CS 253 Cyber Security
Network/Internet Security

ShanghaiTech University

Admin

- HW 3 was released on 11/19. DDL: 11/26 by midnight.

- Correction: in the bonus problem, the “*” mark is not correctly shown on Gradescope. It is in fact LEN*256.

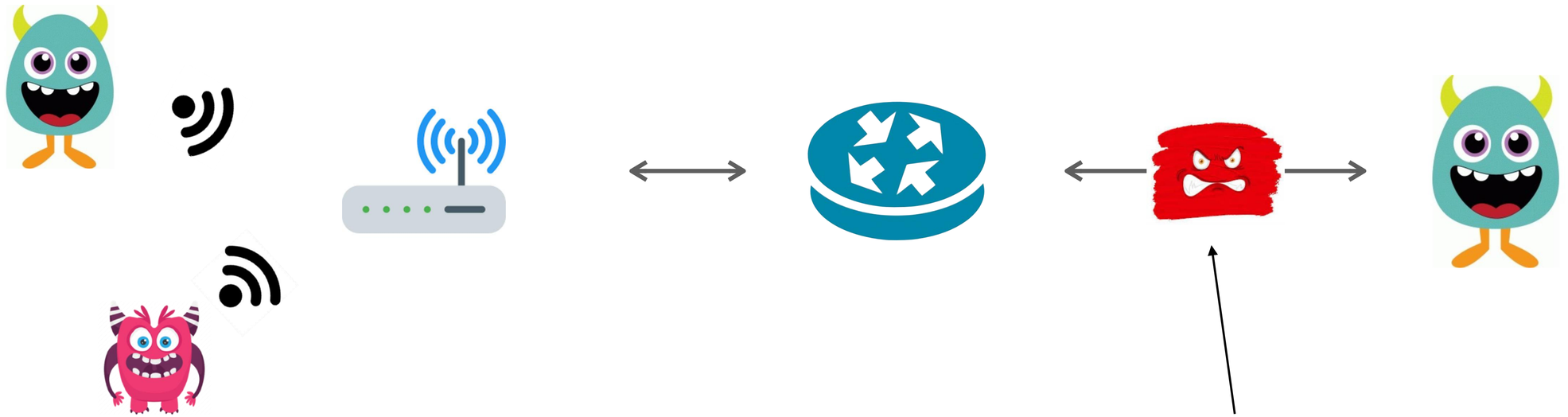
> Describe a simple attack that lets a local attacker outside the enclave extract *correctPwd from the enclave using at most* *LEN 256* login calls into the enclave. You don't have to answer this question unless you want to get the bonus 2 points to the total grade.

PART ONE

01

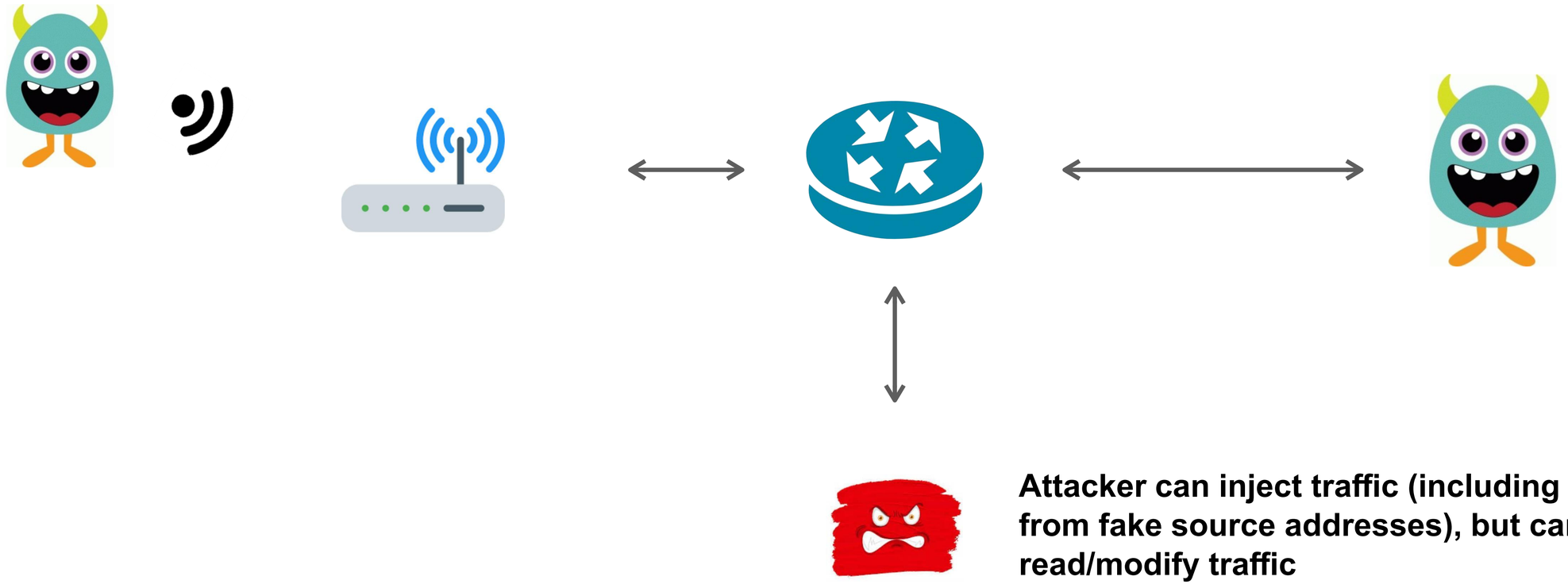
Network Security Overview

Notation: On Path Attacker



Attacker has access to read, manipulate, and drop traffic because they are on the path that the traffic takes across the Internet

Notation: Off Path Attacker



Attacker can inject traffic (including from fake source addresses), but can’t read/modify traffic

No security guarantees

Confidentiality — Ethernet, IP, UDP, and TCP do not provide any confidentiality. All traffic is in cleartext.

On-path attacker can do anything. ARP and BGP attacks allow an off-path attacker to become on-path and MITM connections.

Integrity — No guarantees that attacker hasn't modified traffic. Ethernet, IP and UDP have no protection against spoofed packets. TCP provides *weak* guarantee of source authentication against off-path attacker

Availability — Attackers can attempt to inject packets or launch "denial of service" attacks against services

Assume network is malicious

The network is out to get you.

Solution: Always use TLS if you want any protection against large-scale eavesdropping or guarantee that data hasn't been modified or corrupted by an on-path (or off-path since less strong) attacker

Note! HTTPS and TLS aren't just for sensitive material! There have been attacks where malicious Javascript or malware is injected into websites.

ARP: Address Resolution Protocol

ARP lets hosts to find each others' MAC addresses on a local network. For example, when you need to send packets to the upstream router to reach Internet hosts

Client: Broadcast (all MACs): Which MAC address has IP 192.168.1.1?
Response: I have this IP address (sent from correct MAC)

No built-in security. Attacker can impersonate a host by faking its identity and responding to ARP requests or sending gratuitous ARP announcements

IP: Internet Protocol

Provides routing between hosts on the Internet. Unreliable. Best Effort.

- Packets can be dropped, corrupted, repeated, reordered

Routers simply route IP packets based on their destination address.

- Must be simple in order to be fast — insane number packets FWD'ed

No inherent security. Packets have a checksum, but it's non-cryptographic. Attackers can change any packet.

Source address is set by sender—can be faked by an attacker

BGP (Border Gateway Protocol)

Internet Service Providers (ISPs) announce their presence on the Internet via BGP. Each router maintains list of routes to get to different announced prefixes

No authentication—possible to announce someone else's network

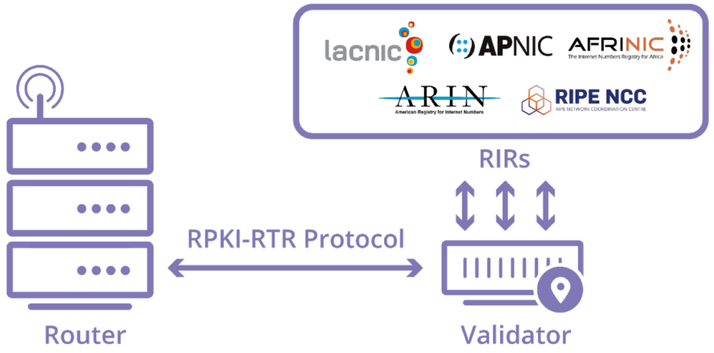
Commonly occurs (often due to operator error but also due to attacks)

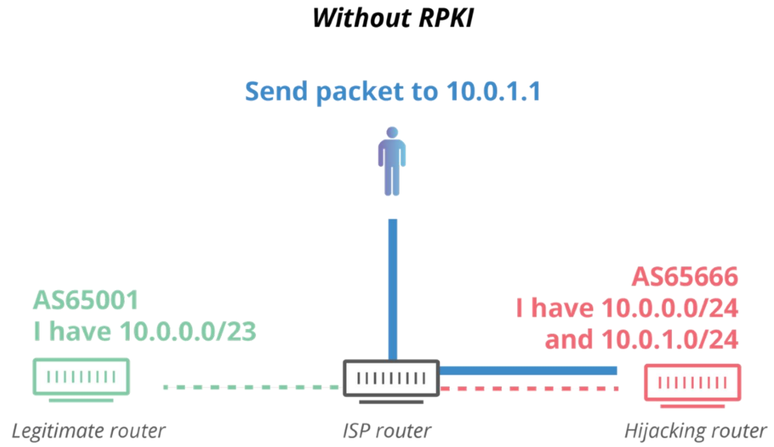
Resource Public Key Infrastructure (RPKI)

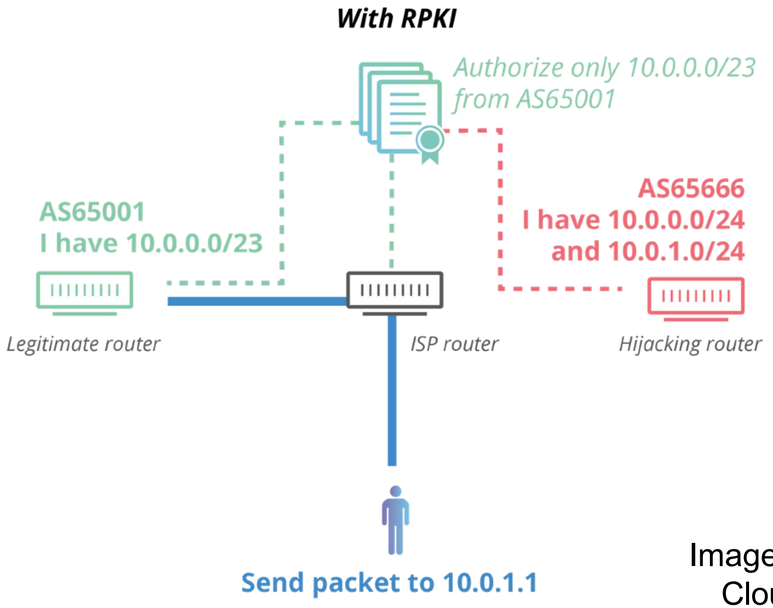
RPKI is a relatively new PKI to help improve BGP security

Networks ask regional registrars to sign a “Route Origin Authorization” that indicates a specific ASN is allowed to advertise a given IP range

Networks validate signed ROA against the PKI before deciding to accept a new advertisement







Images from Cloudflare

02

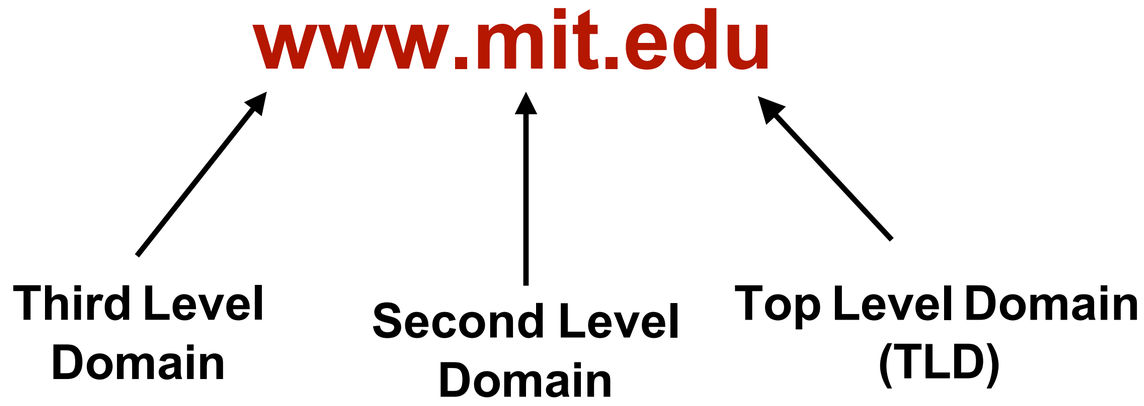
PART TWO

DNS Security

DNS (Domain Name System)

Application-layer protocols (and people) usually refer to Internet host by host name (e.g., google.com)

DNS is a delegatable, hierarchical name space



DNS Security

Users/hosts trust the host-address mapping provided by DNS

- Used as basis for many security policies:
  - Browser same origin policy, URL address bar

Interception of requests or compromise of DNS servers can result in incorrect or malicious responses

Caching

DNS responses are cached

- Quick response for repeated translations
- NS records for domains also cached

DNS negative queries are cached

- Save time for nonexistent sites, e.g. misspelling

Cached data periodically times out

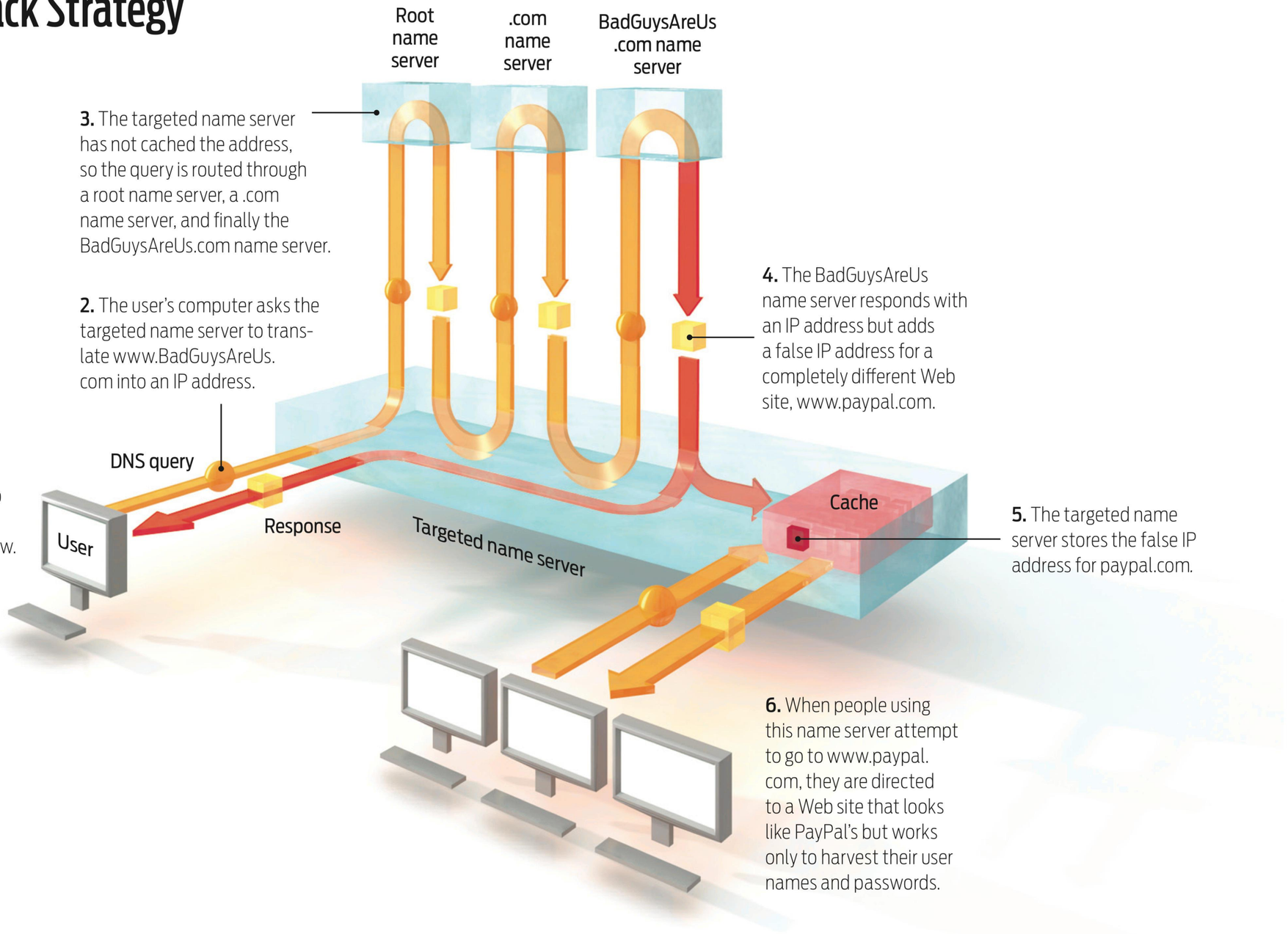
- Lifetime (TTL) of data controlled by owner of data
- TTL passed with every record

DNS Cache Poisoning

DNS query results include Additional Records section
– Provide records for anticipated next resolution step

Early servers accepted and cached all additional records provided in query response

Early Attack Strategy



Glue Records

Can we just stop using additional section?

- Only accept answers from authoritative servers?

Glue records: non-authoritative are records necessary to contact next hop in resolution chain

- Necessary given current design of DNS

Bailiwick Checking: Only accept additional records that are for a domain in the original question.

DNS Spoofing

Scenario: DNS client issues query to server

Attacker would like to inject a fake reply
  Attacker does not see query or real response

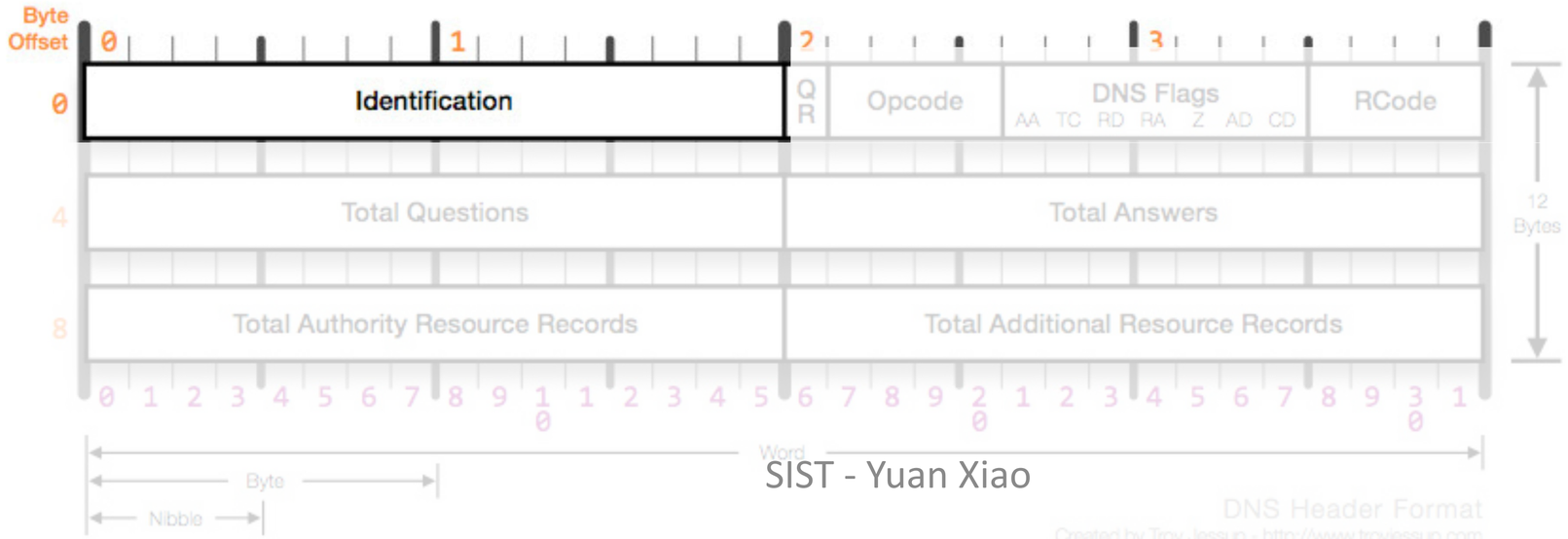
How does client authenticate response?

DNS Spoofing

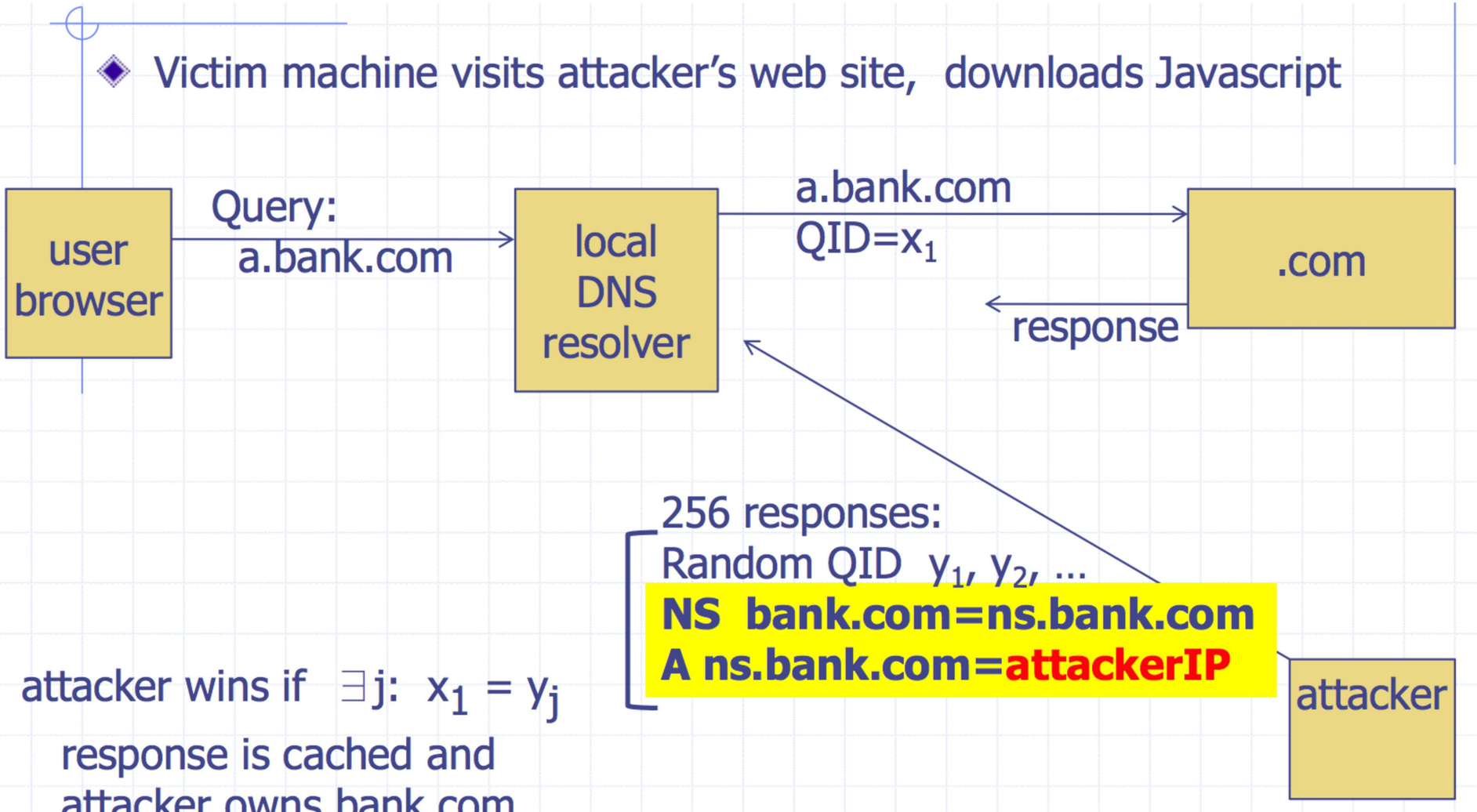
How does client authenticate response?

UDP port numbers must match
Destination port usually port 53 by convention

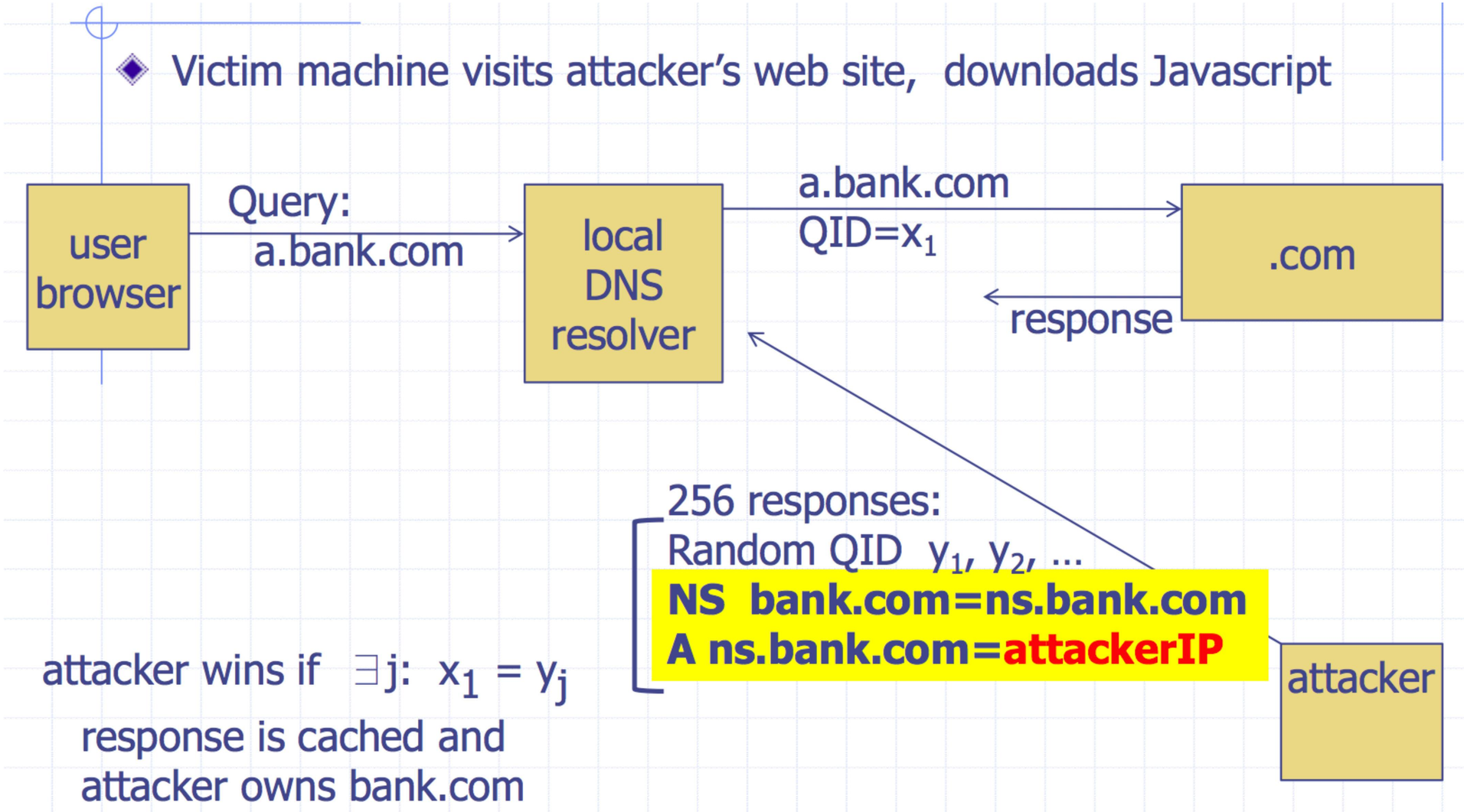
16-bit query ID must match



Kaminsky Attack



Try Again!


Victim machine visits attacker's web site, downloads Javascript
user browser
Query:
a.bank.com
local DNS resolver
a.bank.com
QID=x1
.com
response
256 responses:
Random QID y1, y2, ...
NS bank.com=ns.bank.com
A ns.bank.com=attackerIP
attacker
attacker wins if ∃j: x1 = yj
response is cached and
attacker owns bank.com

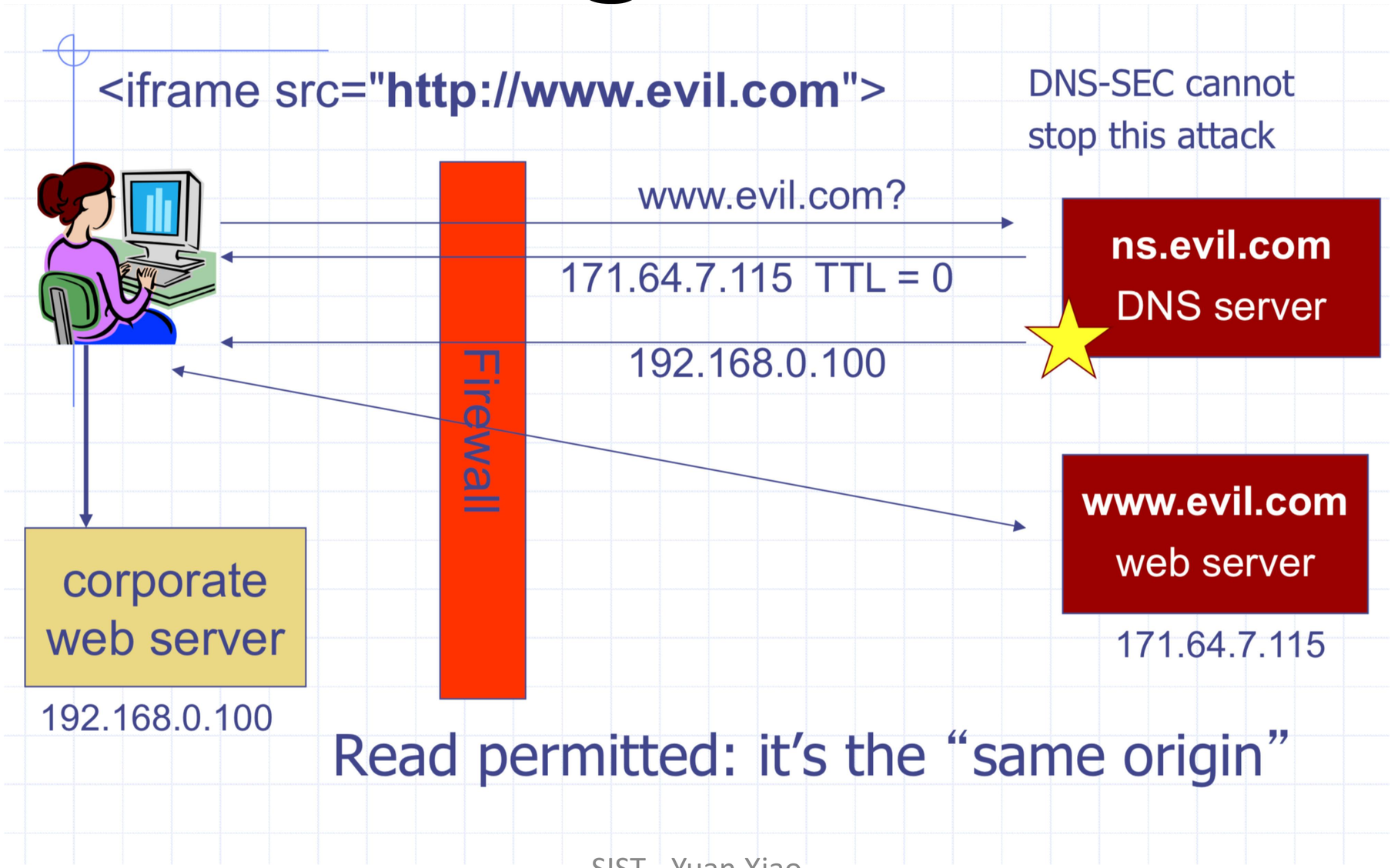
Defenses

Increase QueryID space. But how? Don’t want to change packet.

Randomize src port, additional 11 bits of entropy

- Attack now takes several hours

DNS Rebinding



Rebinding Defenses

Browser Mitigations:

- Refuse to switch IPs mid session
- Interacts poorly with proxies, VPNs, CDNs, etc
- Not consistently implemented in any browser

Server Defenses

- Check Host header for unrecognized domains
- Authenticate users with something else beyond IP address

DNSSEC

Adds authentication and integrity to DNS responses

Authoritative DNS servers sign DNS responses using cryptographic key

Clients can verify that a response is legitimate by checking signature through PKI similar to HTTPS

Most people don’t use DNSSEC and never will. Use TLS instead.

PART THREE

03

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

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

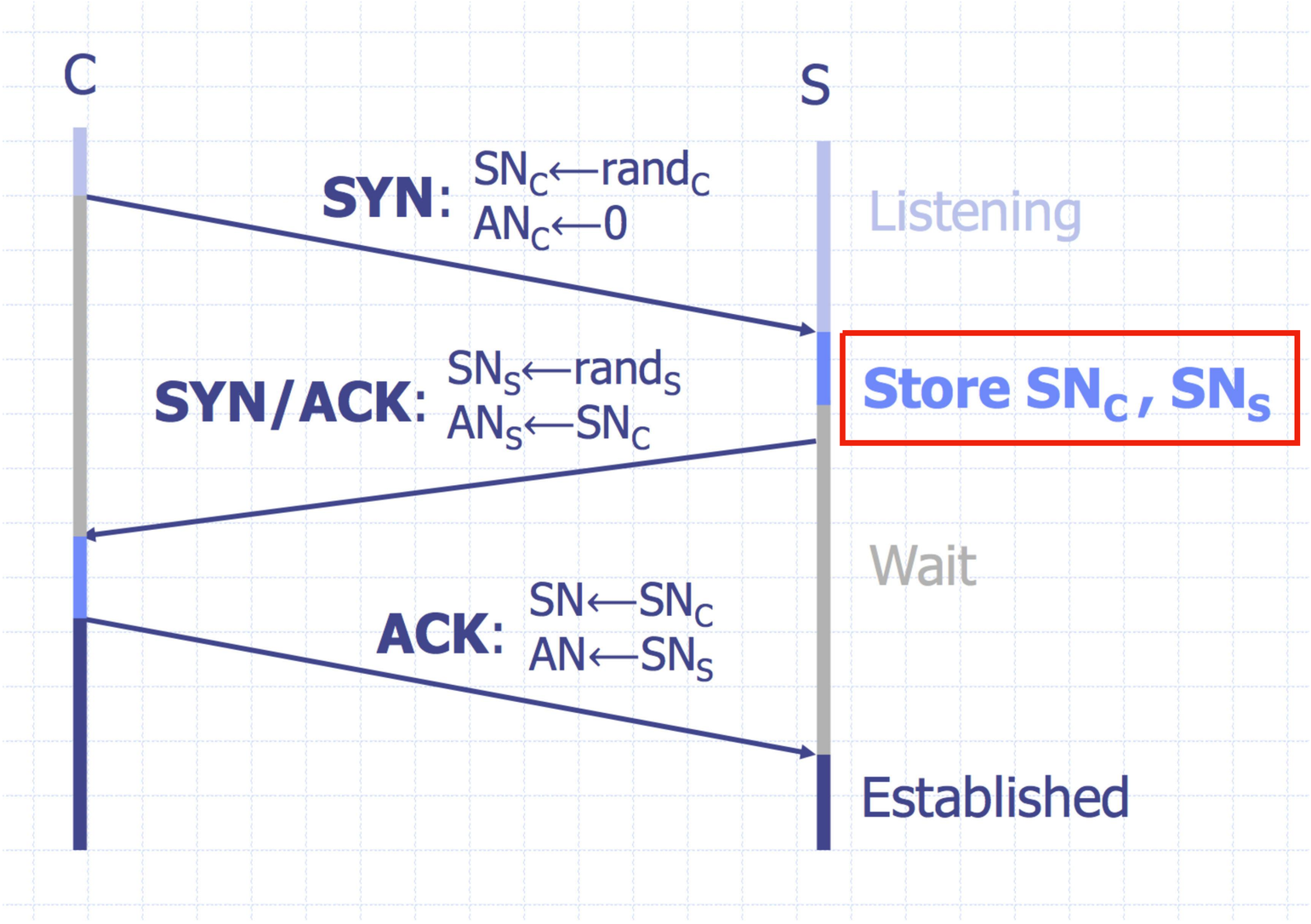
DoS Opportunities at Every Layer

Link Layer: send too much traffic for switches/routers to handle

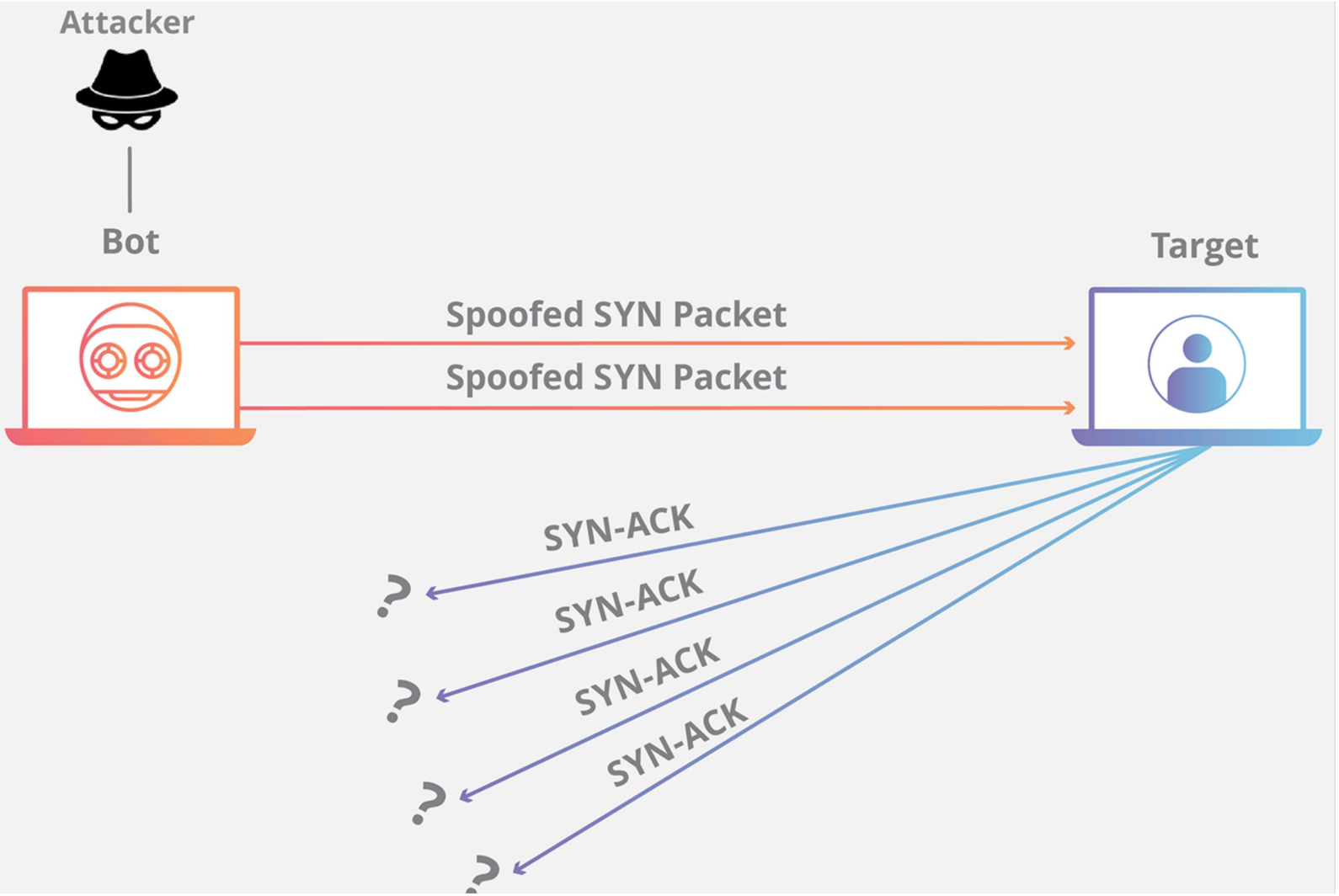
TCP/UDP: require servers to maintain large number of concurrent connections or state

Application Layer: require servers to perform expensive queries or cryptographic operations

TCP Handshake



SYN Floods


Attacker
Bot
Target
Spoofed SYN Packet
Spoofed SYN Packet
SYN-ACK
SYN-ACK
SYN-ACK
SYN-ACK
?
?
?
?

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 SN_c and SN_s…
send a cookie back to the client.

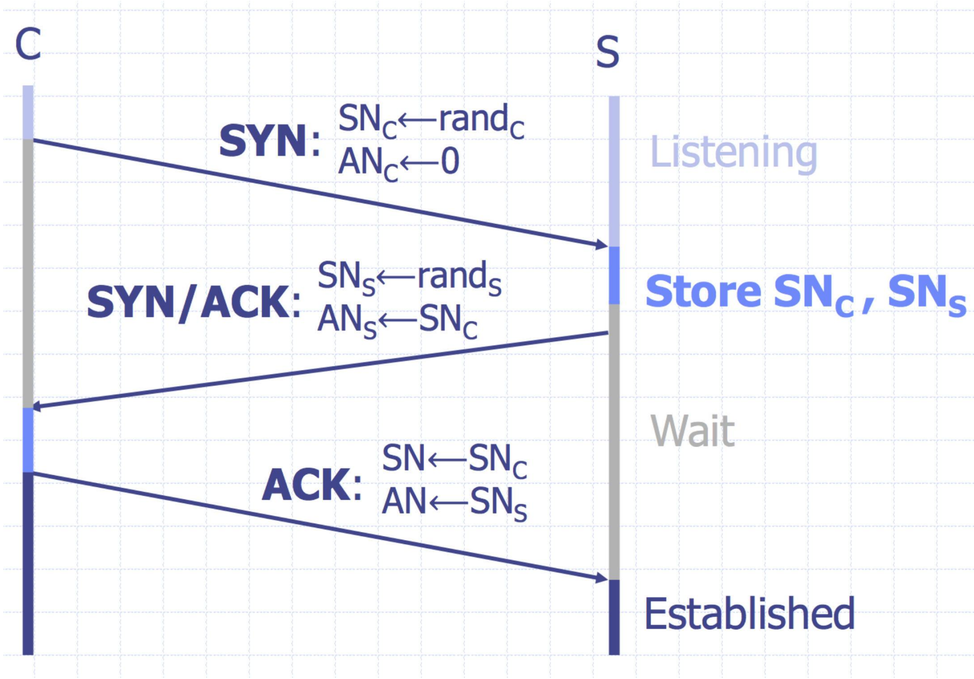
$L = MAC_{key}(SAddr, SPort, DAddr, DPort, SN_C, T)$

key: picked at random during boot

T = 5-bit counter incremented every 64 secs.

$SN_s = (T \,||\, mss \,||\, L)$

Honest client sends ACK ($AN=SN_s$, $SN=SN_C+1$)
Server allocates space for socket only if valid SNs



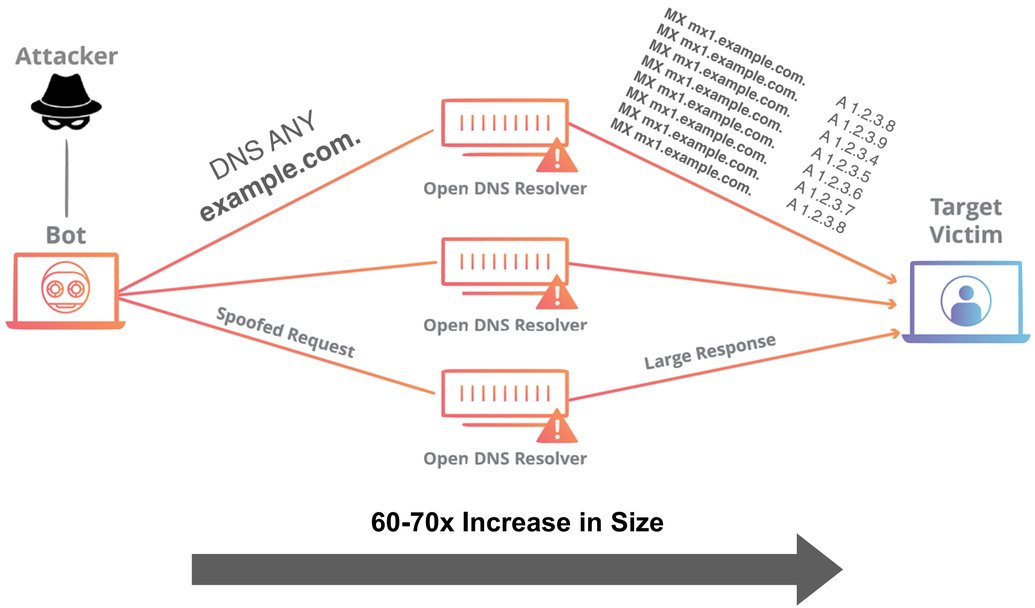
Server does not save state
(loses TCP options)

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

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

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

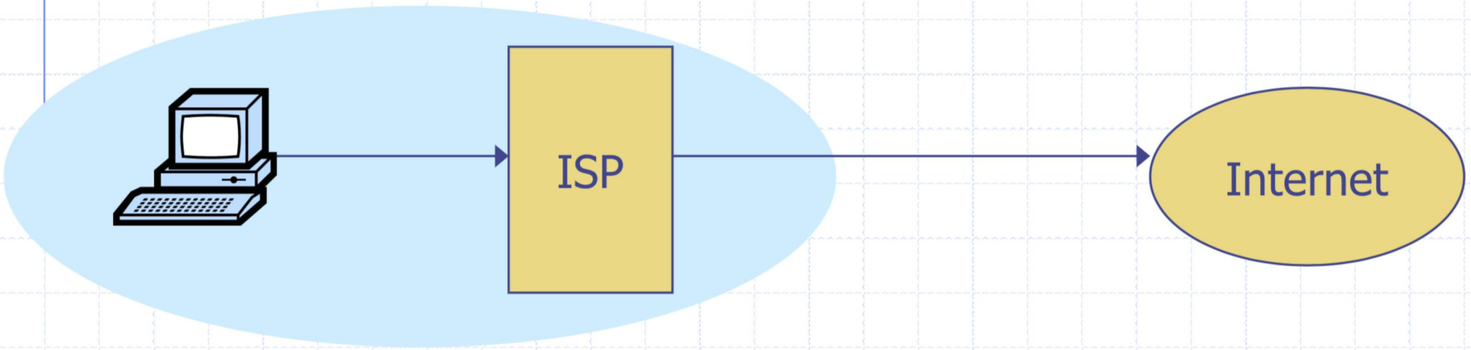
2013: DDoS attack generated 300 Gbps (DNS)

- 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

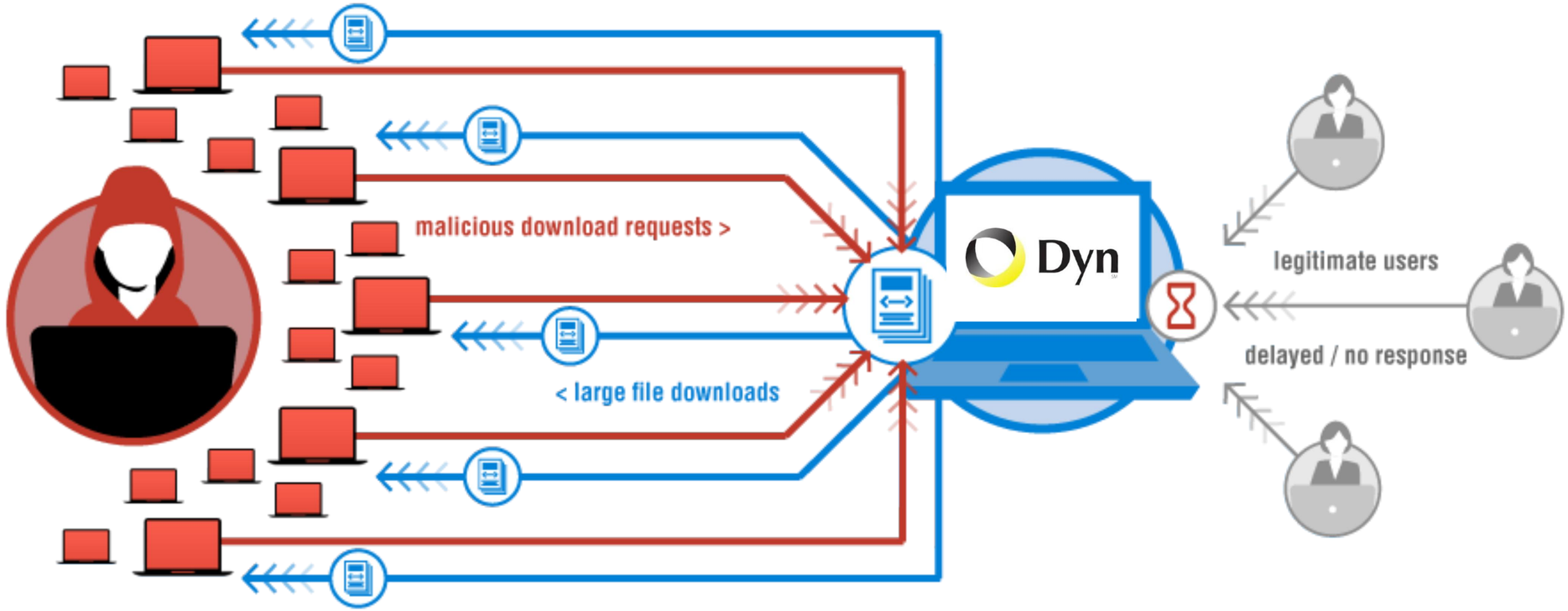
SOUNDCLOUD

Spotify

GitHub

reddit

New York Times



> “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.”

Image: Verisign

A Botnet of IoT Devices

200K IoT devices

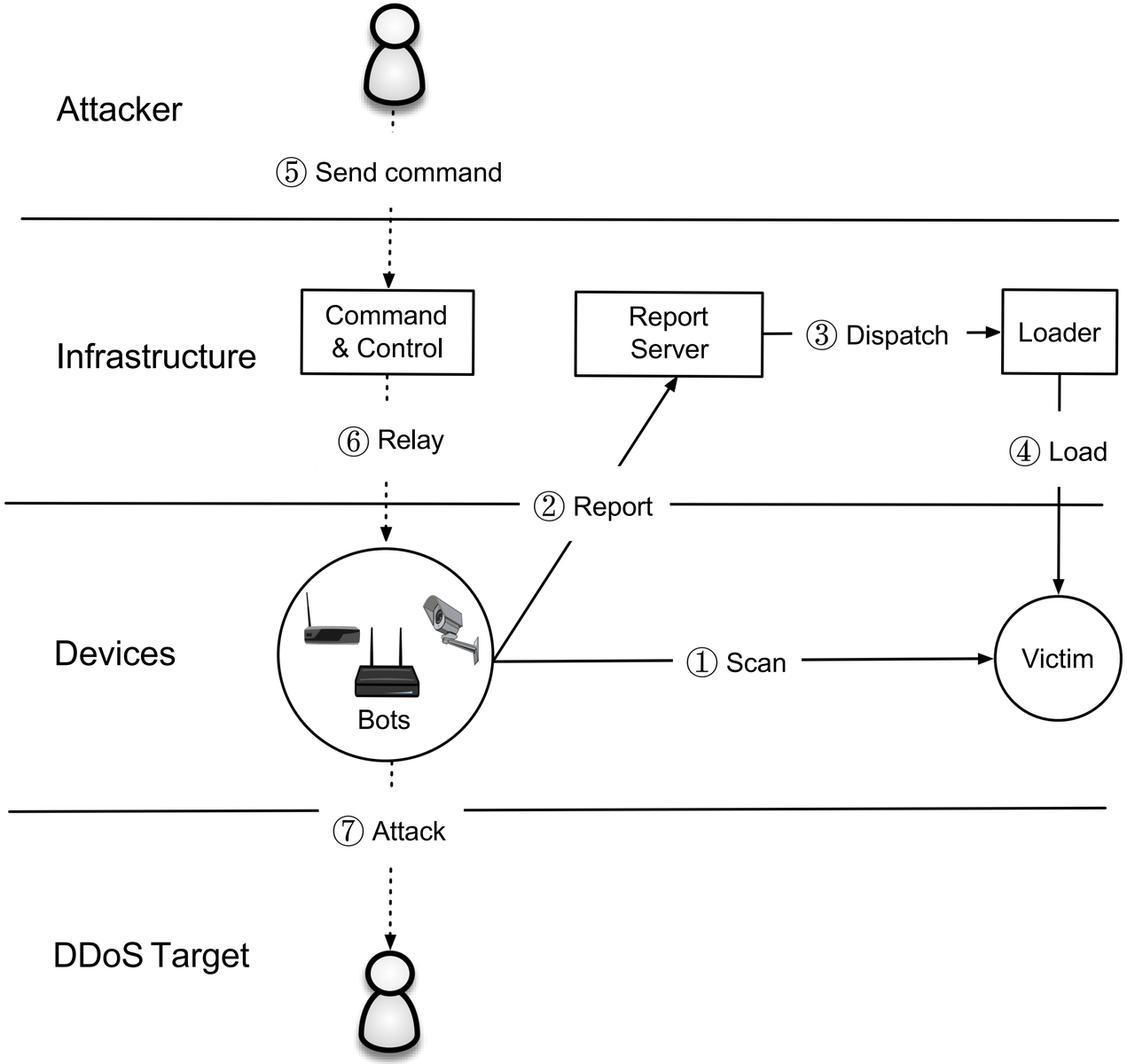
Not Amplification.
Flood with SYN, ACK, UDP, and GRE packets

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

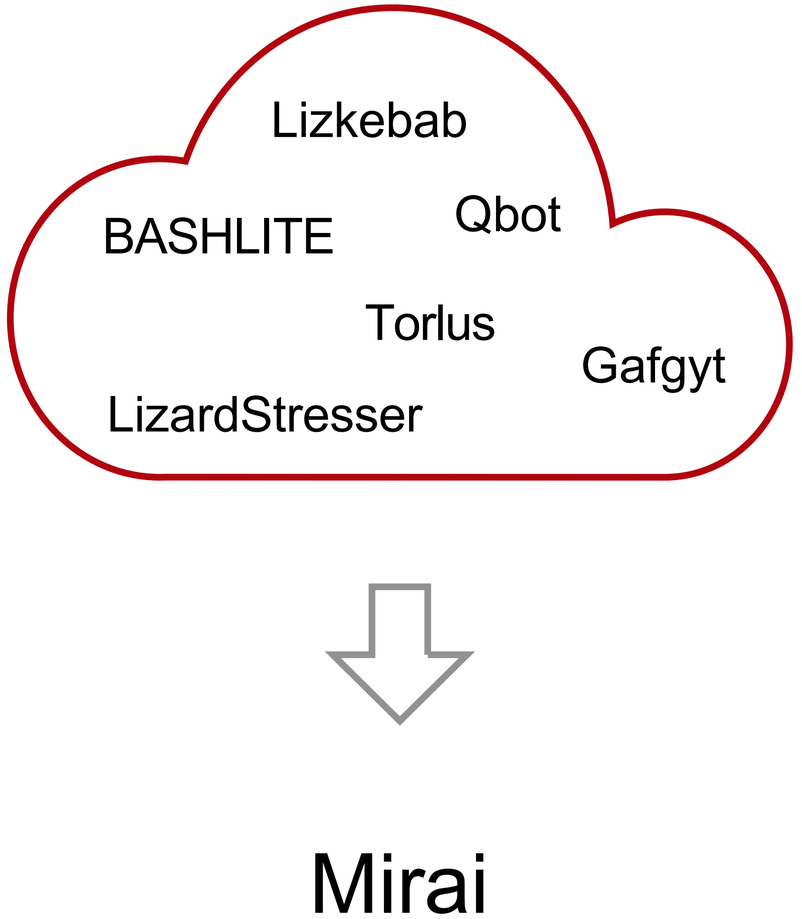


What made Mirai Successful?

The Mirai malware is (astoundingly) badly written. It uses no new or complex techniques.

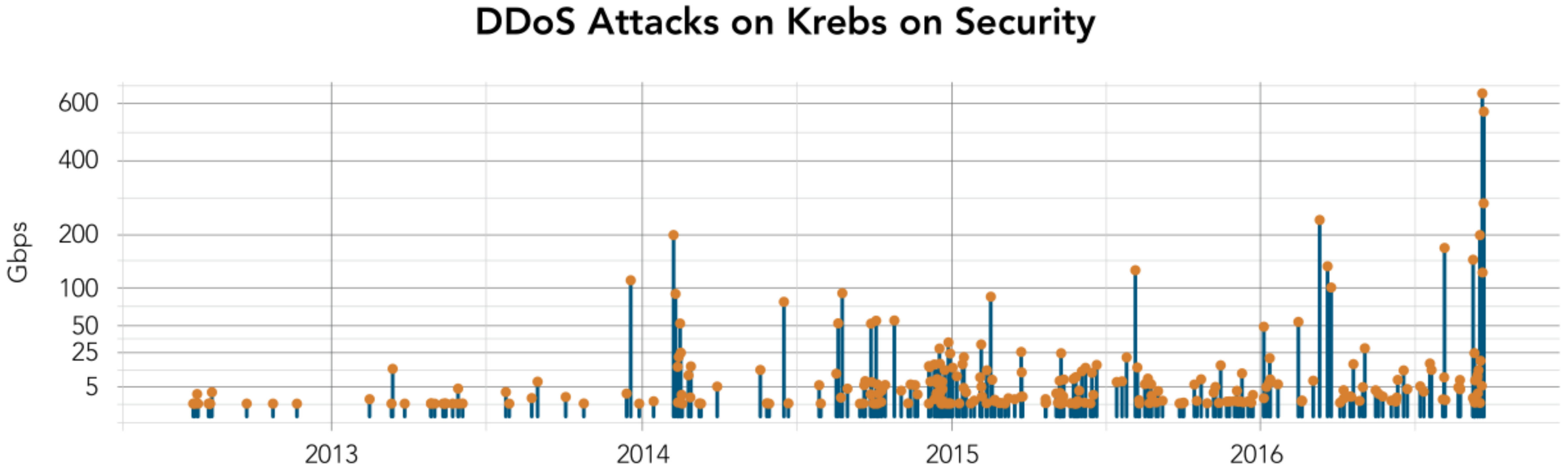
Mirai was successful because:

1. IoT security bar is very low
2. Attack simplicity enabled the malware to compromise heterogeneous hardware
3. Stateless scanning was an improvement over prior versions



Password Guessing

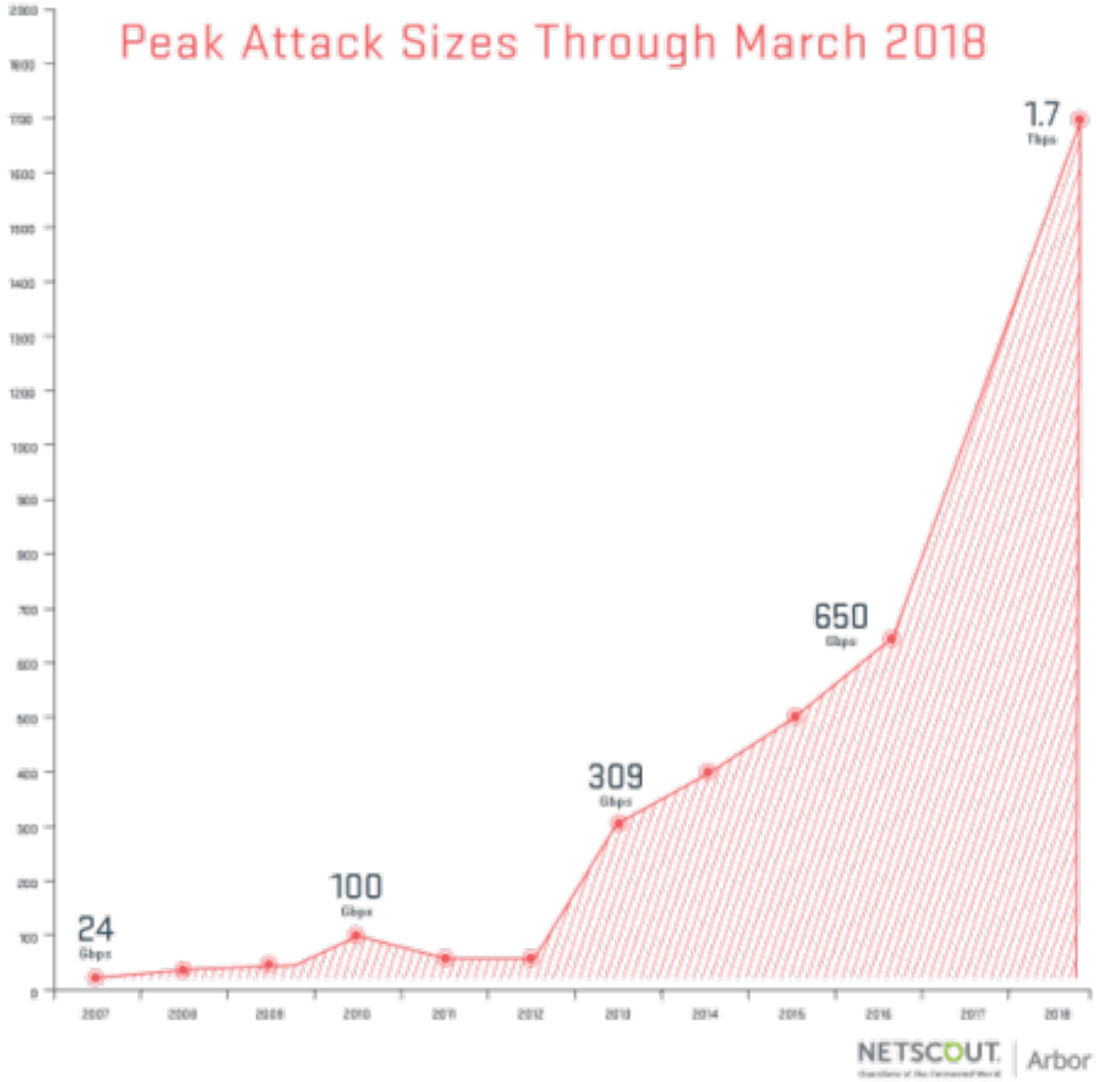
Password	Device Type	Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera	1111	Xerox Printer
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera	Zte521	ZTE Router
pass	Axis IP Camera	admin	IPX-DDK Network Camera	1234	Unknown
888888	Dahua DVR	system	IQinVision Cameras	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				



> “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.”

Source: 2017 Akamai State of the Internet

Memcache

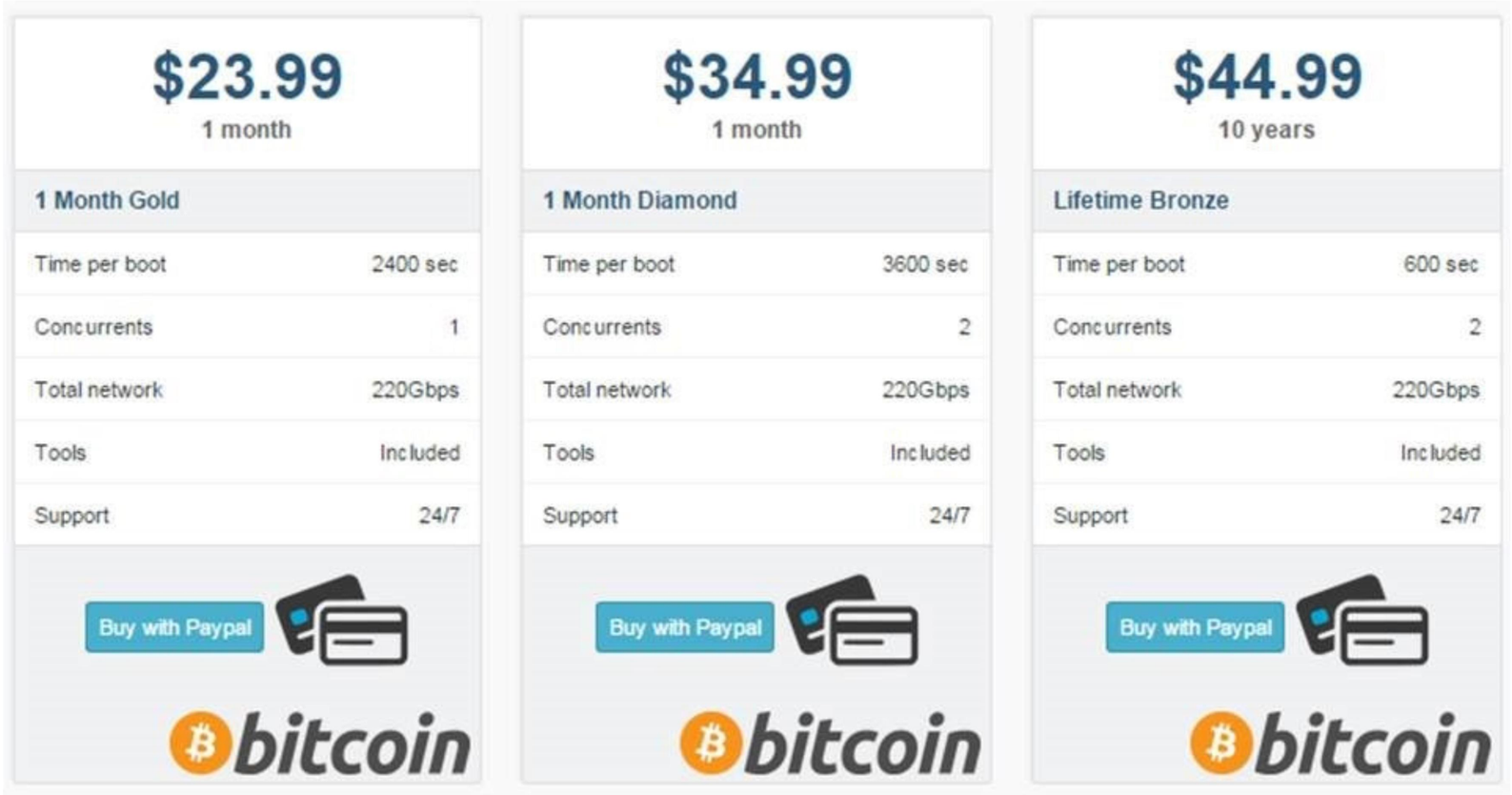


Memcache: retrieve large record

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.

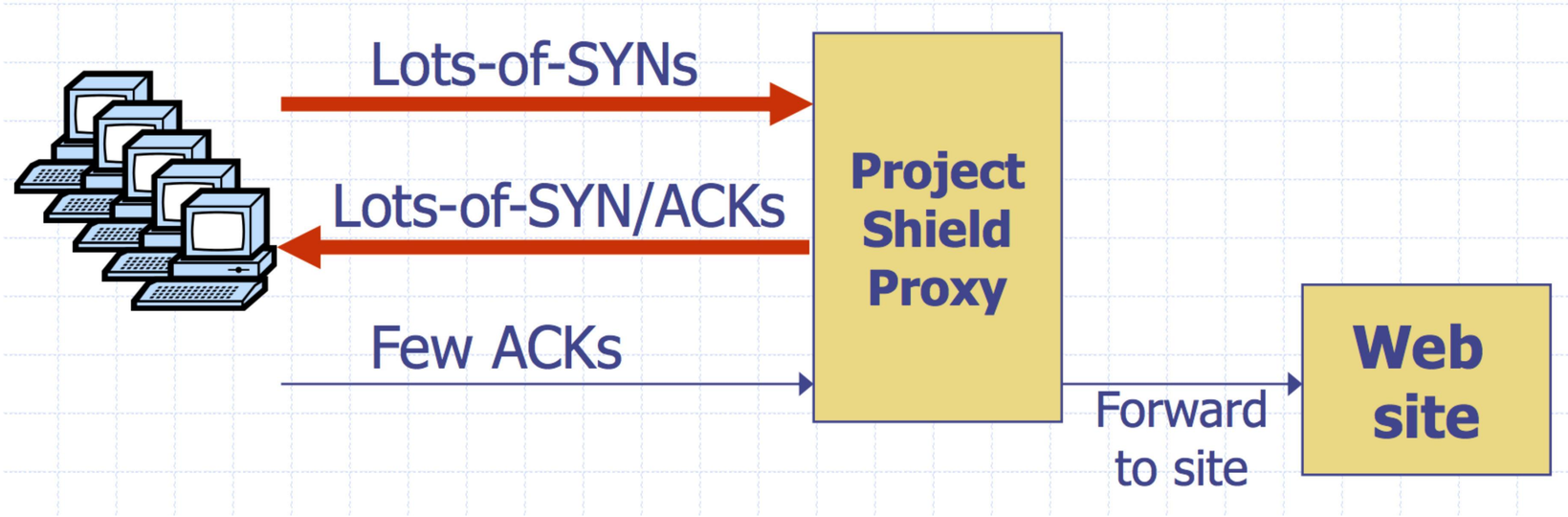
Booter Services



Google Project Shield

DDoS Attacks are often used to censor content. In the case of Mirai, Brian Kreb's blog was under attack.

Google Project shield uses Google bandwidth to shield vulnerable websites (e.g., news, blogs, human rights orgs)



Moving Up Stack: GET Floods

Command bot army to:

* Complete real TCP connection
* Complete TLS Handshake
* GET large image or other content

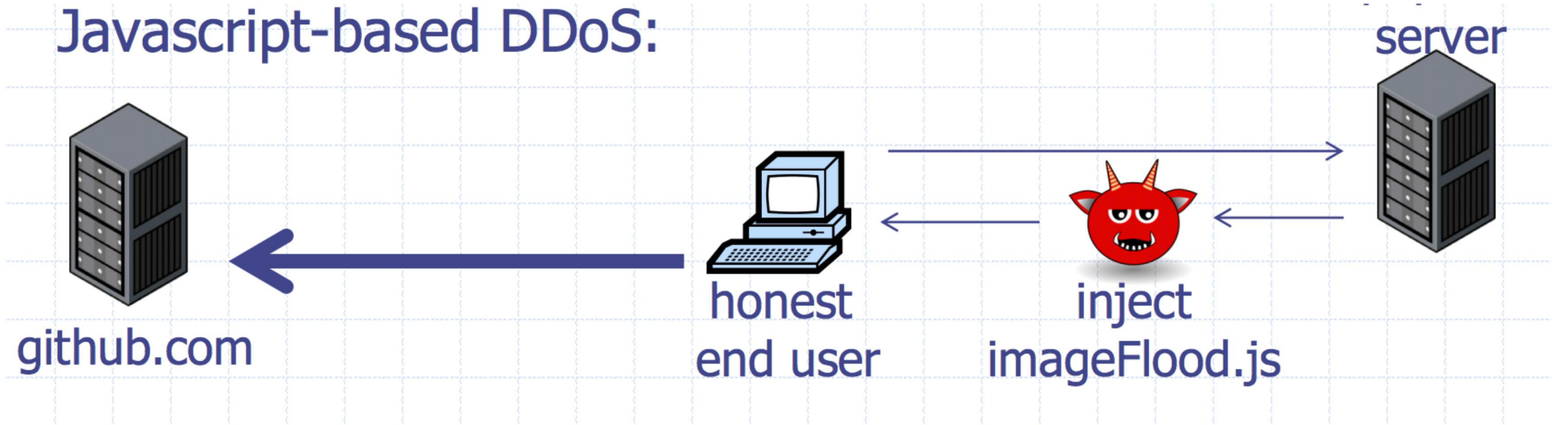
Will bypass flood protections…. but attacker can no longer use random source IPs

Victim site can block or rate limit bots

Github Attacks

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

The Chinese government was widely suspected to be behind the attack



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