Network Security III

CSE 565: Fall 2024
Computer Security

Xiangyu Guo (xiangyug@buffalo.edu)

University at Buffalo

Disclaimer

- We don’t claim any originality of the slides. The content is developed heavily based on
  - Slides from Prof. Dan Boneh and Prof. Zakir Durumeric’s lecture on Computer Security (https://cs155.stanford.edu/syllabus.html)
  - Slides from Prof Nick McKeown’s lecture on Computer Network (https://vixbob.github.io/cs144-web-page/)
  - Slides from Prof Ziming Zhao’s past offering of CSE565 (https://zzm7000.github.io/teaching/2023springcse410565/index.html)
  - Slides from Prof Hongxin Hu’s past offering of CSE565

Announcement

- Midterm Grades will be released tonight (10/29)
- HW3 and Project3 will be released tonight (10/29), **due Tue, Nov 12, 23:59 pm**.

Review of Last Lecture

- IP: how packets travel across network boundaries and reach its destination
  - Subnet & LAN.
- TCP: From packets to bytestream
  - Protocol def
  - Attacks: Spoofing; SYN-Flood; Reset; Session Hijacking.
- DNS: Mapping host name to IP address
  - Protocol def
  - Attacks: Cache poisoning; Kaminsky; Rebinding.

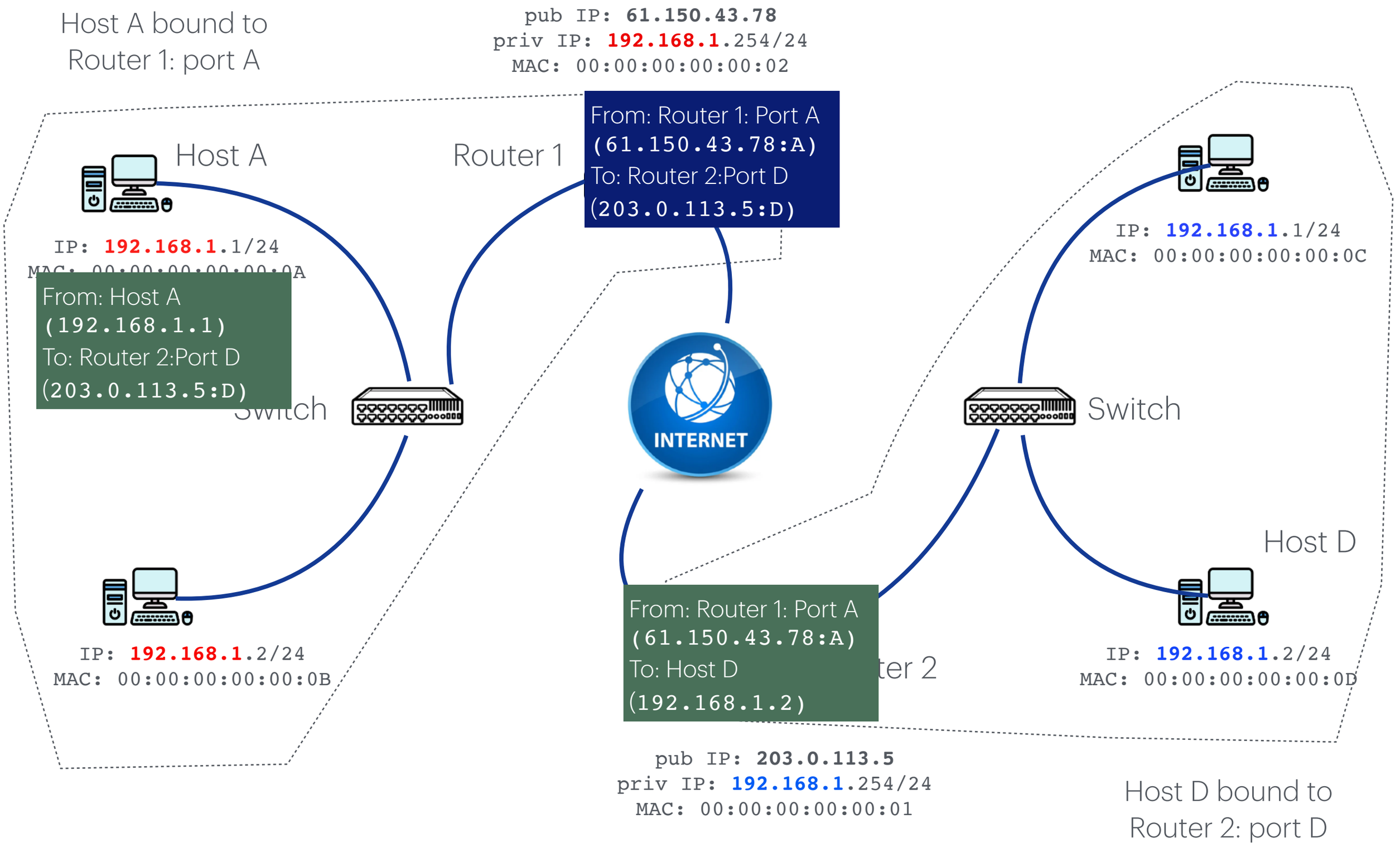
Communicating at the Link & Network Layer

- Summary
  - **Within a same LAN**
    - Only need Link Layer support: MAC address (from e.g. ARP)
    - Peer-to-peer or centralized (forwarded by a switch)
  - **Between different LANs**
    - Need Network Layer support: IP address (from e.g. DNS)
    - Go through routers: Network Address Translation

Communicate between LANs

- Devices in a same LAN shares a *same* public IP via the router
  - Usually they are bound to different ports of the router
  - E.g., a web server may bind port 80 of the router
- Anyone outside the LAN can *only* send msg to the router's public IP addr
- The router will forward the msg based on the receiving port
  - Usually involves *translating* the destination from **`public_ip:port_A`** to some **`private_ip:port B`**
  - Known as Network Address Translation (NAT)

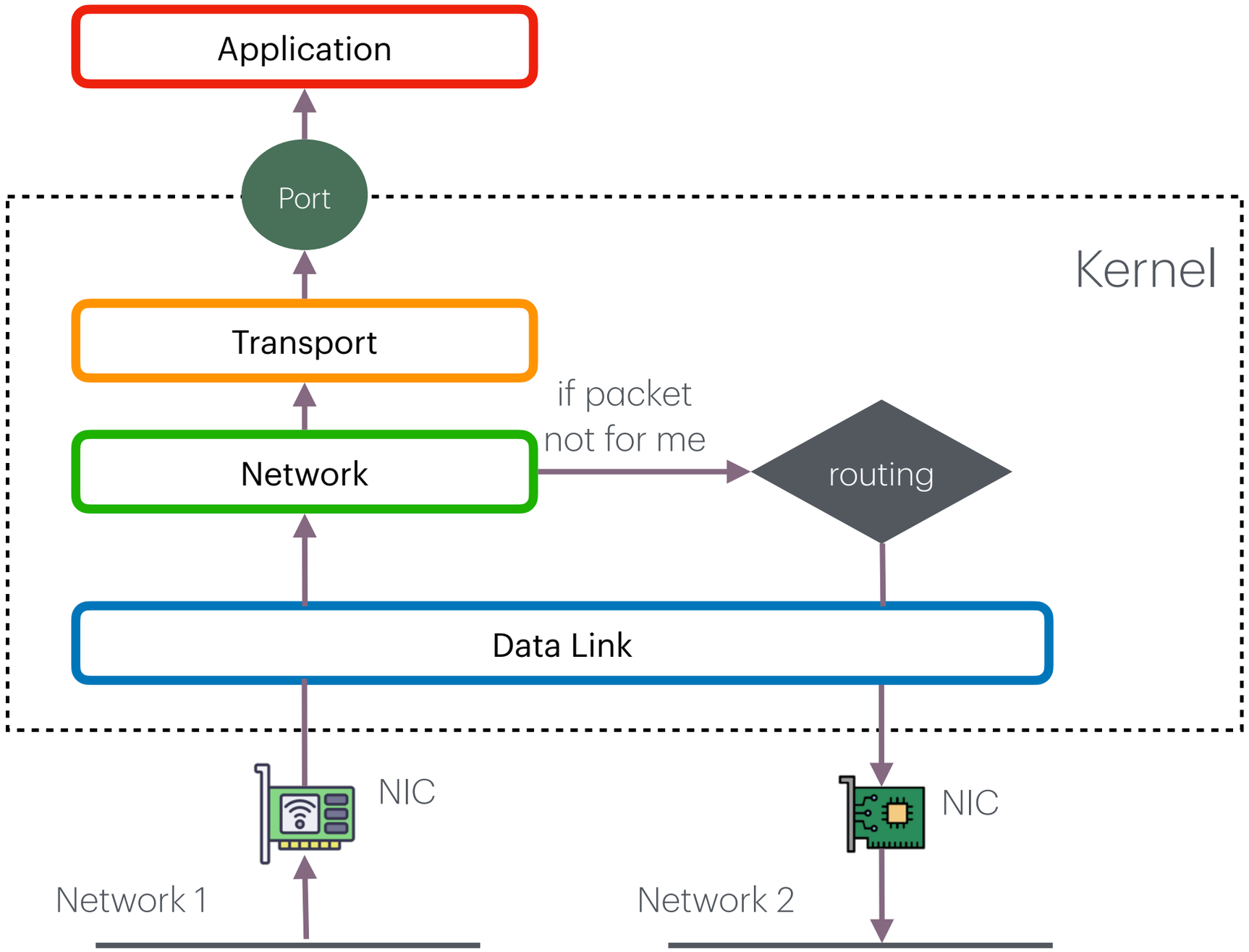
Communicate between LANs



Ports

- Each **application/process** on a host is identified by a port number
  - Ports are numbered from 1 – 65535 (16 bits)
- TCP connection established between port `A` on host `X` to port `B` on host `Y`
- Extend network layer (IP)'s service from host-to-host delivery to process-to-process delivery.

Packet receiving

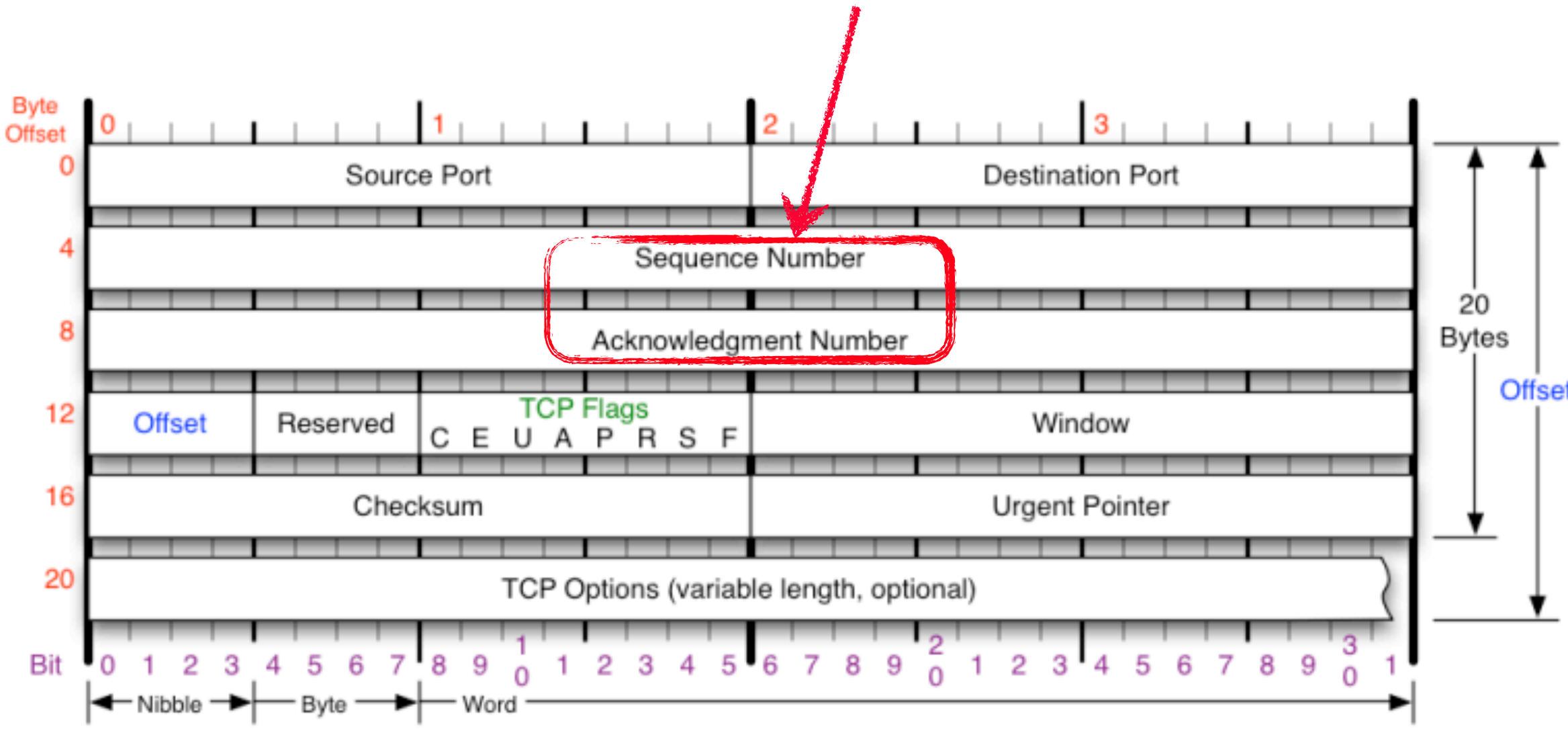


From Packets to Bytestreams

- Many applications want a stream of bytes delivered reliably and in-order between applications on different hosts
- **Transmission Control Protocol (TCP)** provides …
  - Connection-oriented protocol with explicit setup/teardown
  - Reliable in-order byte stream
  - Congestion control
- Despite IP packets being dropped, re-ordered, and duplicated

TCP Packet

Key fields that enable reliable transmission



TCP Seq. No. and Ack. No.

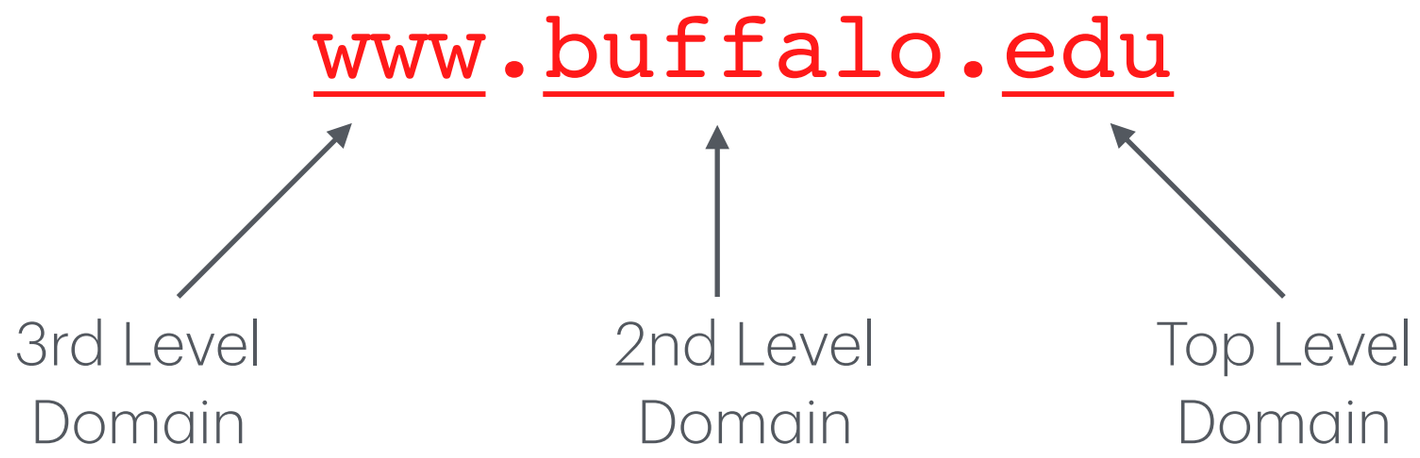
- Two data streams in a TCP session, one in each direction
- **Sequence Number**: Bytes in each data stream are numbered with a 32-bit number.
  - The numbering starts with an random offset.
- **Acknowledge Number**: Receiver sends acknowledgement number that indicates data received
  - The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.

TCP Attacks

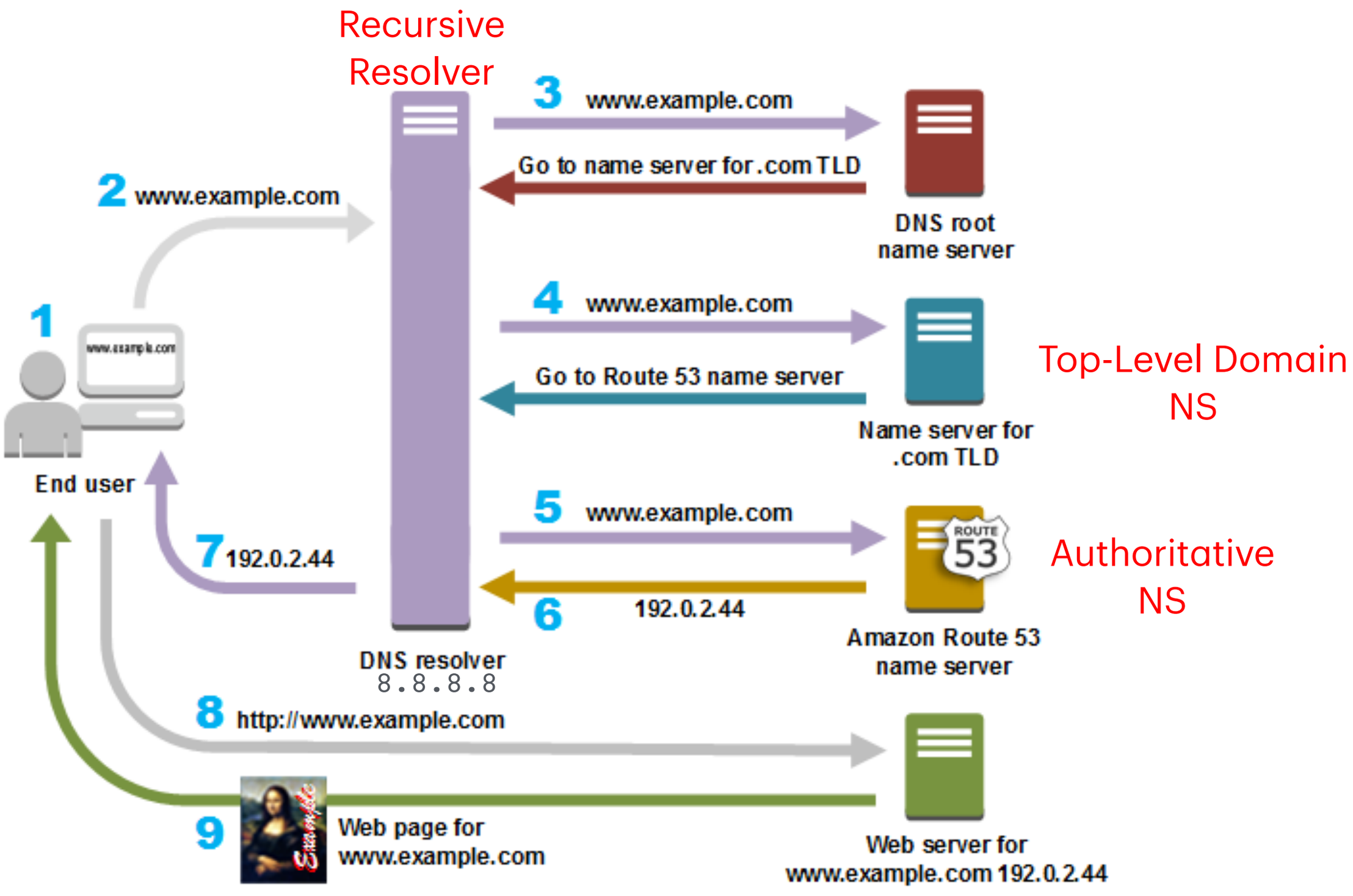
- Packet ordering are based on header information sent in **cleartext**, no authentication / integrity
  - Leads to spoofing attacks: Session Hijacking
    - Needs to get IP, Port, & Seq Number correct
- TCP is **stateful**:
  - To implement reliable & ordered transmission & traffic control, the two sides need to maintain state info $\Longrightarrow$ consume resources
  - Leads to Denial of Service attack: SYN flooding; TCP Reset Attack

DNS (Domain Name Service)

- Map host name to IP; Similar role as ARP in Link Layer (map IP to MAC).
- Implemented as a distributed, delegatable, and hierarchical name space (database)



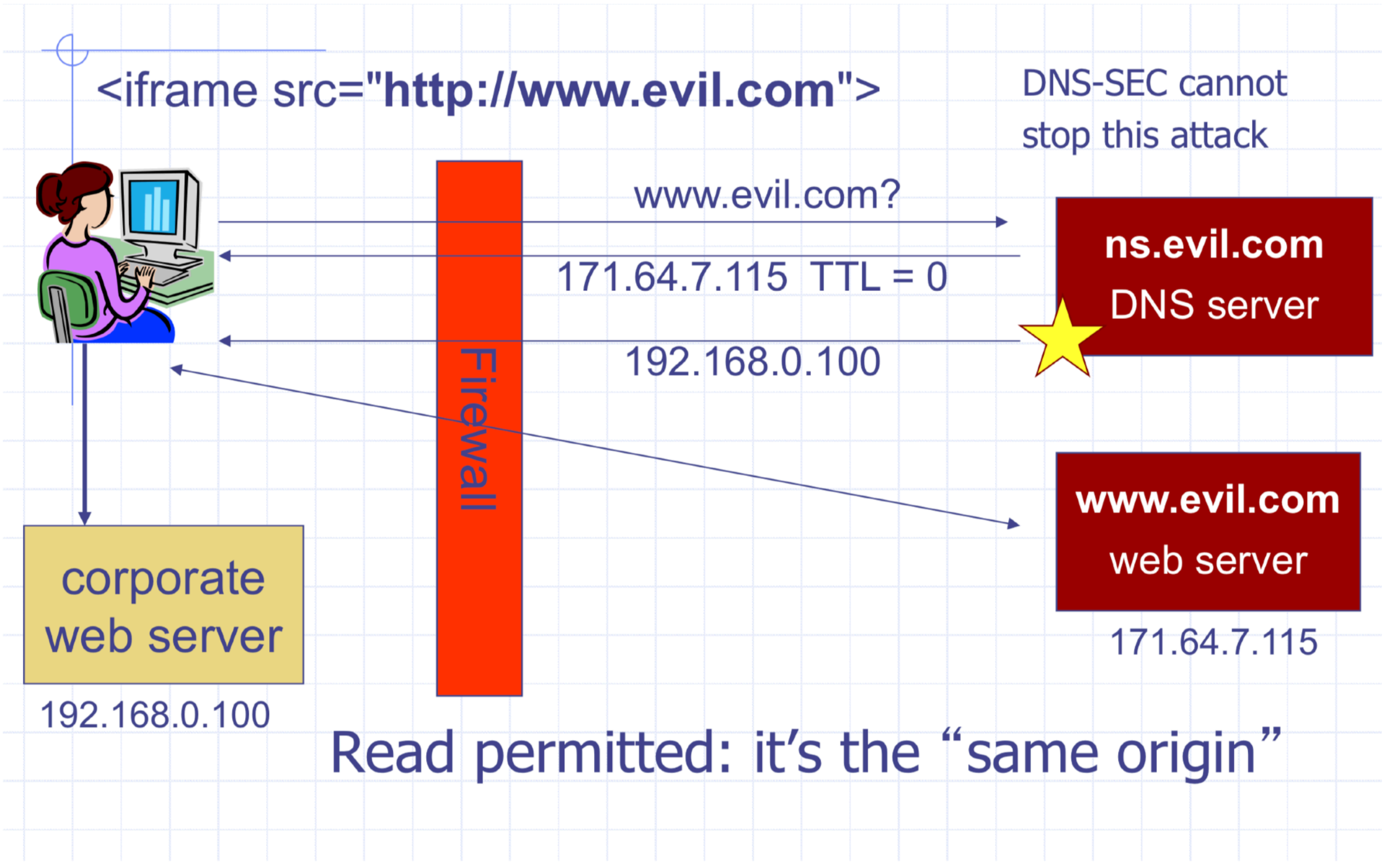
(Recursive) DNS resolution



DNS Security

- Again, like ARP, no built-in authentication / integrity.
  - Leads to DNS Poisoning attacks: Kaminsky Attack (Birthday attack)
- Working at the boundary of Application layer & Network layer
  - Able to circumvent application-layer security measures: DNS Rebinding attack
- Natural message amplifier: DNS response might be much larger than DNS request
  - Leads to Denial-of-Service attack (Today's topic)

DNS Rebinding Attack



Today's Topic

- Denial-of-Service Attacks

Packet Spoofing

Denial of Service (DoS) Attacks

Denial of Service Attacks

- **Goal:** take large service/network/org offline by overwhelming it with network traffic such that they can't process real requests
  - Exhausting CPU cycles: processing packets requires CPU resource.
  - Exhausting Network bandwidth: sending/receiving packets requires network badwidty.

- **How**: find mechanism where attacker doesn't spend a lot of effort, but requests are difficult/expensive for victim to process

Types of Attacks

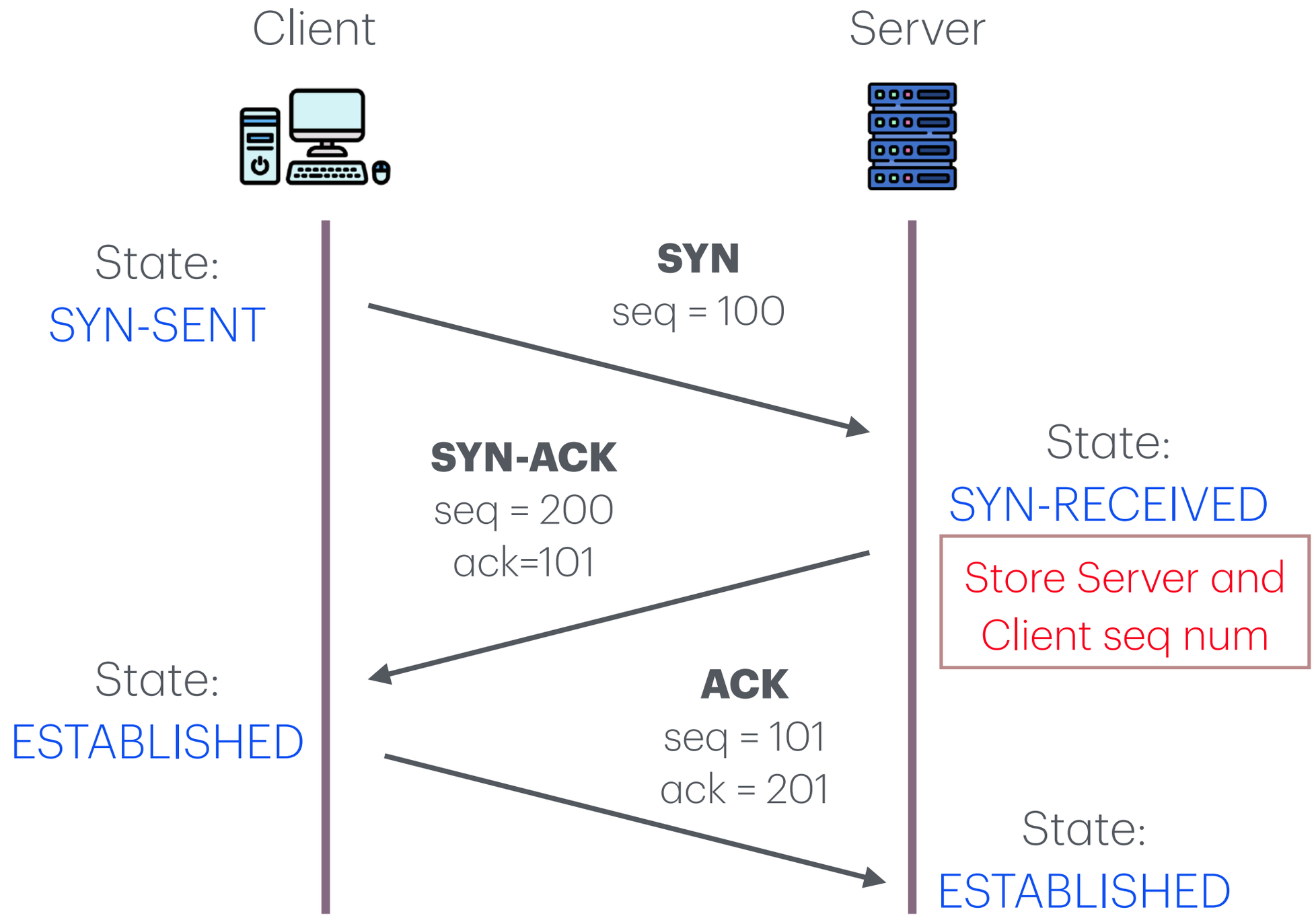
- **DoS Bug:** design flaw that allows one machine to disrupt a service. Generally a protocol asymmetry, e.g., easy to send request, difficult to create response. Or requires server state.

- **DoS Flood**: control a large number of requests from a botnet or other machines you control

DoS Opportunities at Every Layer

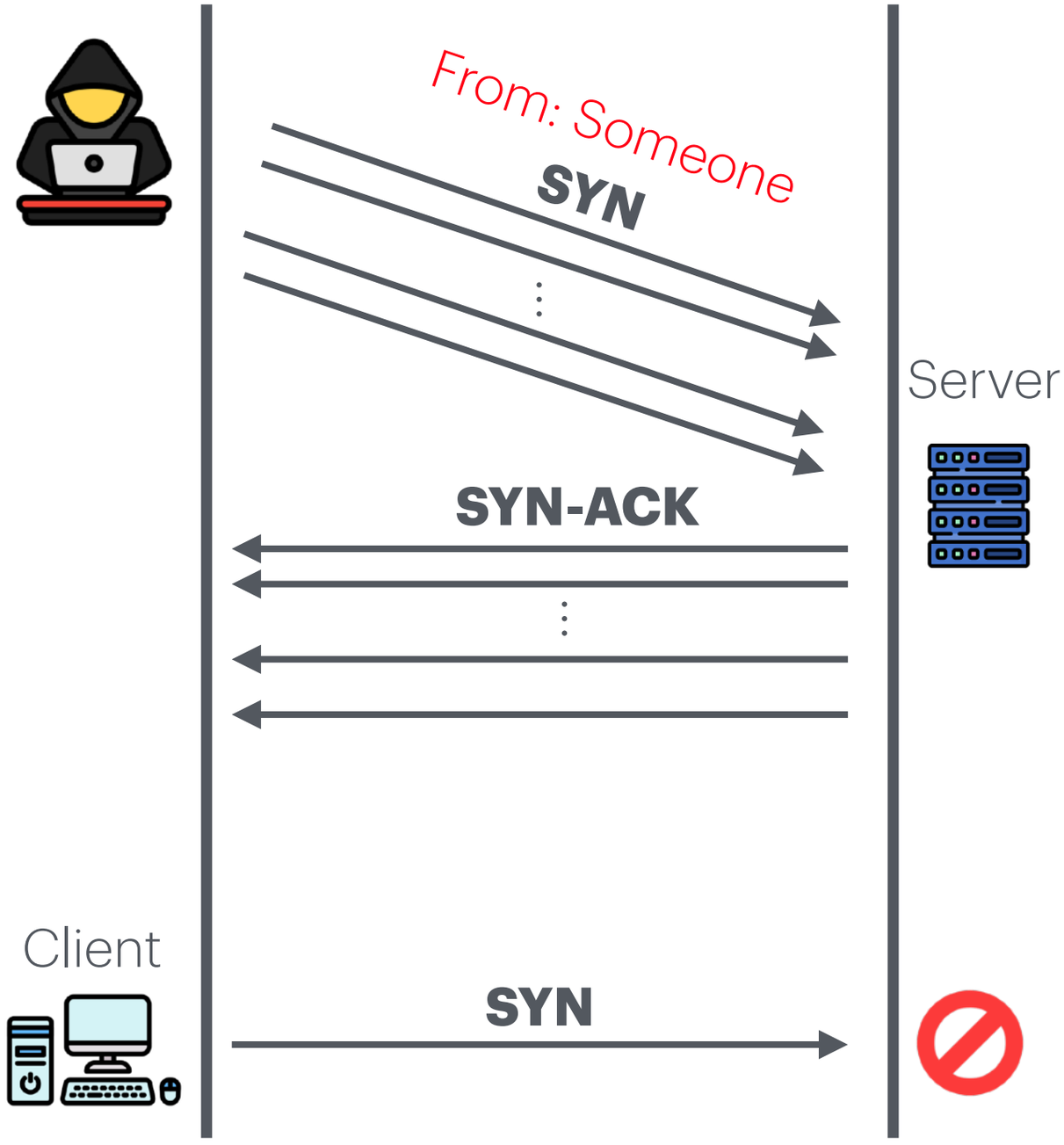
- **Link Layer:** send too much traffic for switches/routers to handle
- **TCP/UDP:** require servers to maintain large number of concurrent connections or state
- **Application Layer:** require servers to perform expensive queries or cryptographic operations

TCP Three-Way Handshake



TCP SYN Flooding

- Attacker sends many connection requests
  - May use spoofed source IP addresses
- Victim allocates resources for each request
  - Connection requests exist until timeout
- Resources exhausted ⇒ legitimate requests rejected

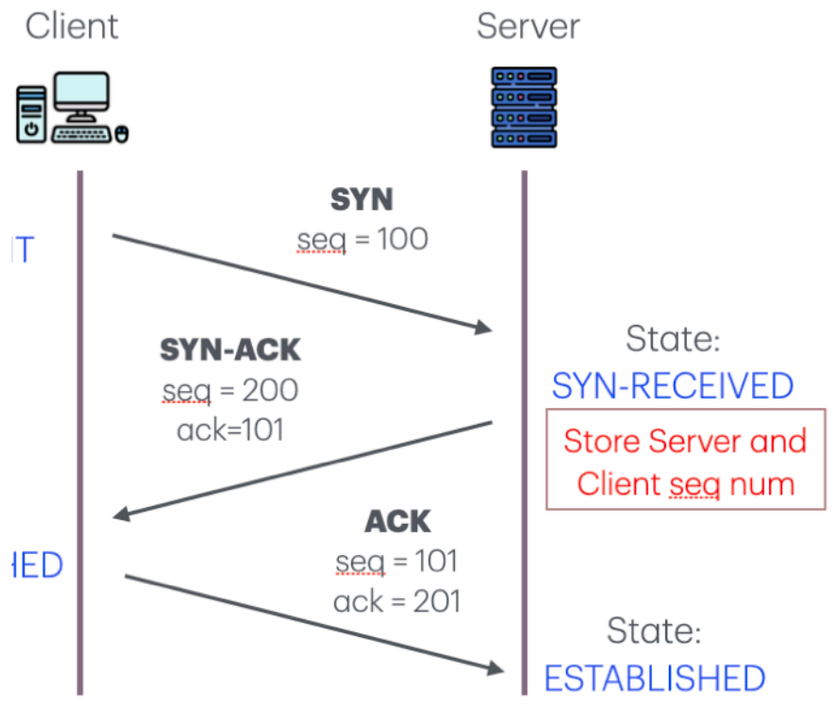


Core Problem

- **Problem:** server commits resources (memory) *before* confirming identify of the client (when client responds)
- **Bad Solution:**
  - Increase backlog queue size
  - Decrease timeout
- **Real Solution:** Avoid state until 3-way handshake completes

SYN Cookies

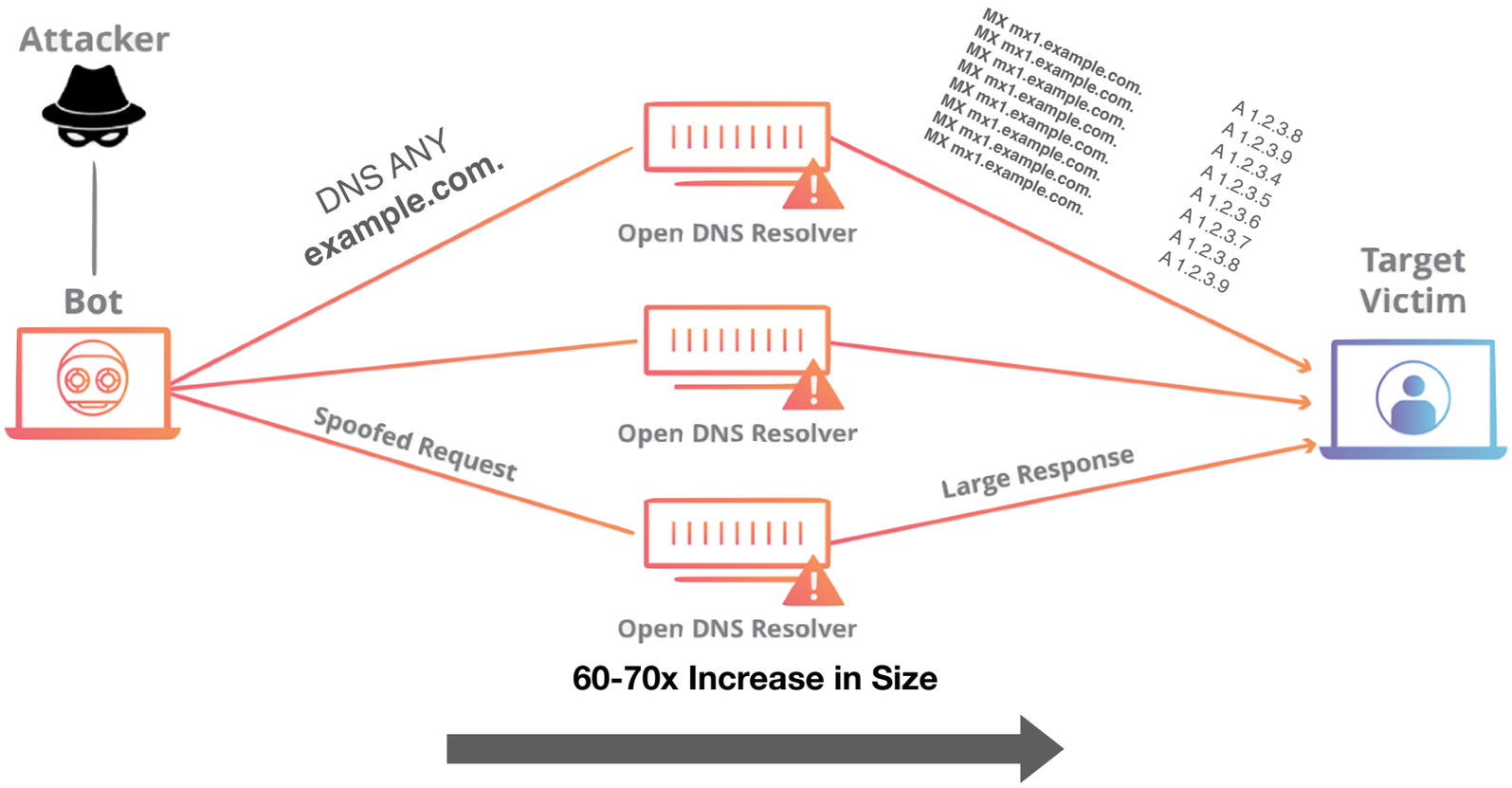
- **Idea:** Instead of storing Server seq SN_S and Client seq SN_C, the server sends a cookie back to the client.
- $L = \mathrm{MAC}_k(\mathsf{SAddr}, \mathsf{SPort}, \mathsf{CAddr}, \mathsf{CPort}, \mathsf{SN}_C, T)$
  - k: key picked at random during boot
  - T = 5-bit counter incremented every 64 secs.
  - $\mathsf{SN}_S = (T \,||\, \mathsf{mss} \,||\, L)$
- Honest client sends ACK=SN_S , seq=$\mathsf{SN}_C + 1$
- Server allocates space for socket only if valid SN_S



Server does not save state
(loses TCP options)

Amplification Attacks

- Services that respond to a single (*small*) UDP packet with a *large* UDP packet can be used to amplify DOS attacks
- Attacker forges packet and sets source IP to victim's IP address. When service responds, it sends large amount of data *to the spoofed victim*
- The attacker needs a large number of these services to amplify packets. Otherwise, the victim could just drop the packets from the small number of hosts



Common UDP Amplifiers

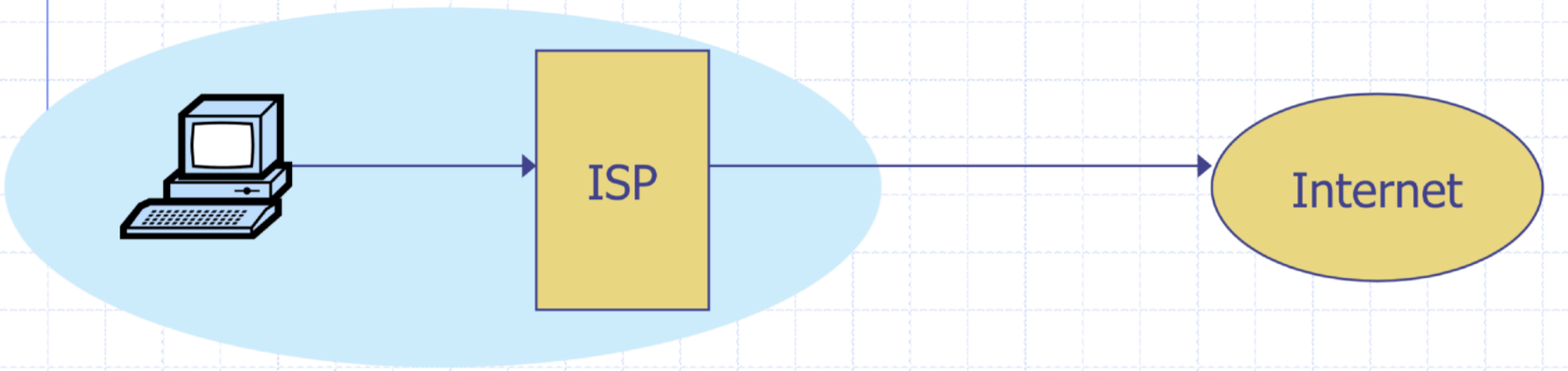
- Vulnerabilities
  - **DNS:** `ANY` query returns *all* records server has about a domain
  - **NTP**: `MONLIST` returns list of last 600 clients who asked for the time recently
- Countermeasure
  - **DNS**: Do not have recursive resolvers on the public Internet.
  - **NTP**: Do not respond to commands like `MONLIST`
- Both are considered misconfigurations today, but often 100Ks of misconfigured hosts on the public Internet

Amplification Attacks

- 2013: DDoS attack generated 300 Gbps (DNS)
  - 31,000 misconfigured open DNS resolvers, each at 10 Mbps
  - Source: 3 networks that allowed IP spoofing
- 2014: 400 Gbps DDoS attacked used 4,500 NTP servers

Ingress Filtering

- Big problem: DDoS with spoofed source IPs



- Ingress filtering policy: ISP only forwards packets with legitimate source IP (see also SAVE protocol)

Ingress Filtering

- **All ISPs need to do this — requires global coordination**
  - If 10% of networks don't implement, there's no defense
  - No incentive for an ISP to implement — doesn't affect them
- **As of 2017 (from CAIDA):**
  - 33% of autonomous systems allow spoofing
  - 23% of announced IP address space allow spoofing
- **2013:** 300 Gbps attack sent attack traffic from only 3 networks

THE WALL STREET JOURNAL.

October 21, 2016

Cyberattack Knocks Out Access to Websites

Popular sites such as Twitter, Netflix and PayPal were unreachable for part of the day

twitter

amazon web services™

PayPal

NETFLIX

SOUNDCLOUD

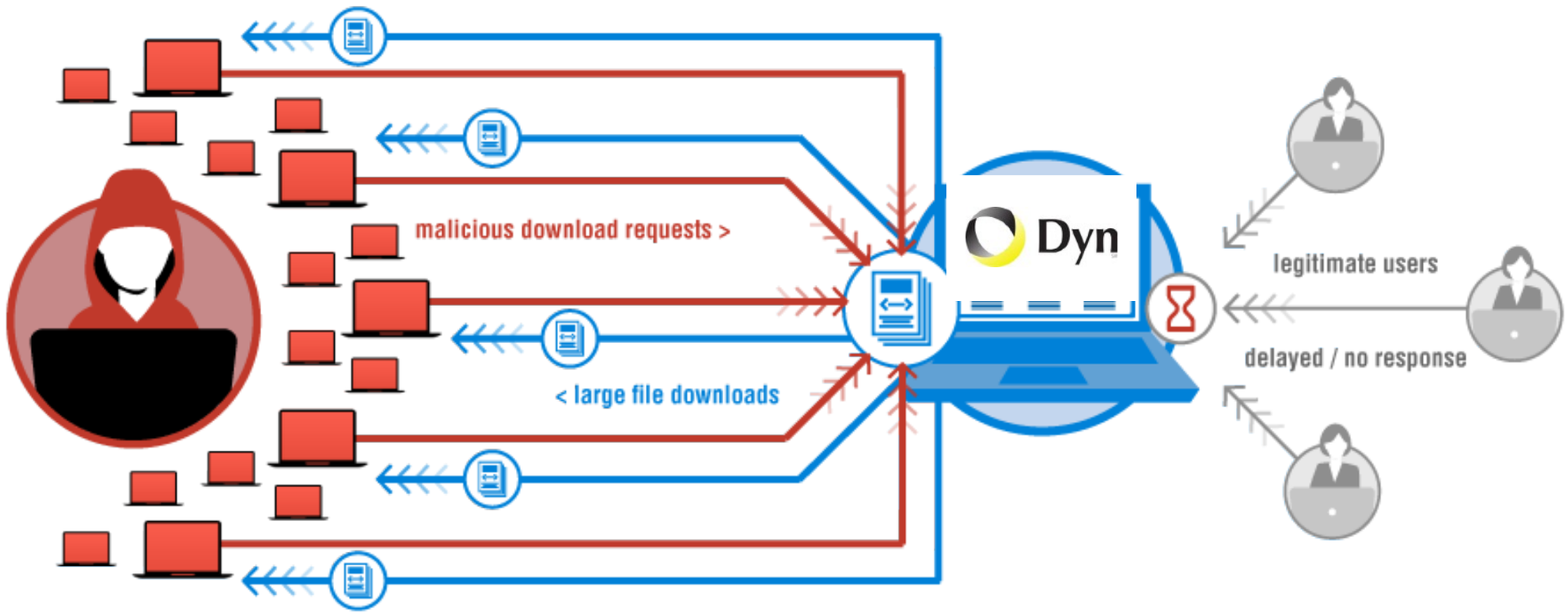
Spotify

GitHub

reddit

New York Times

DDoS on DNS provider



> *"We are still working on analyzing the data but the estimate at the time of this report is up to 100,000 malicious endpoints. [...] There have been some reports of a magnitude in the 1.2 Tbps range; at this time we are unable to verify that claim."*

A Botnet of IoT Devices

200K IoT devices

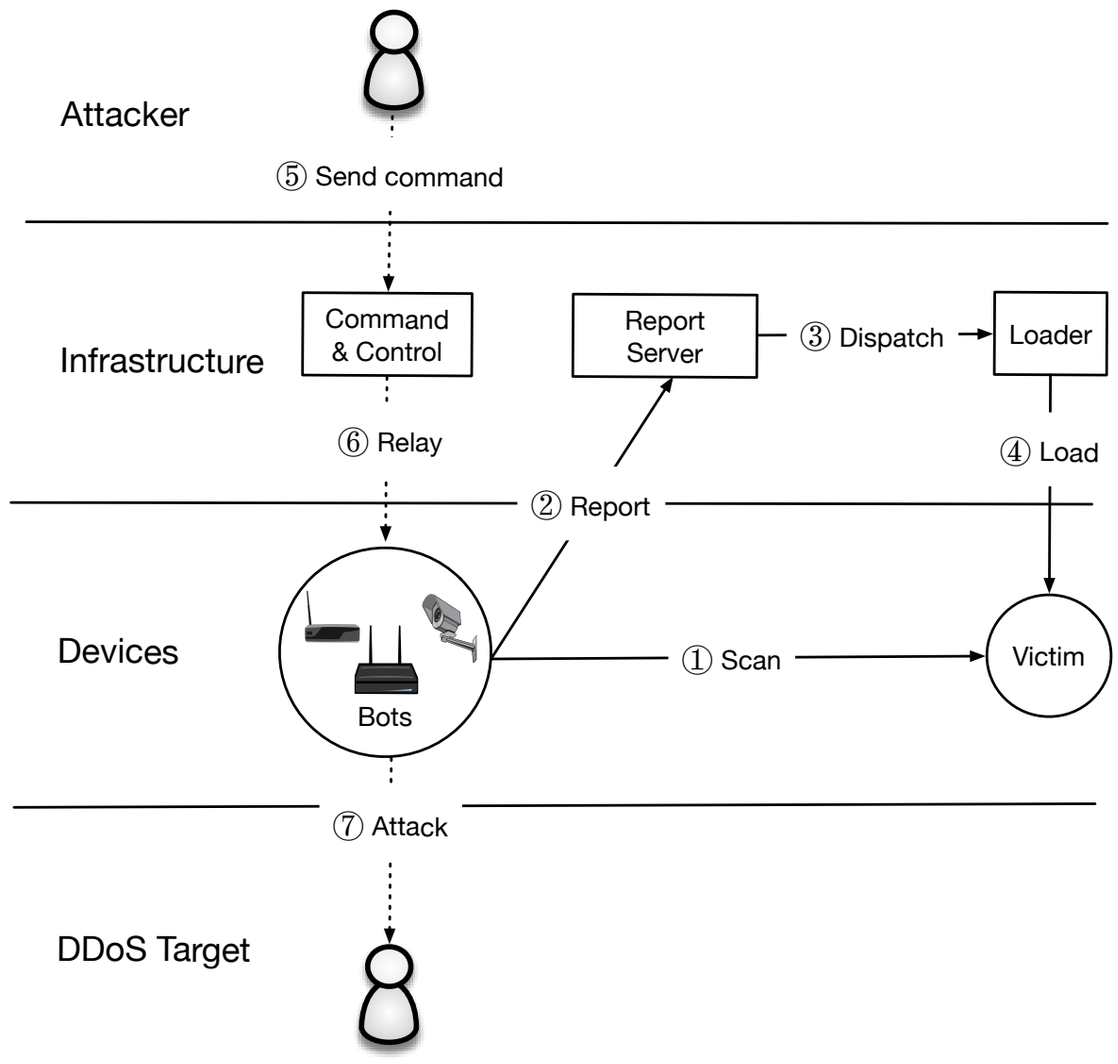
Not Amplification.
Flood with SYN, ACK, UDP, and GRE packets

The Mirai Malware

Bot master will issue commands to scan or start an attack

Attack Command:

- Action (e.g., START, STOP)
- Target IP(s)
- Attack Type (e.g., GRE, DNS, TCP)
- Attack Duration



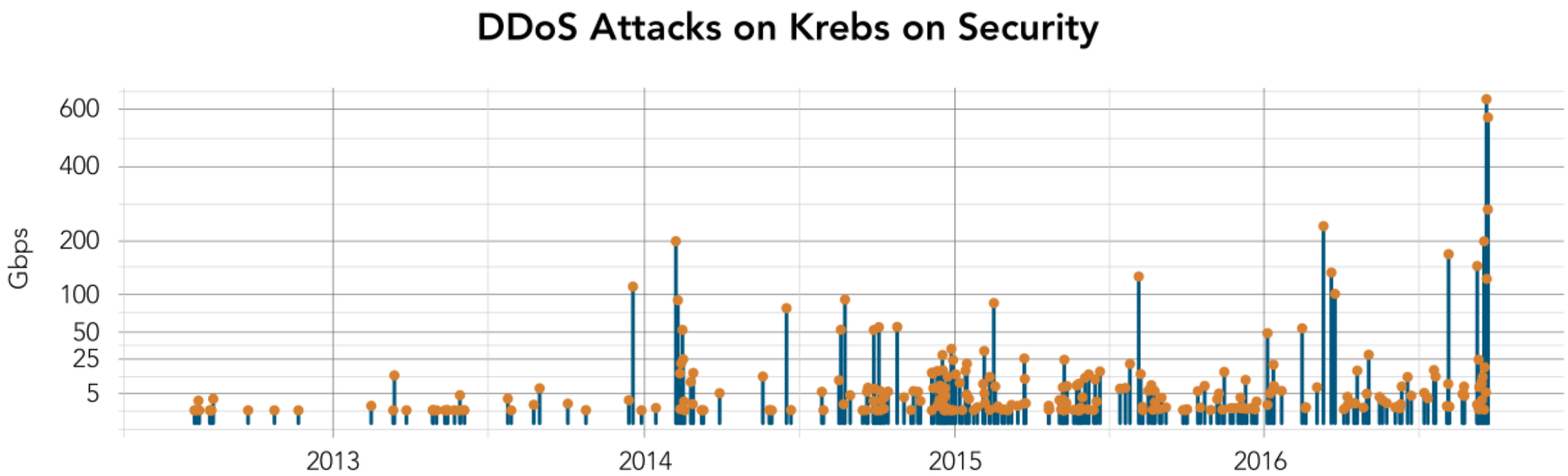
What made Mirai Successful?

- The Mirai malware is (astoundingly) badly written. It uses no new or complex techniques.
- Mirai was successful because:
  - IoT security bar is very low
  - Attack simplicity enabled the malware to compromise heterogeneous hardware
  - Stateless scanning was an improvement over prior versions

Password Guessing

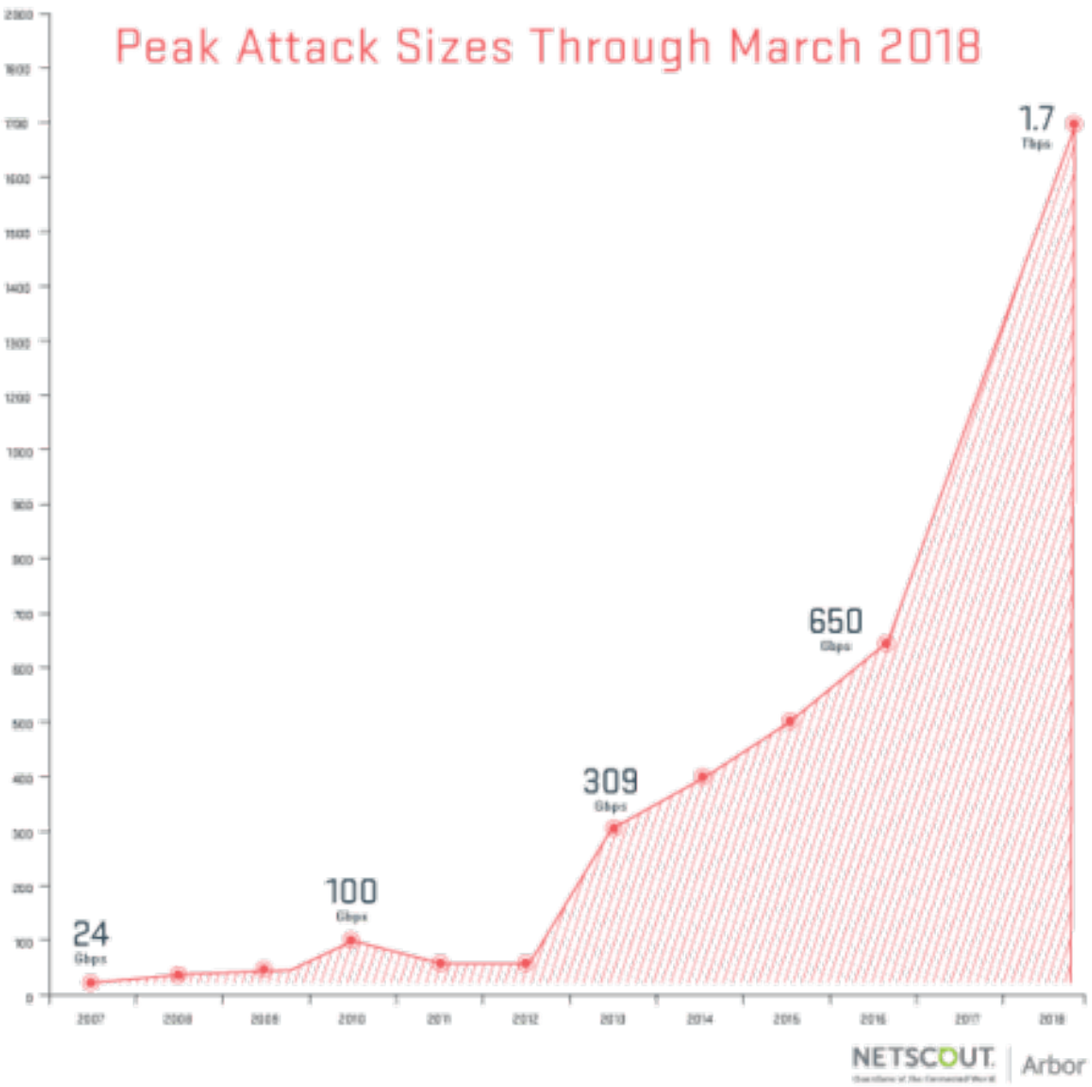
Password	Device Type	Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera	1111	Xerox Printer
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera	Zte521	ZTE Router
pass	Axis IP Camera	admin	IPX-DDK Network Camera	1234	Unknown
888888	Dahua DVR	system	IQinVision Cameras	12345	Unknown
666666	Dahua DVR	meinsm	Mobotix Network Camera	admin1234	Unknown
vizxv	Dahua IP Camera	54321	Packet8 VOIP Phone	default	Unknown
7ujMko0vizxv	Dahua IP Camera	00000000	Panasonic Printer	fucker	Unknown
7ujMko0admin	Dahua IP Camera	realtek	RealTek Routers	guest	Unknown
666666	Dahua IP Camera	1111111	Samsung IP Camera	password	Unknown
dreambox	Dreambox TV Receiver	xmhdipc	Shenzhen Anran Camera	root	Unknown
juantech	Guangzhou Juan Optical	smcadmin	SMC Routers	service	Unknown
xc3511	H.264 Chinese DVR	ikwb	Toshiba Network Camera	support	Unknown
OxhlwSG8	HiSilicon IP Camera	ubnt	Ubiquiti AirOS Router	tech	Unknown
cat1029	HiSilicon IP Camera	supervisor	VideoIQ	user	Unknown
hi3518	HiSilicon IP Camera	<none>	Vivotek IP Camera	zlxx.	Unknown
klv123	HiSilicon IP Camera				

DoS on Krebs' Blog



> *"The magnitude of the attacks seen during the final week were significantly larger than the majority of attacks Akamai sees on a regular basis. [...] In fact, while the attack on September 20 was the largest attack ever mitigated by Akamai, the attack on September 22 would have qualified for the record at any other time, peaking at 555 Gbps."*

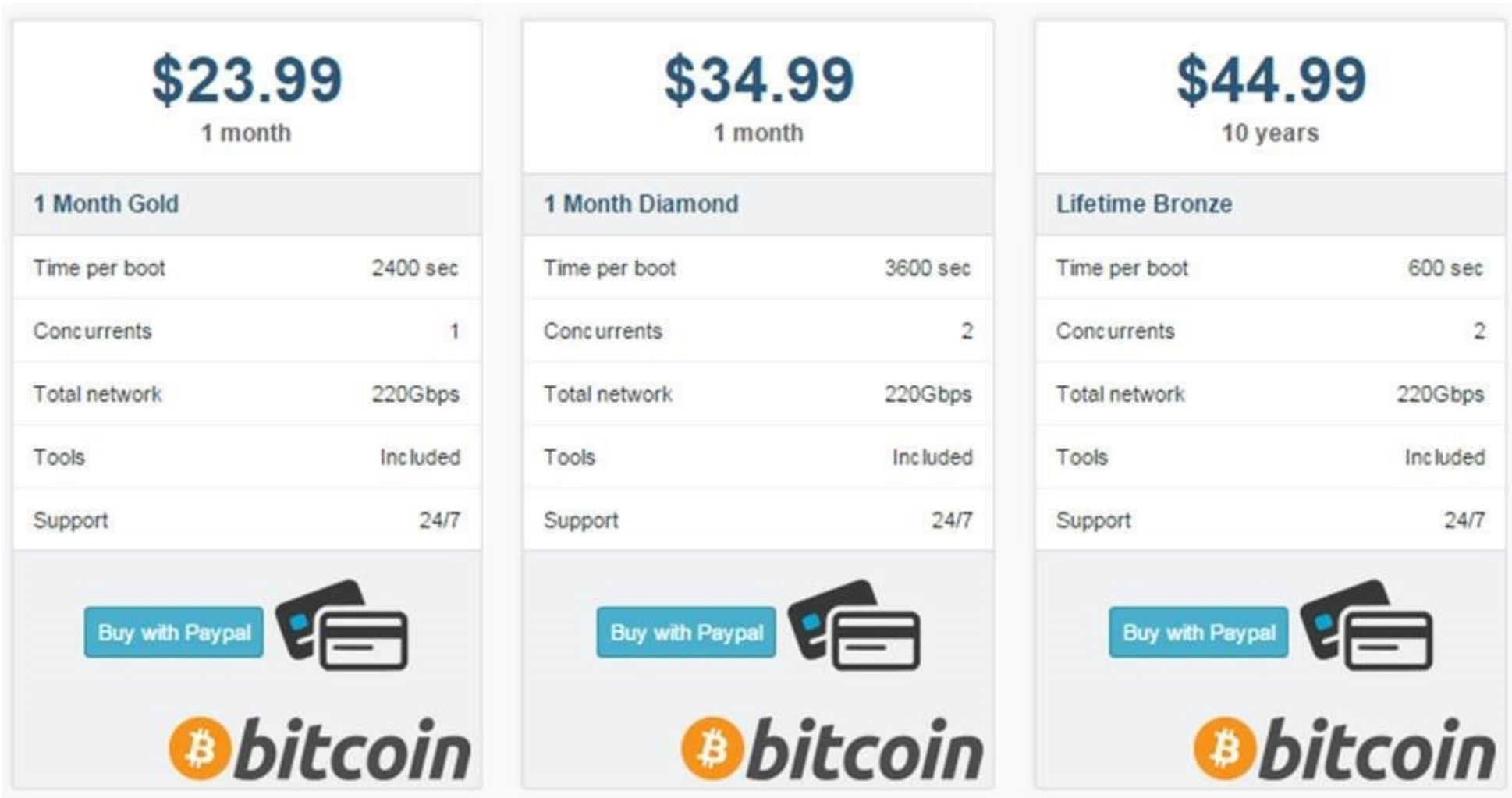
Memcache



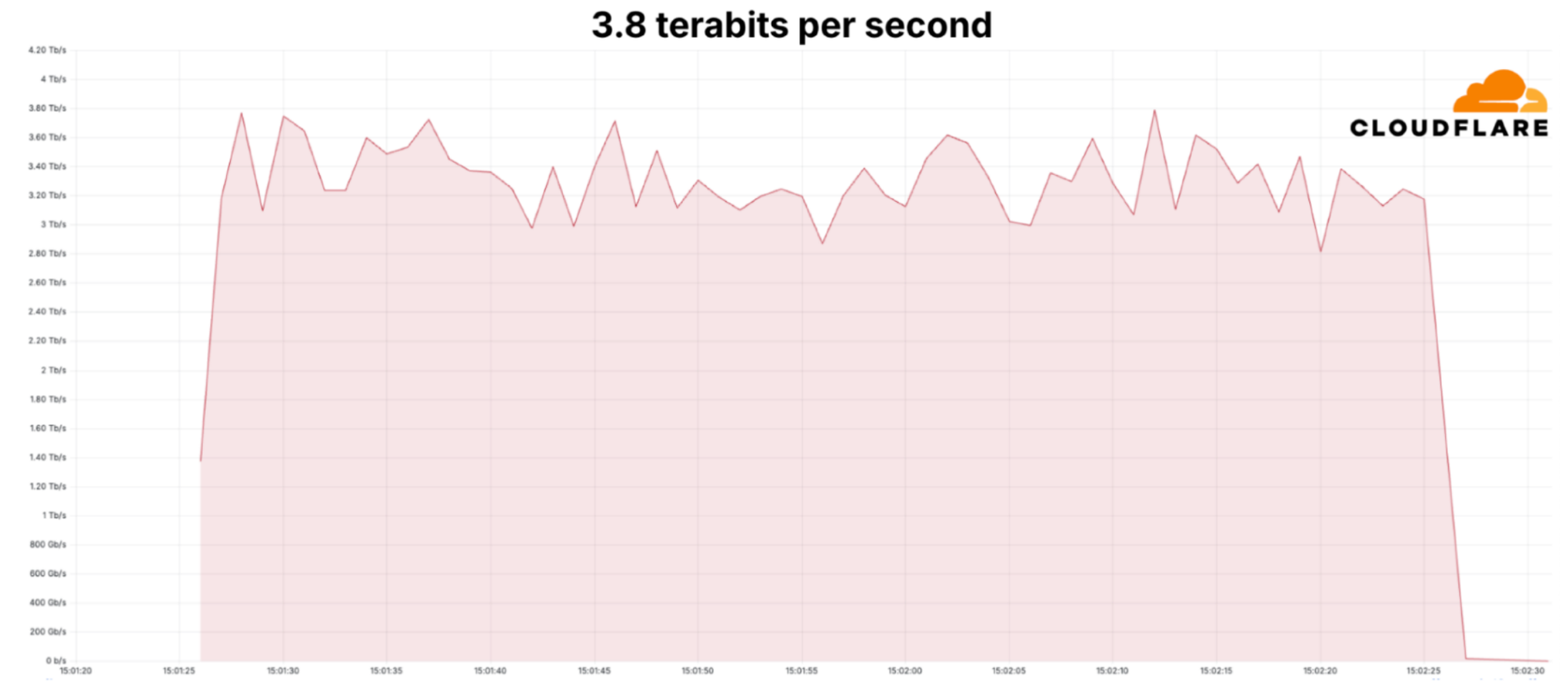
- **Memcache**: retrieve large record
- The server responds by firing back as much as 50,000 times the data it received.
- Exist both a UDP and TCP version. Only works for UDP! TCP would require a three-way handshake and server would realize IP had been spoofed.

Stresser (Booter) Services

Stresser (or **booter**) services provide DoS attack as a service, usually as a criminal enterprise



More recent DDoS records: 3.8 Tbps



October 02, 2024

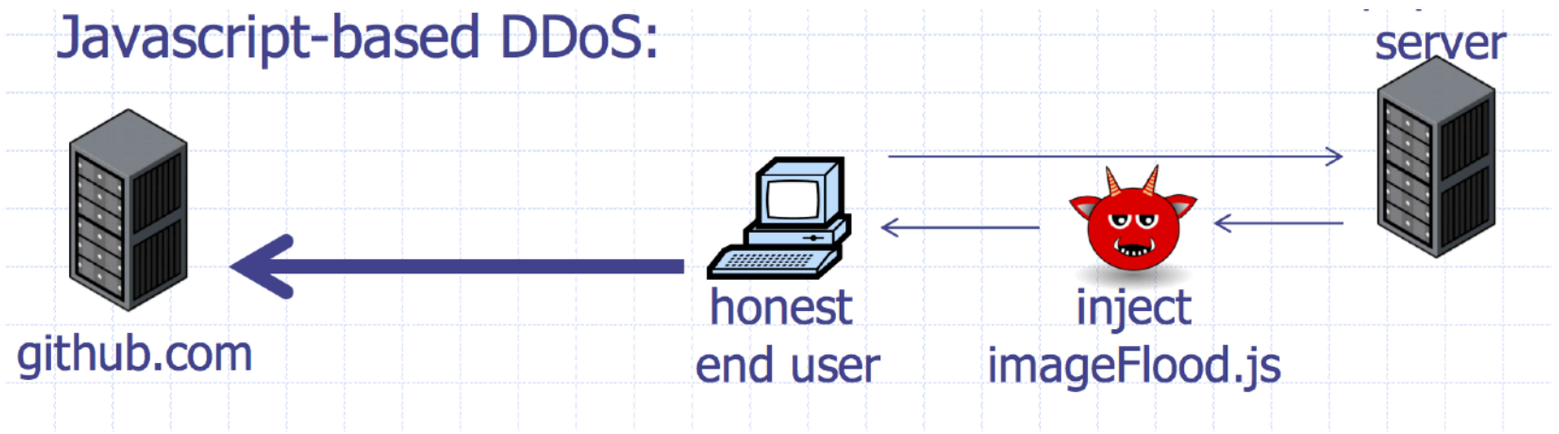
Moving Up Stack: GET Floods

- Command bot army to:
  - Complete real TCP connection
  - Complete TLS Handshake
  - `GET` large image or other content
- Will bypass flood protections, but attacker can no longer use random source IPs
- Victim site can block or rate limit bots

Github Attacks

1.35 Tbps attack against Github caused by JS injected into web requests

National players were widely suspected to be behind the attack



More reason that you should <u>always</u> use HTTPS!

Defense against DoS Attacks

Defend against DDoS

- Individuals usually won't have the resource & skill to defend by themselves.

- Best options if you want to host any online service:

- Cloud-based DDoS Protection: Cloudflare, Akamai, AWS, Google Cloud, etc

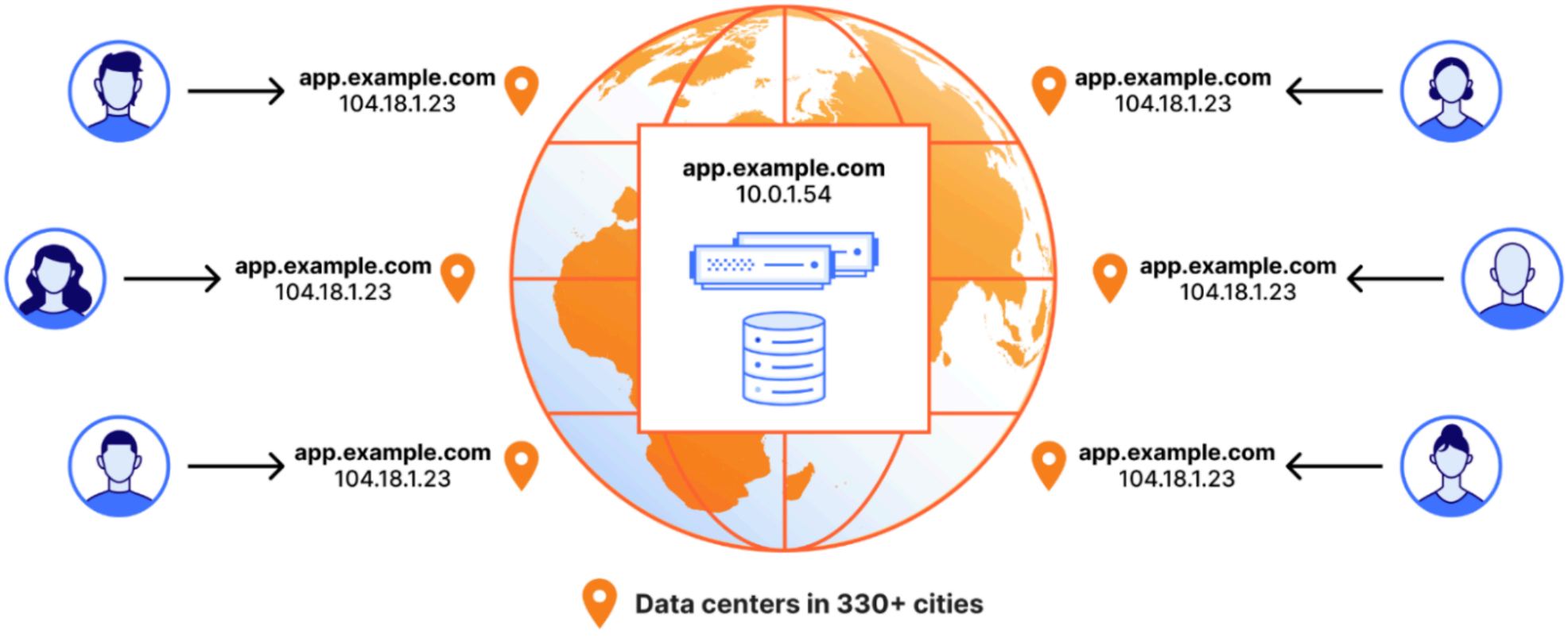
- Route traffic through a VPN and set rate limit on firewalls.

- Use CDN services: Cloudflare, Akamai, etc.

Defend against DDoS

Cloudflare's DDoS protection system

- **Anycast network**: a single IP address to be advertised by multiple machines around the world.
    - Automatic Load Balancing: A packet sent to that IP address will be served by the closest machine.



Defend against DDoS

Cloudflare's DDoS protection system

- **Traffic analysis**: Sampling, Analyzing, and Dropping packets in realtime.

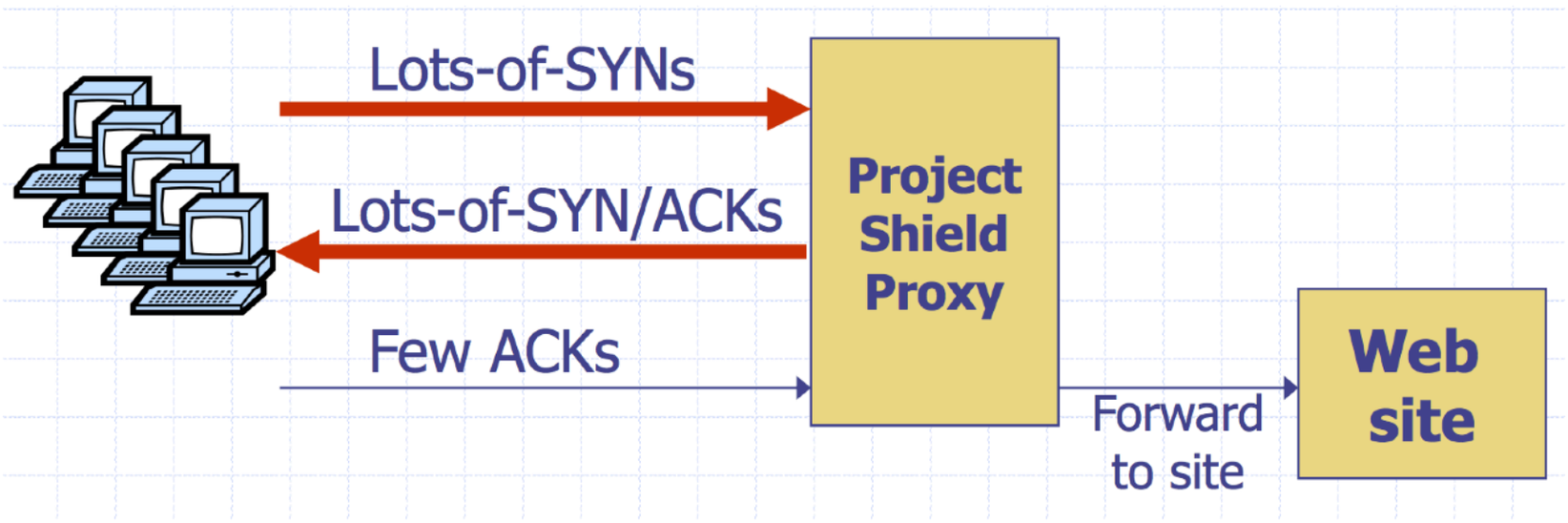
  - Packets from a DDoS attack usually have very different statistic characteristics from normal traffic, which can be easily recognized via rule-based or ML-based models.

  - Packet filters like eBPF can process (in particular, drop) packet at the NIC level (between Physical Layer and Link Layer), this saves lots of *CPU resources*.

Defend against DDoS

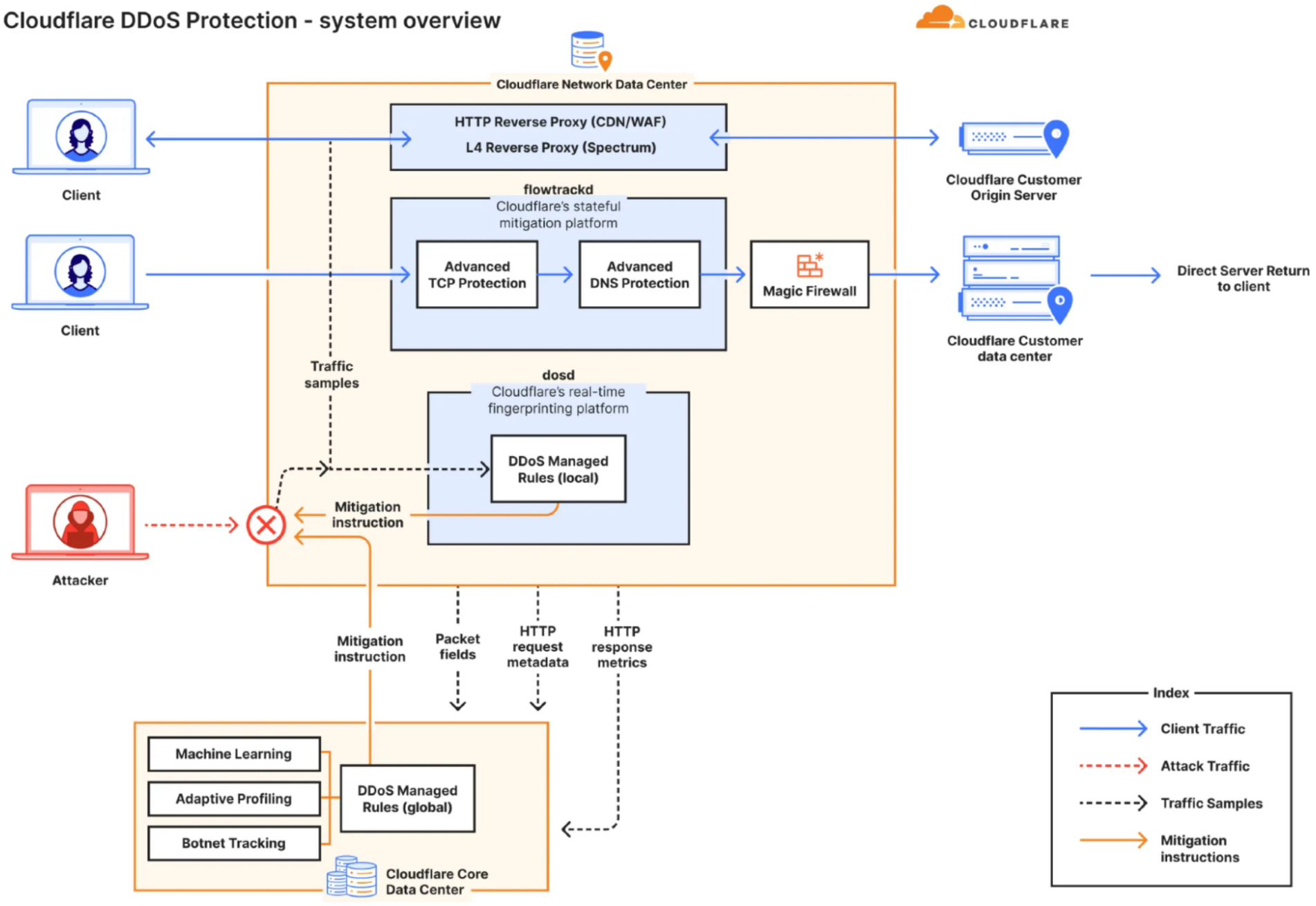
Google Project Shield

- **Reverse proxy**: a server that sits in front of one or more web servers, intercepting requests from clients.
    - Google Project Shield uses Google bandwidth to shield vulnerable websites (e.g., news, blogs, human rights orgs)



Defend against DDoS

Cloudflare's DDoS protection system



Client Puzzles

Idea: What if we force every client to do moderate amount of work for every connection they make?

Example:

1) Server Sends: C

2) Client: find $X \,|\, \text{LSB}_n(\text{SHA1}(C\,||\,X)) = 0^n$

Assumption:

Puzzle takes 2^n for the client to compute (0.3 s on 1Ghz core)

Solution is trivial for server to check (single SHA-1 hash)

Client Puzzles

Not frequently used in the real world

Benefits:

- Can change *n* based on amount of attack traffic

Limitations:

- Requires changes to both protocols, clients, and servers

- Hurts low power legitimate clients during attack (e.g., phones)

Questions?