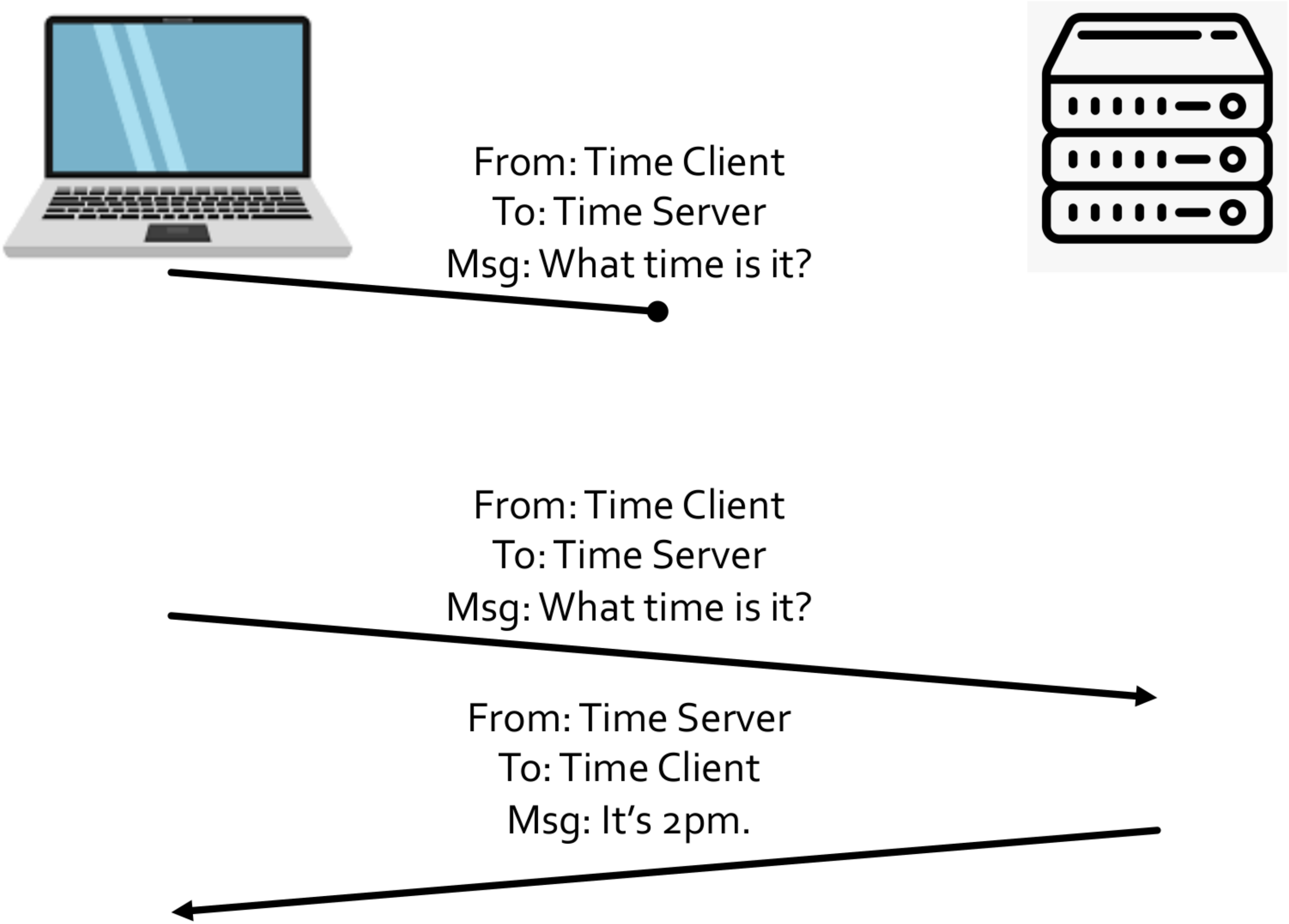
From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It’s 2pm.

From: Time Client
To: Time Server
Msg: What time is it?

From: Time Server
To: Time Client
Msg: It’s 2pm.

UDP Data Flow (packet loss)



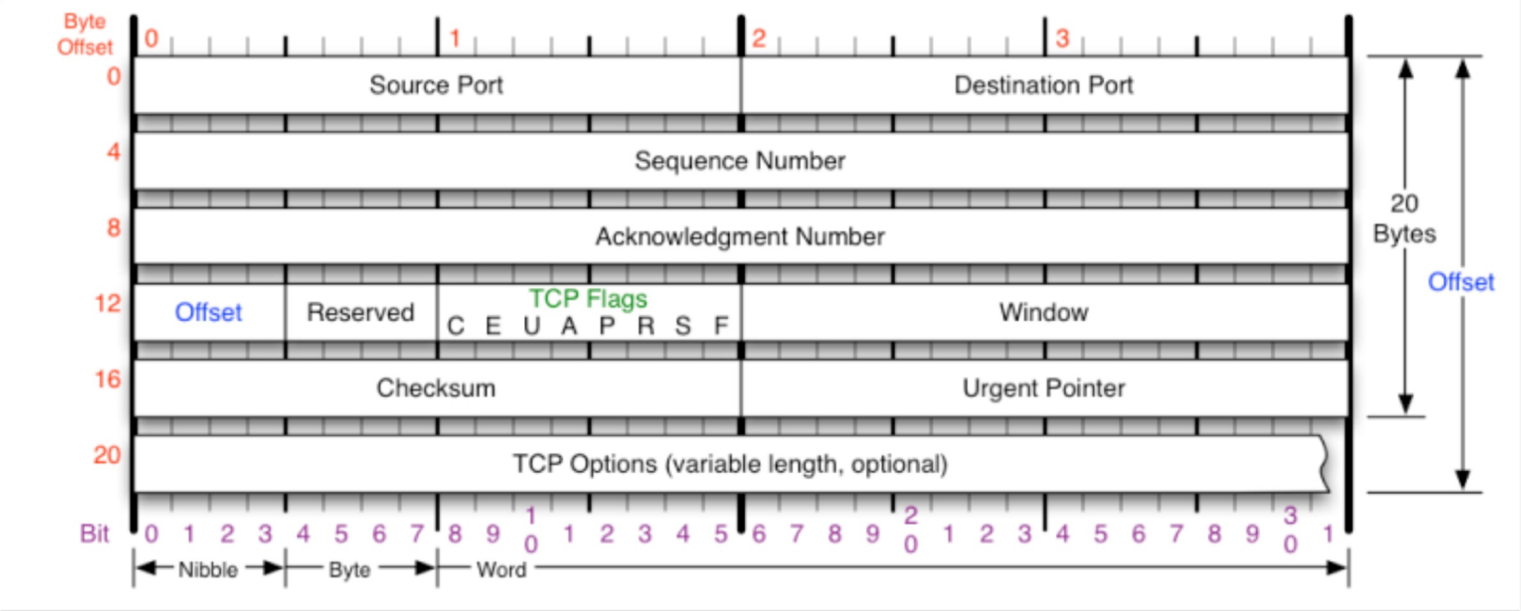
From: Time Client
To: Time Server
Msg: What time is it?

From: Time Client
To: Time Server
Msg: What time is it?

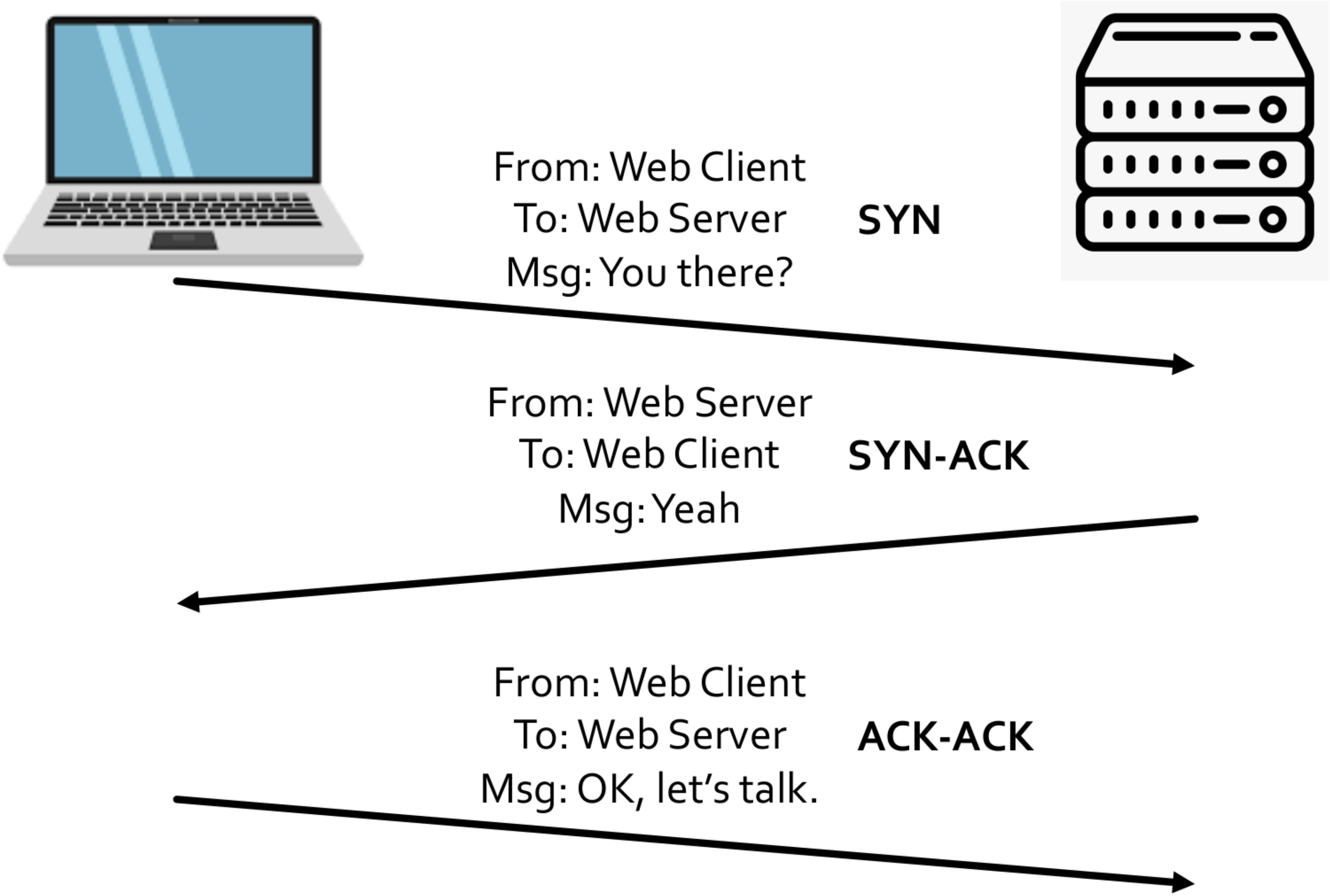
From: Time Server
To: Time Client
Msg: It's 2pm.

Transmission Control Protocol (TCP)

- **Connection-oriented** protocol
- Usually the default for communications
- Handles orderly bit stream details
  - Dropped packets, congestion control, etc



Three-Way TCP Handshake



From: Web Client
To: Web Server
Msg: You there?

SYN

From: Web Server
To: Web Client
Msg: Yeah

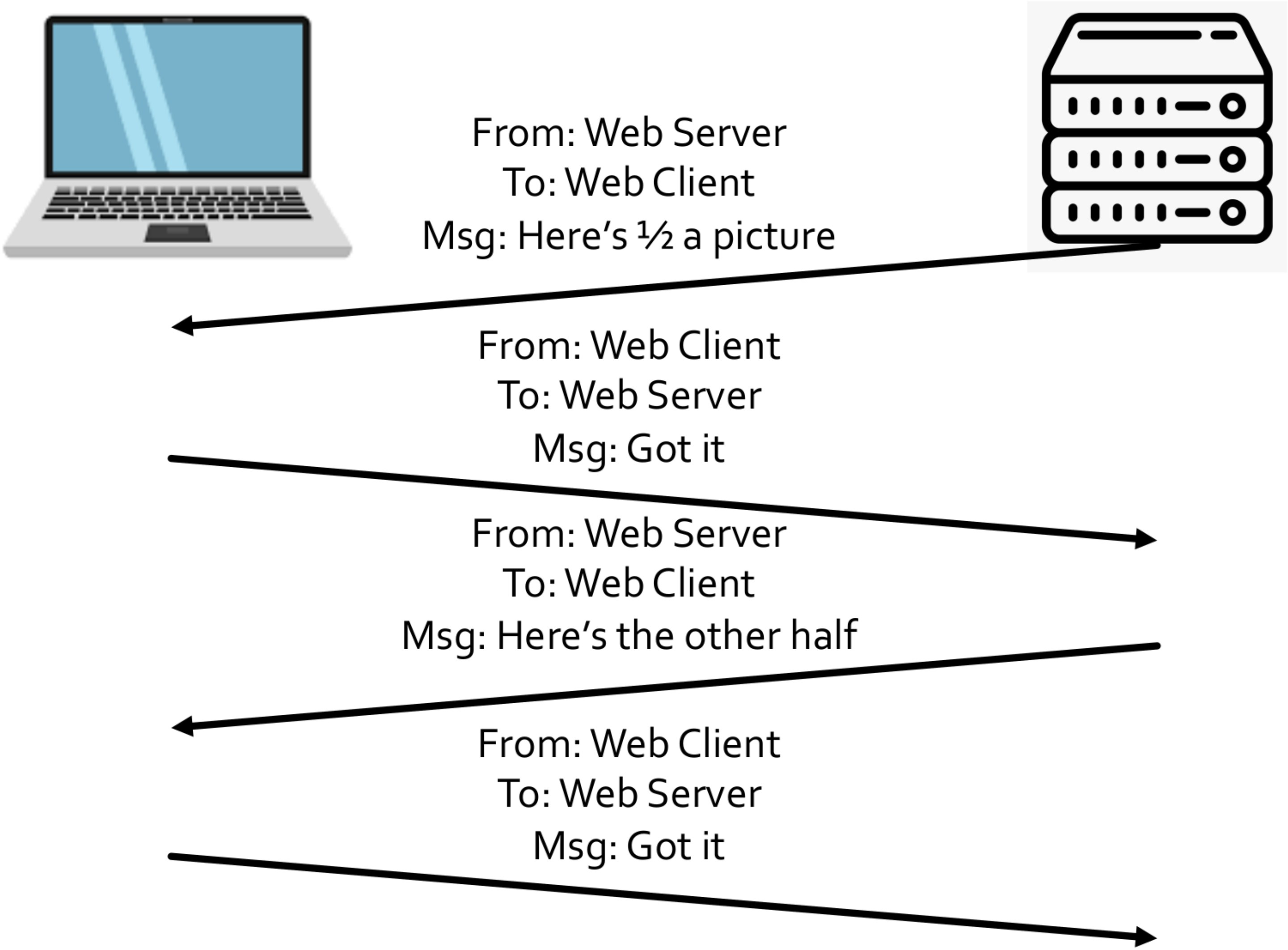
SYN-ACK

From: Web Client
To: Web Server
Msg: OK, let’s talk.

ACK-ACK

`--- BEGIN CONTENT ---`

TCP Acknowledgements



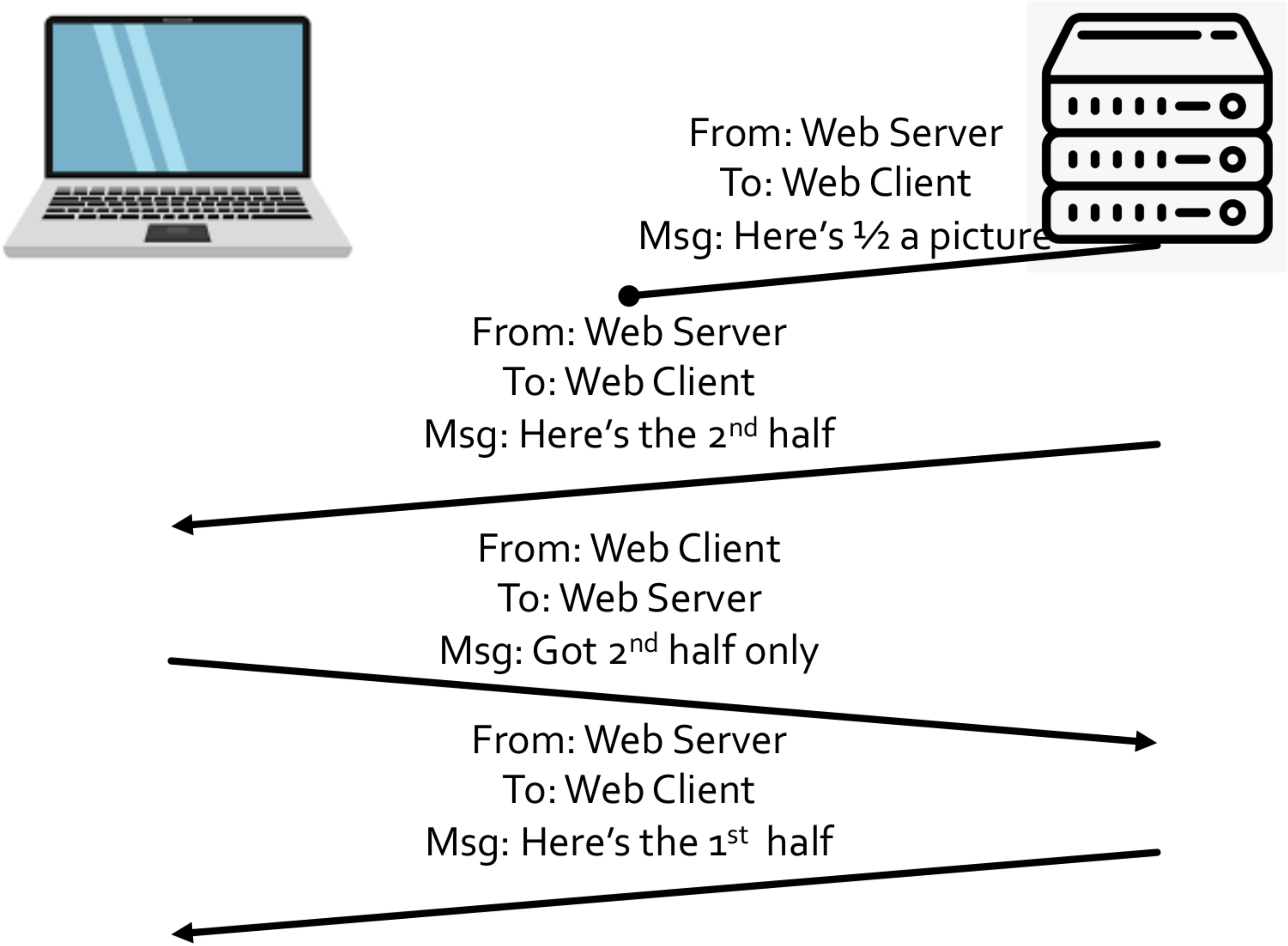
From: Web Server
To: Web Client
Msg: Here’s ½ a picture

From: Web Client
To: Web Server
Msg: Got it

From: Web Server
To: Web Client
Msg: Here’s the other half

From: Web Client
To: Web Server
Msg: Got it

TCP Handling Dropped Data



From: Web Server
To: Web Client
Msg: Here's ½ a picture

From: Web Server
To: Web Client
Msg: Here's the 2nd half

From: Web Client
To: Web Server
Msg: Got 2nd half only

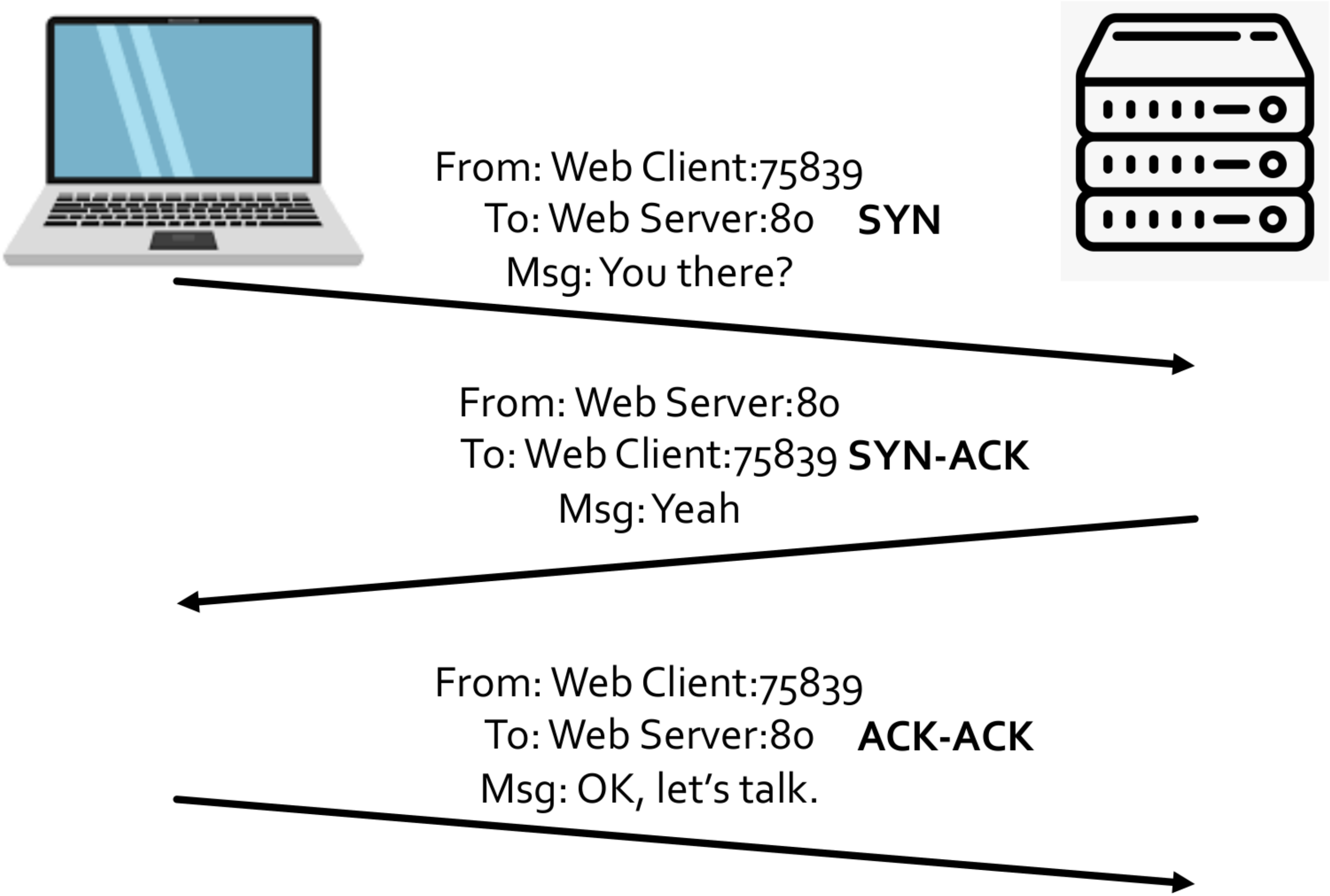
From: Web Server
To: Web Client
Msg: Here's the 1st half

Transport Layer Addressing

Application ports are used to address packets to *applications* running on device.

- Are a SW “port” not a HW “port”
- Often implicit but can be explicit
  - “Auburn website” == 131.204.138.170:80
  - Google’s “Honest” DNS == 8.8.8.8:53

Three-Way TCP Handshake



From: Web Client:75839
To: Web Server:80 **SYN**
Msg: You there?

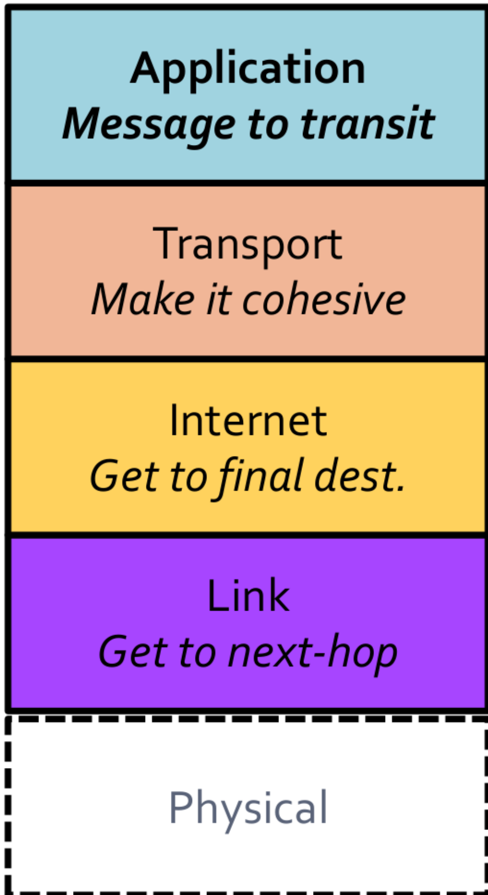
From: Web Server:80
To: Web Client:75839 **SYN-ACK**
Msg: Yeah

From: Web Client:75839
To: Web Server:80 **ACK-ACK**
Msg: OK, let’s talk.

Application Layer

The **application layer** is the highest-layer protocol and handles the logical interactions between endpoints.

- Most well-known protocols
  - DNS, HTTP, SMTP, etc



Application
Message to transit

Transport
Make it cohesive

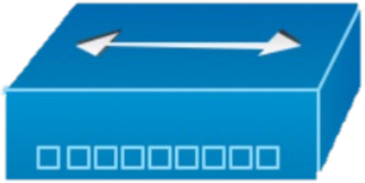
Internet
Get to final dest.

Link
Get to next-hop

Physical

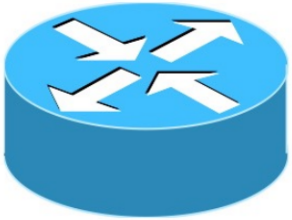
Network Devices

- **Hubs** are L1 devices
  - Packet comes in, packets go-out



- **Switches** are L2 devices
  - Dispatch packets via MAC address
  - “L3 switches” are common but are not what we’re talking about

- **Routers** are L3 devices
  - Dispatch packets via IP address
  - Lots of things called “routers” aren’t actually **routers** (but some are)



HTTP Protocol

The **Hypertext Transfer Protocol (HTTP)** is the base-protocol through which web servers and web clients communicate.

- *Idea* is extremely simple
- *Implementation* is extremely complicated

HTTP Protocol Details

- Methods (often referred to as “verbs”)
  - **GET**: Fetch content from a web server
  - **POST**: Send content to a web server
  - ***others exist*** for different uses

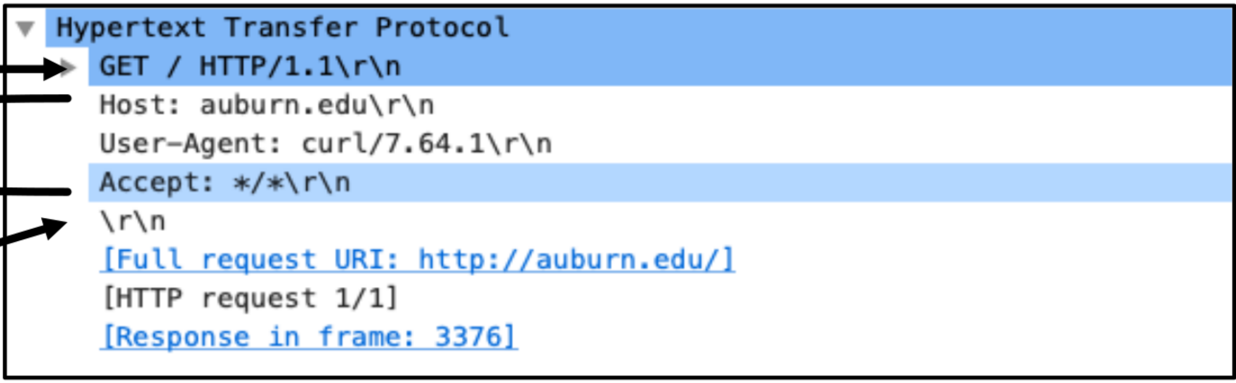
- Passes information via a “body” and arbitrary “headers” describing the body
  - CR/LF (“\r\n”) separated key-value pairs

HTTP Request

Method

Headers

<headers
finished>



```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: auburn.edu\r\n
    User-Agent: curl/7.64.1\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://auburn.edu/]
    [HTTP request 1/1]
    [Response in frame: 3376]
```

<no body in
GET request>

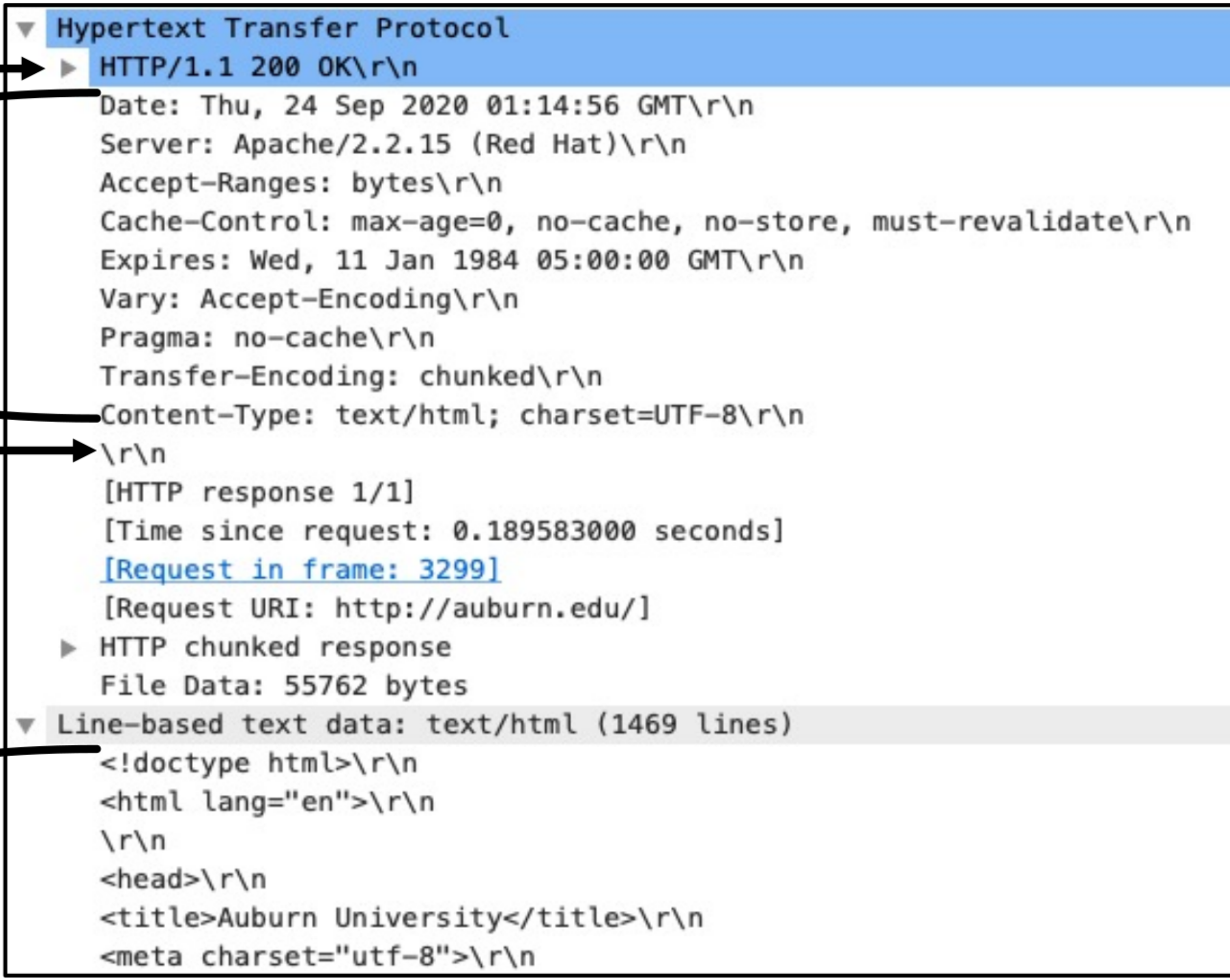
HTTP Response

Status

Headers

<headers finished>

Body



```
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\r\n
    Date: Thu, 24 Sep 2020 01:14:56 GMT\r\n
    Server: Apache/2.2.15 (Red Hat)\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: max-age=0, no-cache, no-store, must-revalidate\r\n
    Expires: Wed, 11 Jan 1984 05:00:00 GMT\r\n
    Vary: Accept-Encoding\r\n
    Pragma: no-cache\r\n
    Transfer-Encoding: chunked\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.189583000 seconds]
    [Request in frame: 3299]
    [Request URI: http://auburn.edu/]
  ▶ HTTP chunked response
    File Data: 55762 bytes
▼ Line-based text data: text/html (1469 lines)
    <!doctype html>\r\n
    <html lang="en">\r\n
    \r\n
    <head>\r\n
    <title>Auburn University</title>\r\n
    <meta charset="utf-8">\r\n
```

……… <truncated> …….

Common HTTP Responses

2XX – Normal

- 200 : OK
- 204 : OK(Unchanged)

3XX – Redirect

- 301 : Permanent Redirect
- 307 : Temporary Redirect

4XX – Client Error

- 400 : Bad Request
- 404 : No resource at requested path

5XX – Server Error

- 500 : *Server is on fire*
- 502 : *Corp network is on fire*

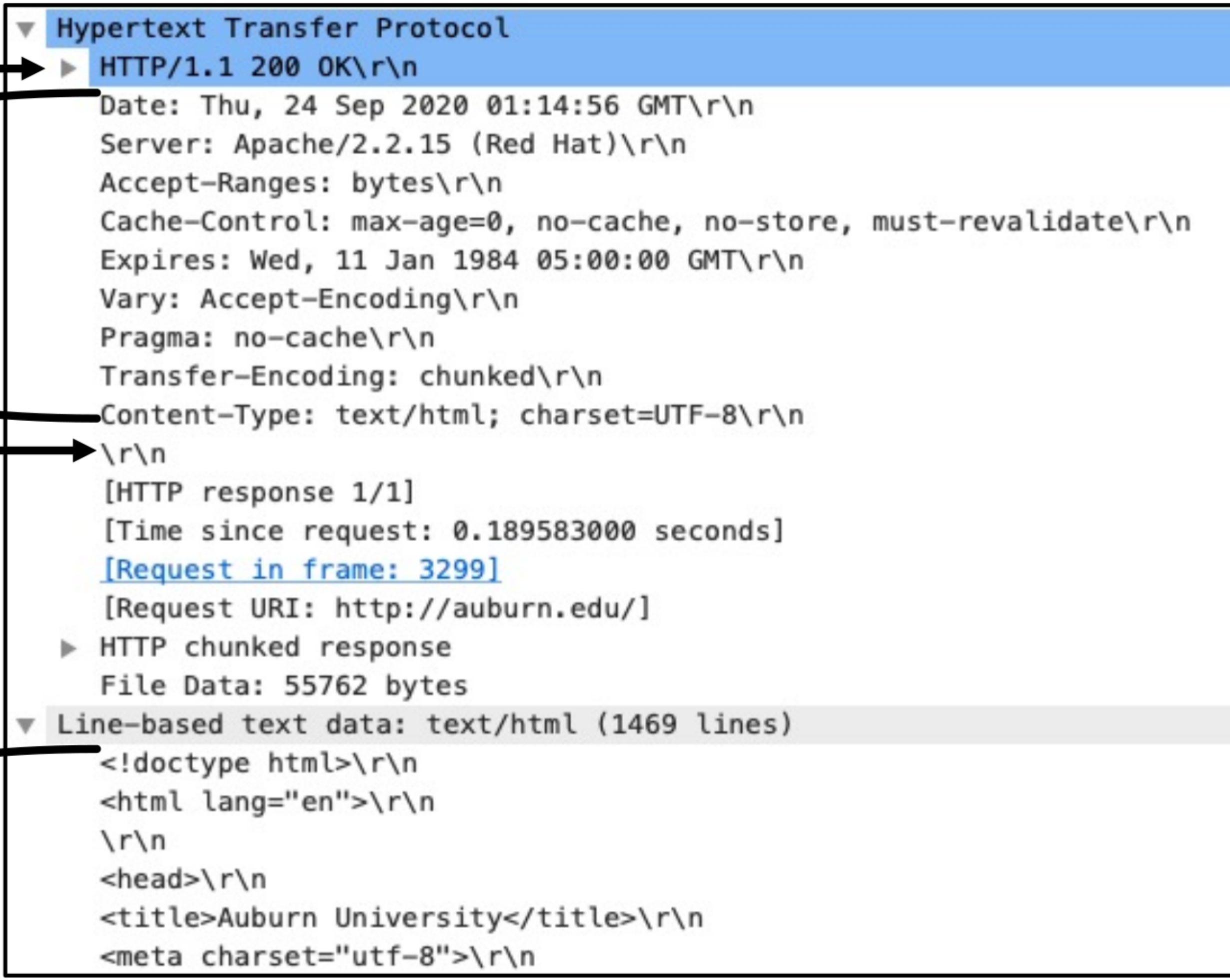
HTTP Response

Status

Headers

<headers finished>

Body



```
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\r\n
    Date: Thu, 24 Sep 2020 01:14:56 GMT\r\n
    Server: Apache/2.2.15 (Red Hat)\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: max-age=0, no-cache, no-store, must-revalidate\r\n
    Expires: Wed, 11 Jan 1984 05:00:00 GMT\r\n
    Vary: Accept-Encoding\r\n
    Pragma: no-cache\r\n
    Transfer-Encoding: chunked\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.189583000 seconds]
    [Request in frame: 3299]
    [Request URI: http://auburn.edu/]
  ▶ HTTP chunked response
    File Data: 55762 bytes
▼ Line-based text data: text/html (1469 lines)
    <!doctype html>\r\n
    <html lang="en">\r\n
    \r\n
    <head>\r\n
    <title>Auburn University</title>\r\n
    <meta charset="utf-8">\r\n
```

.... <truncated> ....

Canonical Protocol Stacks (unencrypted)

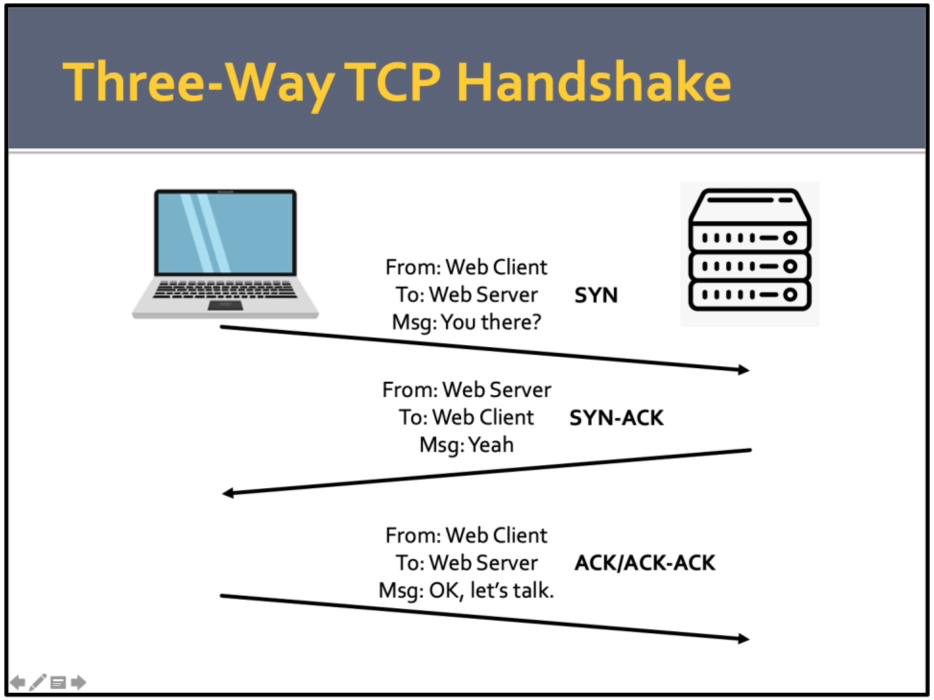
Application Layer Protocol	Transport Layer Protocol	Port	Name
FTP	TCP	20	File Transfer Protocol – Data
FTP	TCP	21	FTP – Connection
Telnet	TCP	23	Telnet
SMTP	TCP	25	Simple Mail Transfer Protocol
DNS	TCP / UDP	53	Domain Name System – Zone Transfer / Lookups
DHCP	UDP	67 / 68	Dynamic Host Configuration Protocol – Server / Client
HTTP	TCP	80	Hypertext Transfer Protocol
POP3	TCP	110	Post Office Protocol
SNMP	UDP	161	Simple Network Management Protocol (v1,2)
RDP	TCP / UDP	3389	Remote Desktop Protocol

Canonical Protocol Stacks (encrypted)

Application Layer Protocol	Transport Layer Protocol	Port	Name
SFTP	TCP	22	SSH FTP
SSH	TCP	22	Secure Shell
SCP	TCP	22	Secure Copy
SNMP	UDP	161	Simple Network Management Protocol (v3)
HTTPS	TCP	443	HTTP over SSL / TLS
SMTPS	TCP	465	Simple Mail Transfer Protocol over SSL / TLS
FTPS	TCP	990	FTP over SSL / TLS
POP3S	TCP	995	Post Office Protocol over SSL / TLS

Network Scanning

Network scanning is a reconnaissance technique that is used by attackers to gain information to aid them in their attacks.



Three-Way TCP Handshake

From: Web Client
To: Web Server
Msg: You there?

SYN

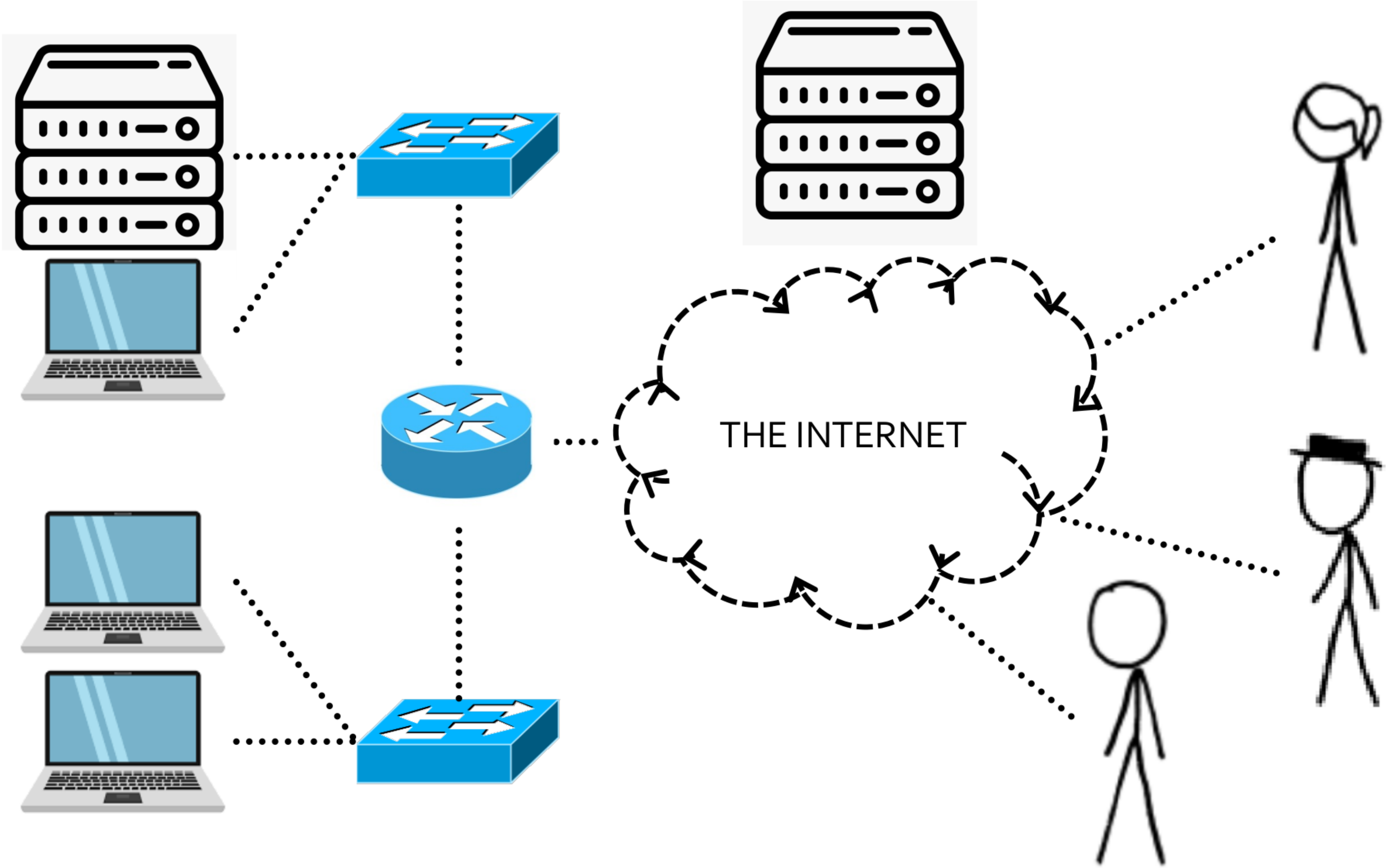
From: Web Server
To: Web Client
Msg: Yeah

SYN-ACK

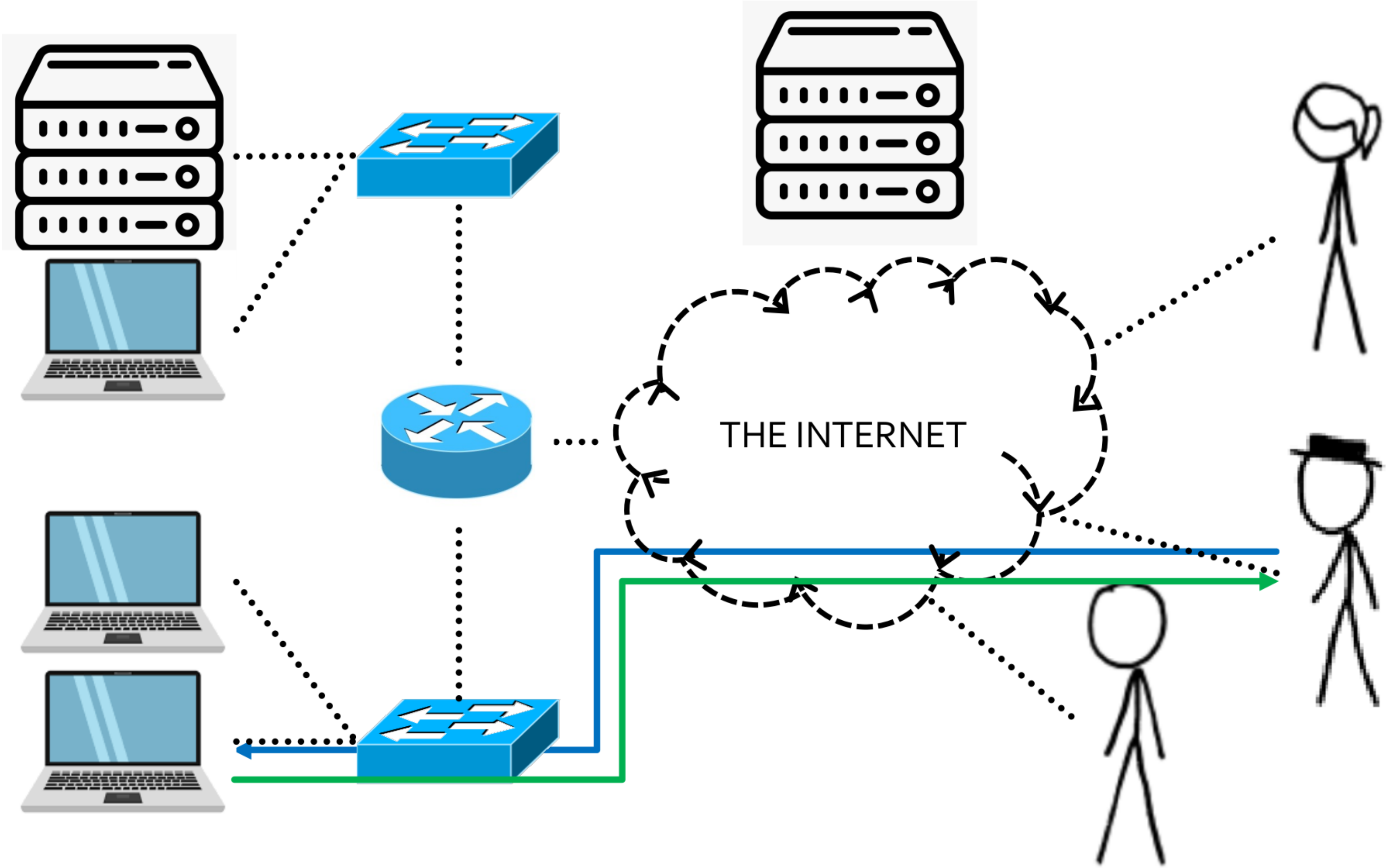
From: Web Client
To: Web Server
Msg: OK, let's talk.

ACK/ACK-ACK

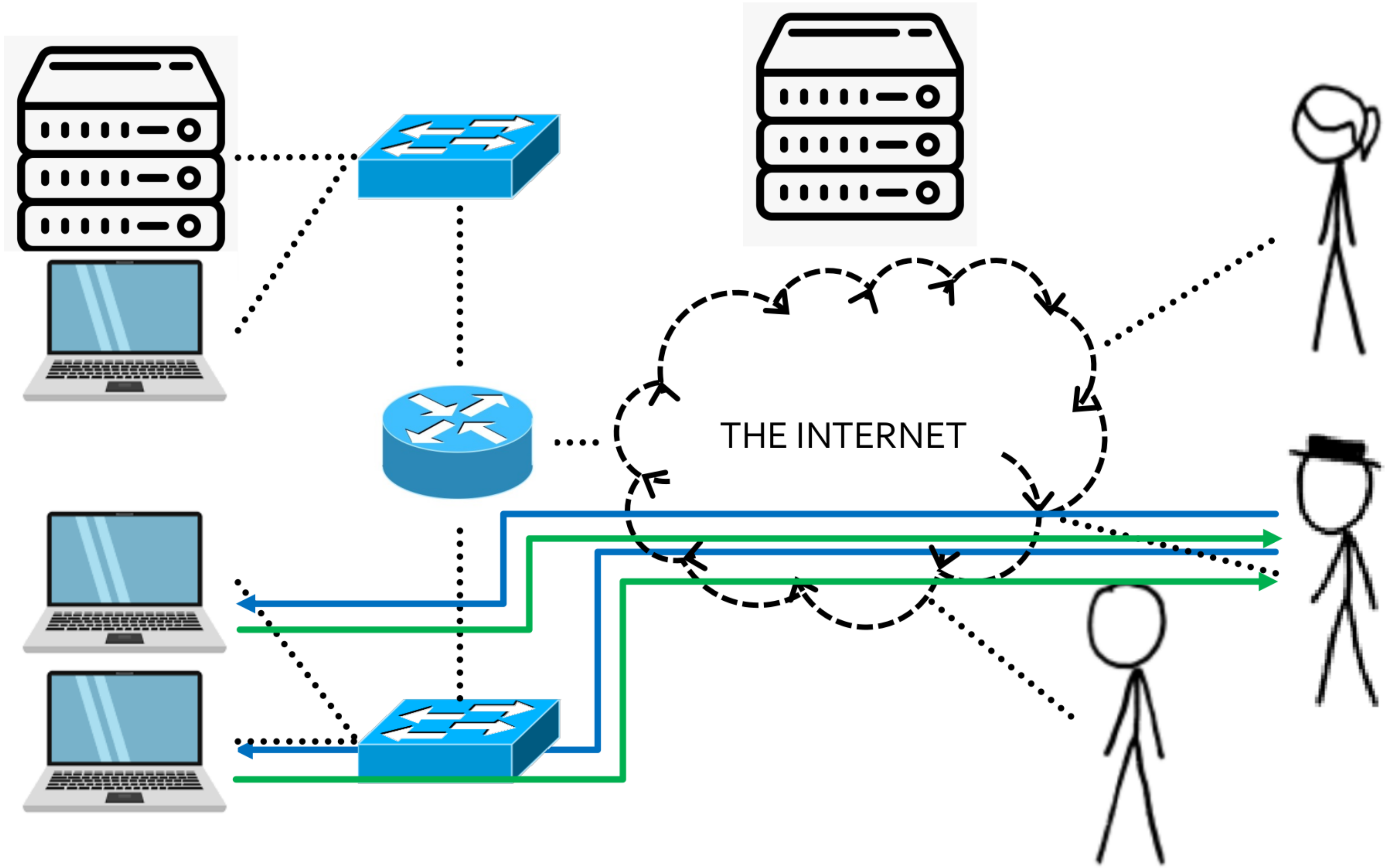
Horizontal TCP Port Scanning



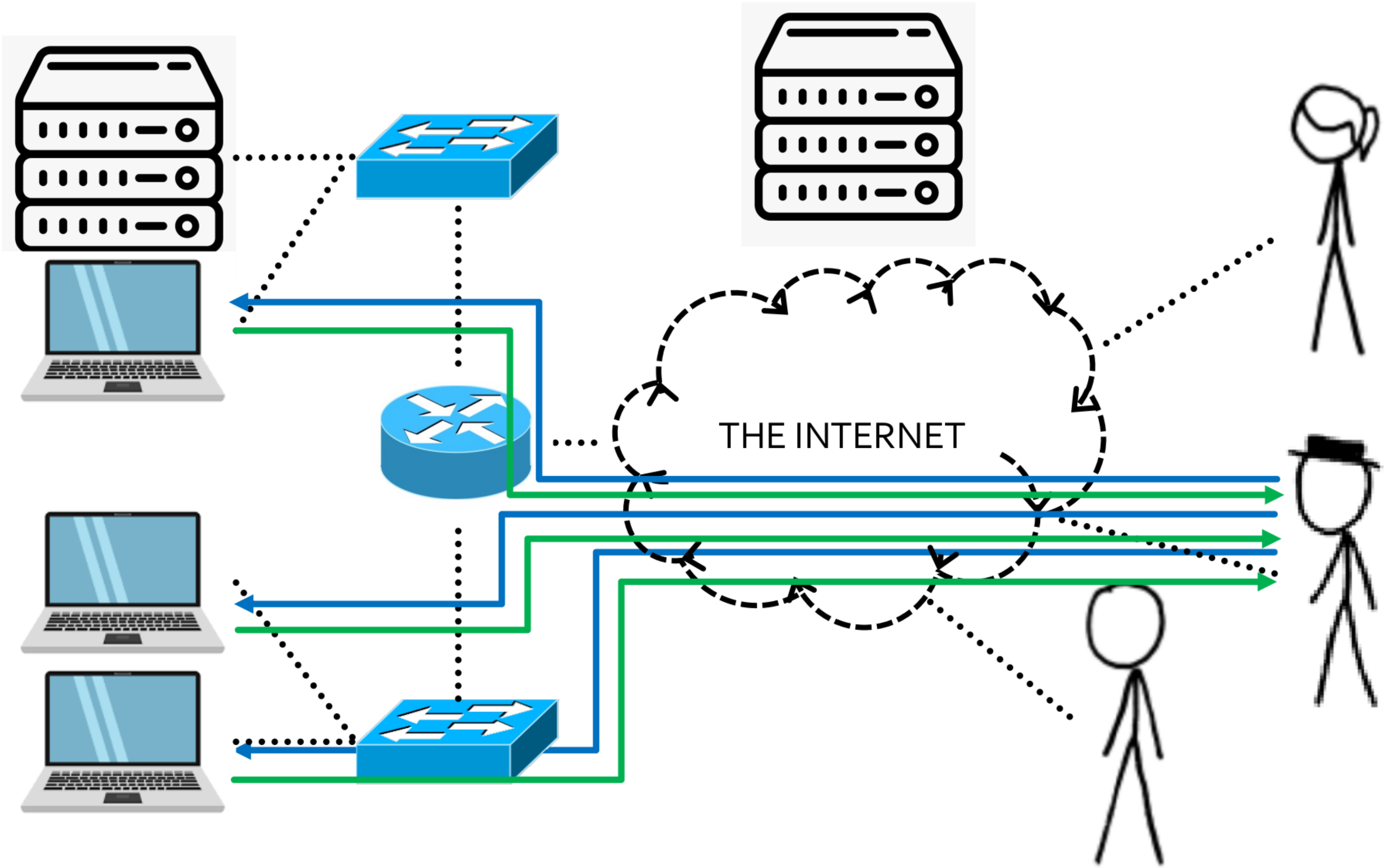
Horizontal TCP Port Scanning



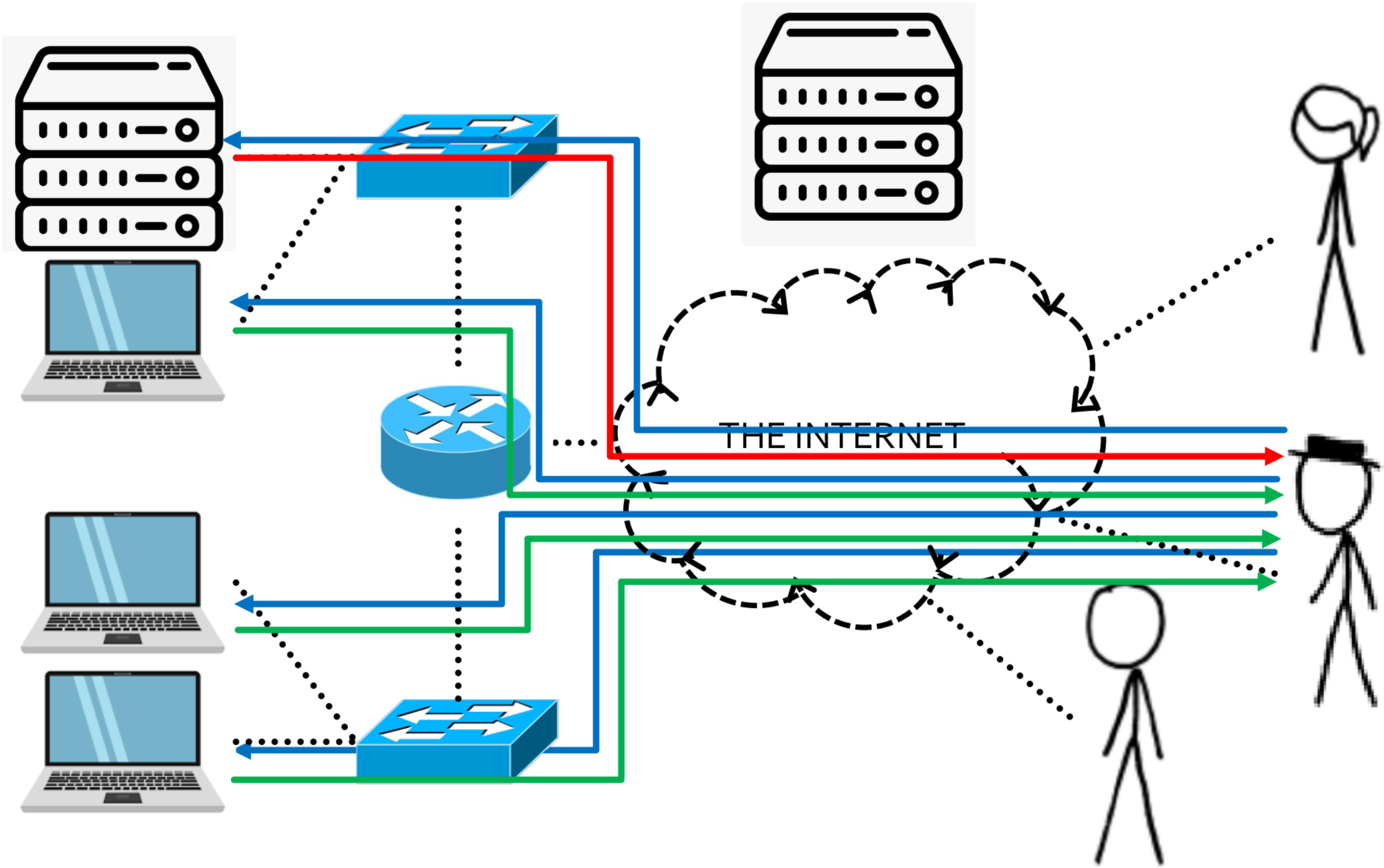
Horizontal TCP Port Scanning



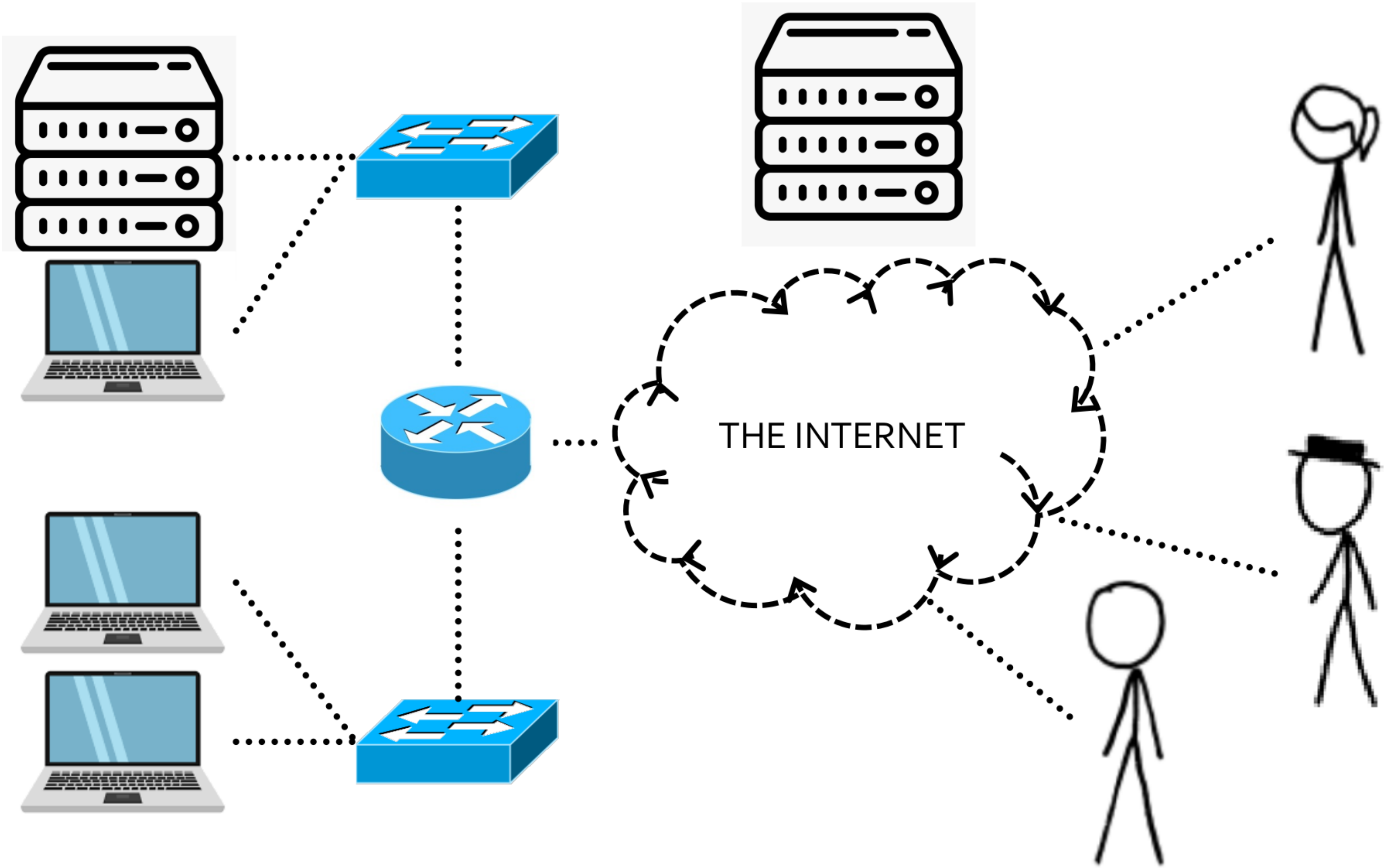
Horizontal TCP Port Scanning



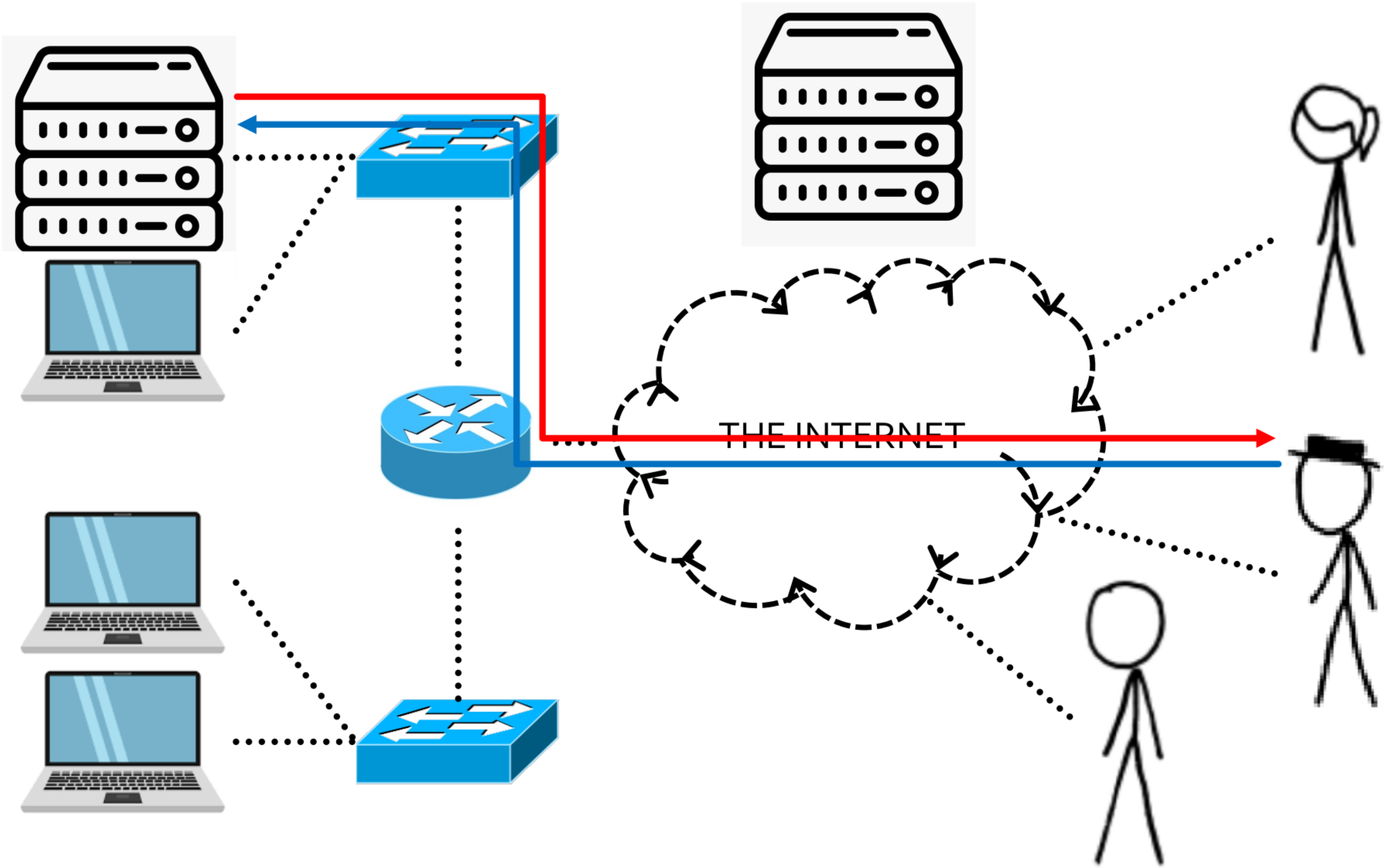
Horizontal TCP Port Scanning



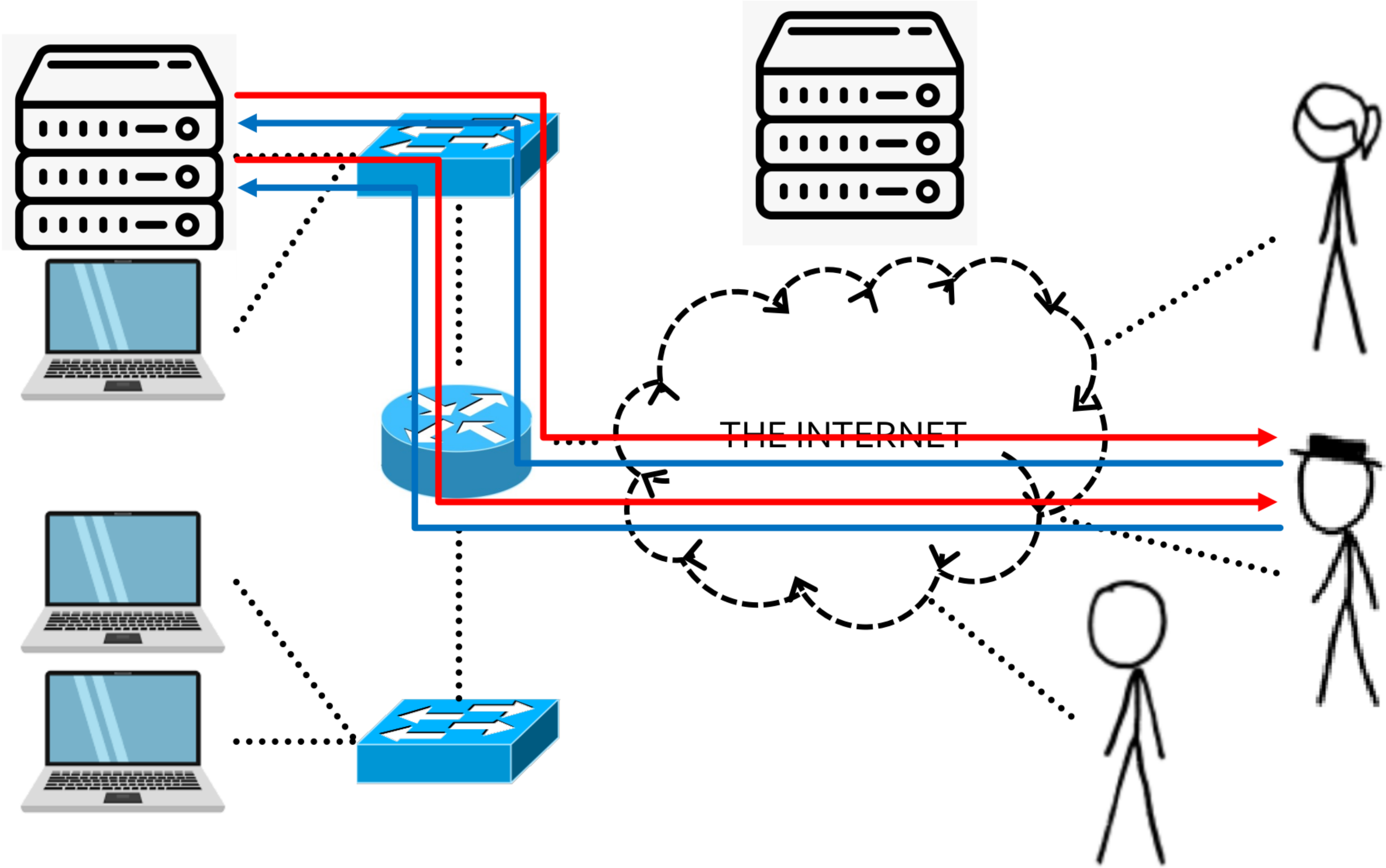
Vertical TCP Port Scan



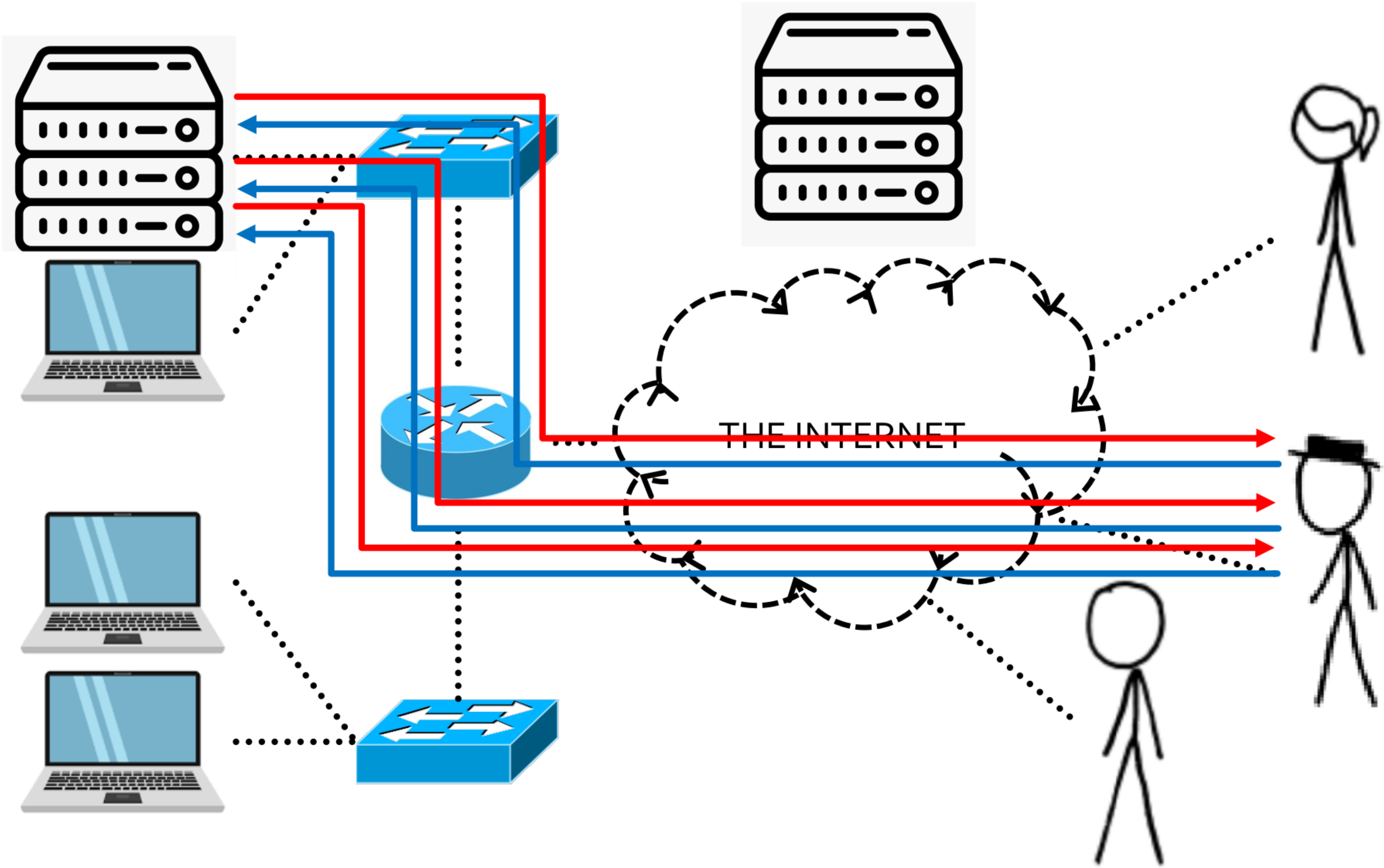
Vertical TCP Port Scan



Vertical TCP Port Scan



Vertical TCP Port Scan


THE INTERNET