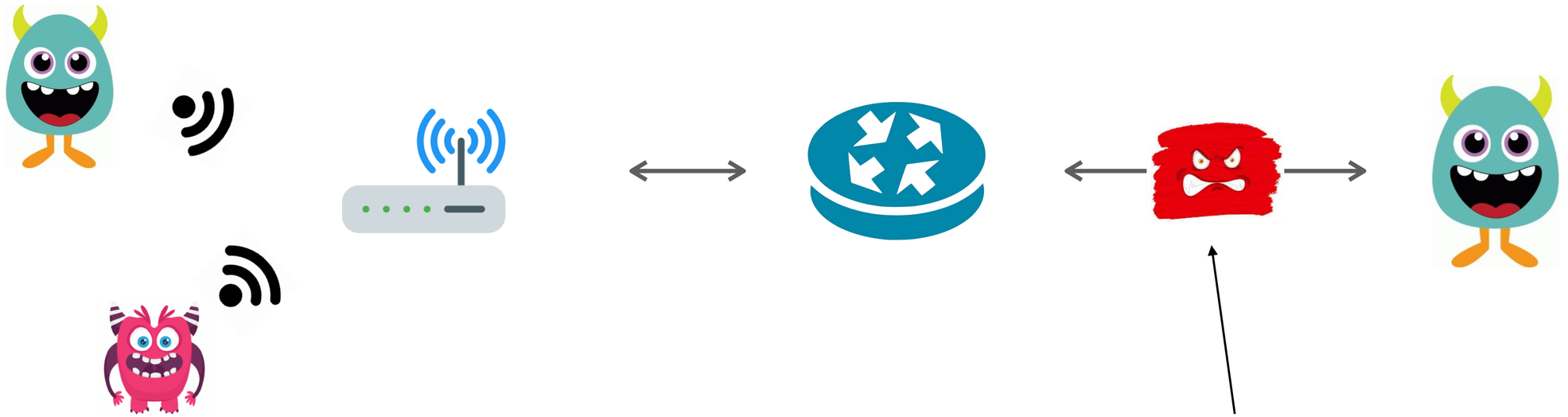
DoS Attacks and Network Defenses

CS155 Computer and Network Security

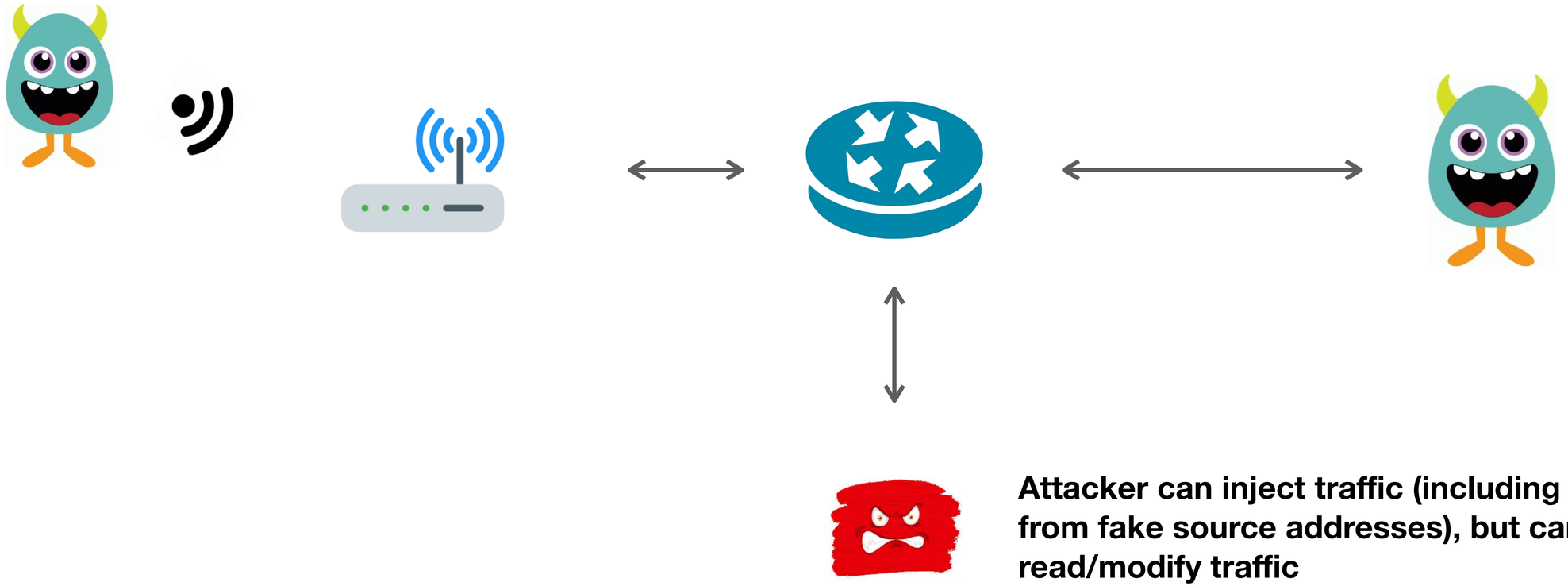
Stanford University

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.

Building a network protocol

Don’t build network proto from scratch

- Never roll your own crypto
- Many opportunities to mess up parsing network packets

gRPC: http2 + TLS 1.3 RPC framework

- Safe parsing in 11 languages
- Exceptionally efficient
- Streaming/Sync/Async
- TLS-based authentication

Or, REST on top of HTTP/2 + TLS 1.3

```
syntax = "proto3";

package calc;

message AddRequest {
  int32 n1 = 1;
  int32 n2 = 2;
}

message AddReply{
  int64 res = 1;
}

service Calculator {
  rpc Add(AddRequest)       returns (AddReply) {}
  rpc Substract(SubRequest) returns (SubReply) {}
  rpc Multiply(MultRequest) returns (MultReply) {}
  rpc Divide(DivideRequest) returns (DivideReply) {}
}
```

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

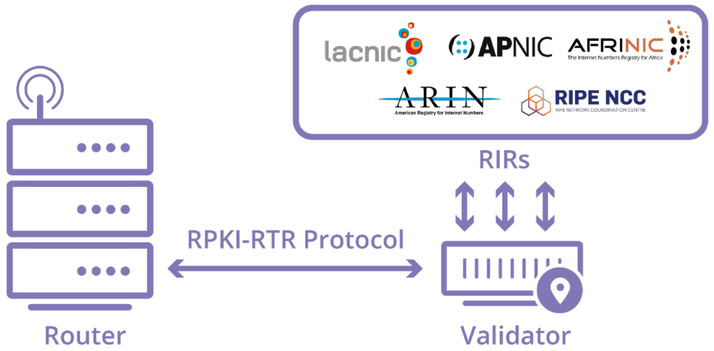
Most people don’t use DNSSEC and never will. Use TLS.

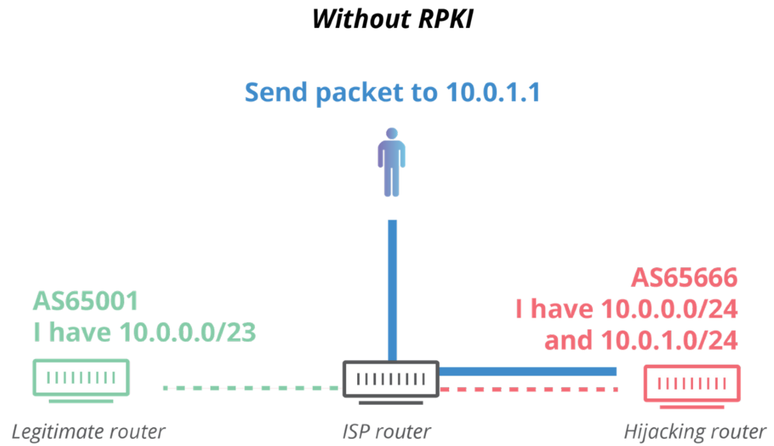
Resource Public Key Infrastructure (RPKI)

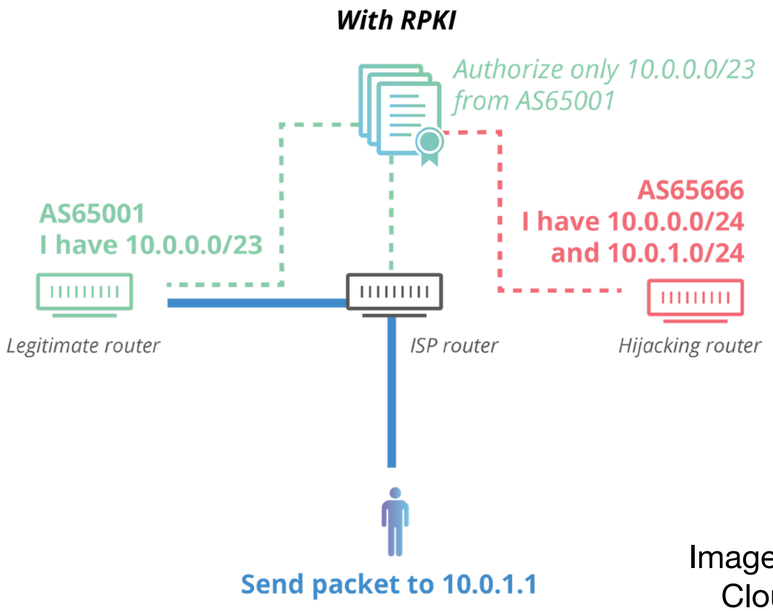
RPKI is a relatively new PKI to help improve BGP security

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

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

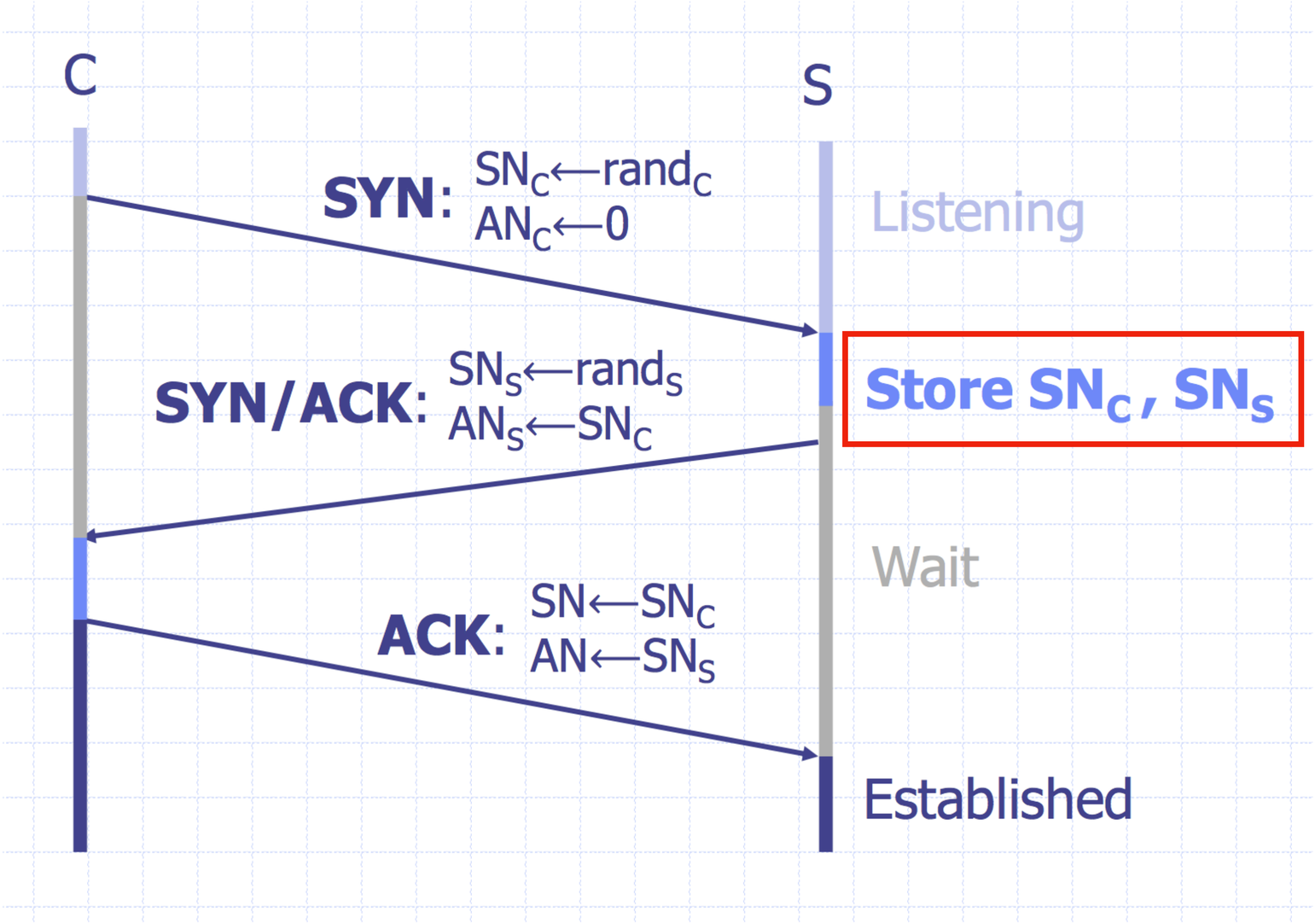
DoS Opportunities at Every Layer

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

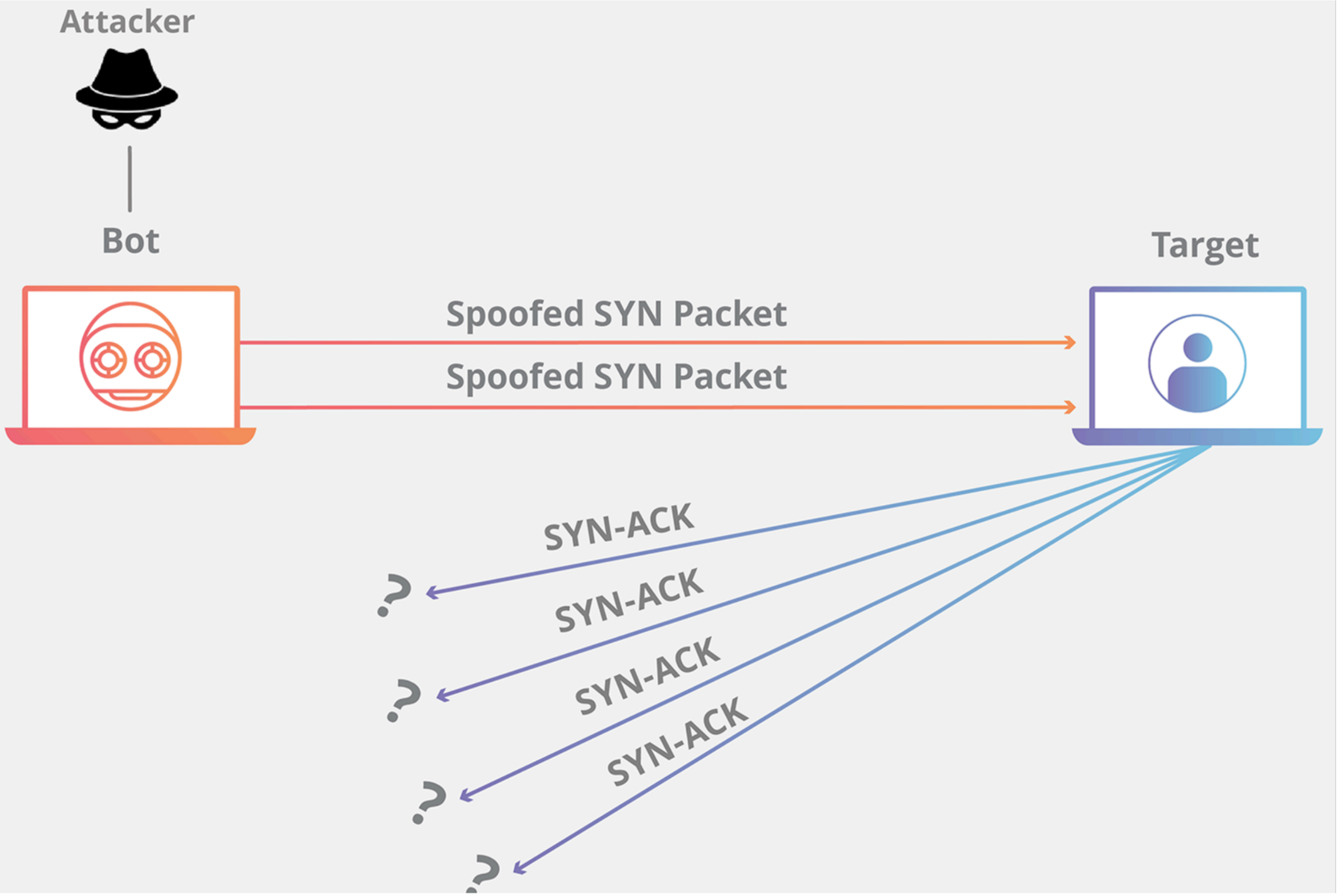
TCP/UDP: require servers to maintain large number of concurrent connections or state

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.

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

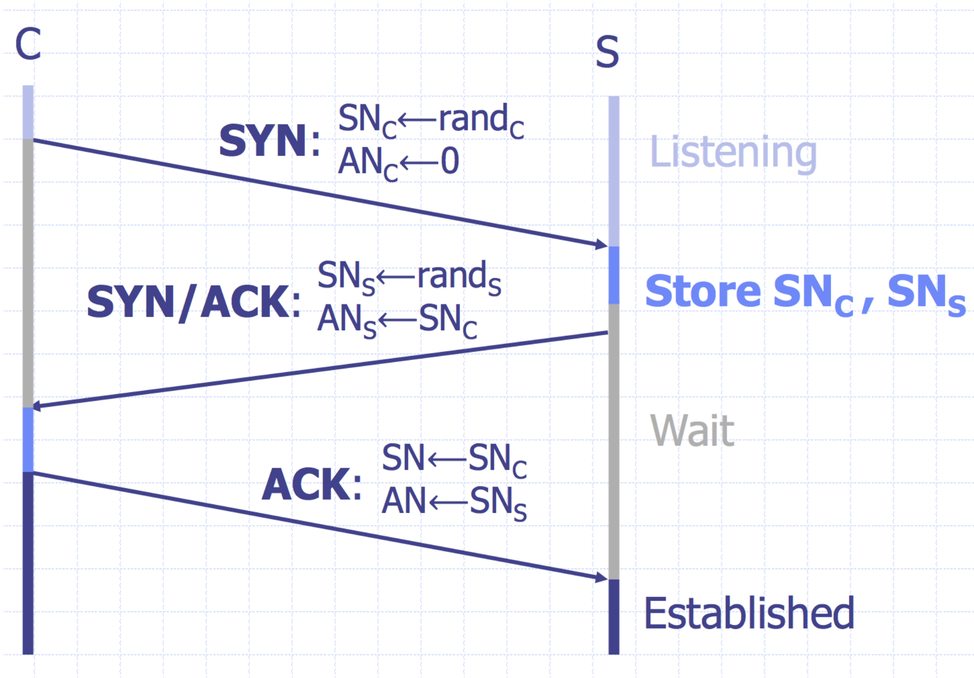
key: picked at random during boot

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



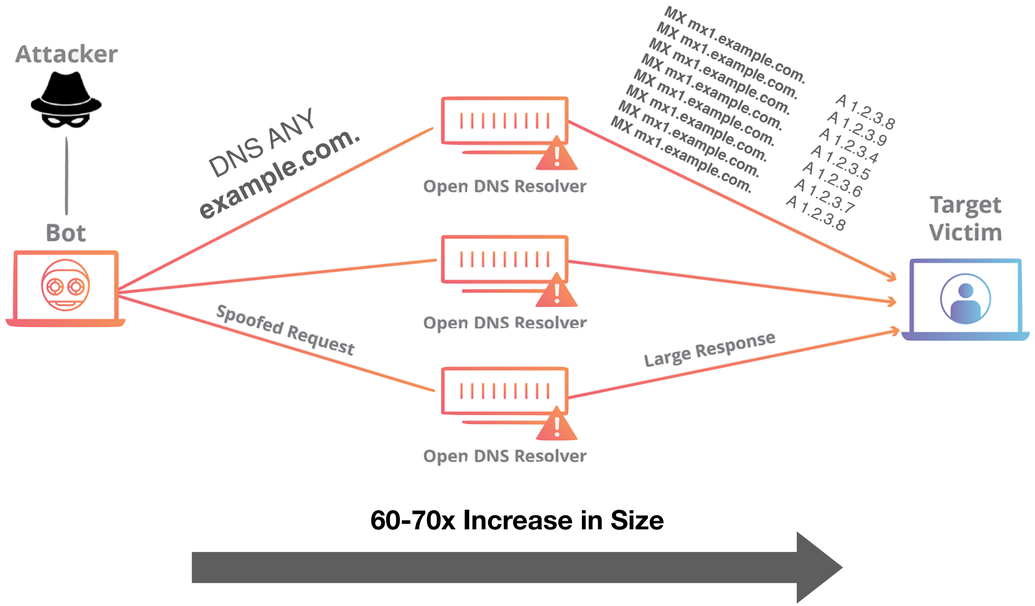
Server does not save state
(loses TCP options)

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

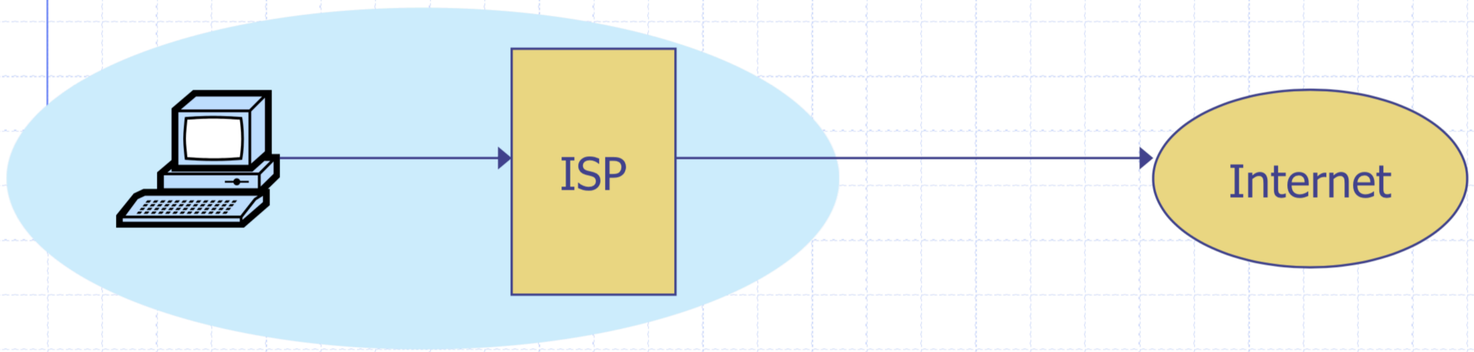
2013: DDoS attack generated 300 Gbps (DNS)

- 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

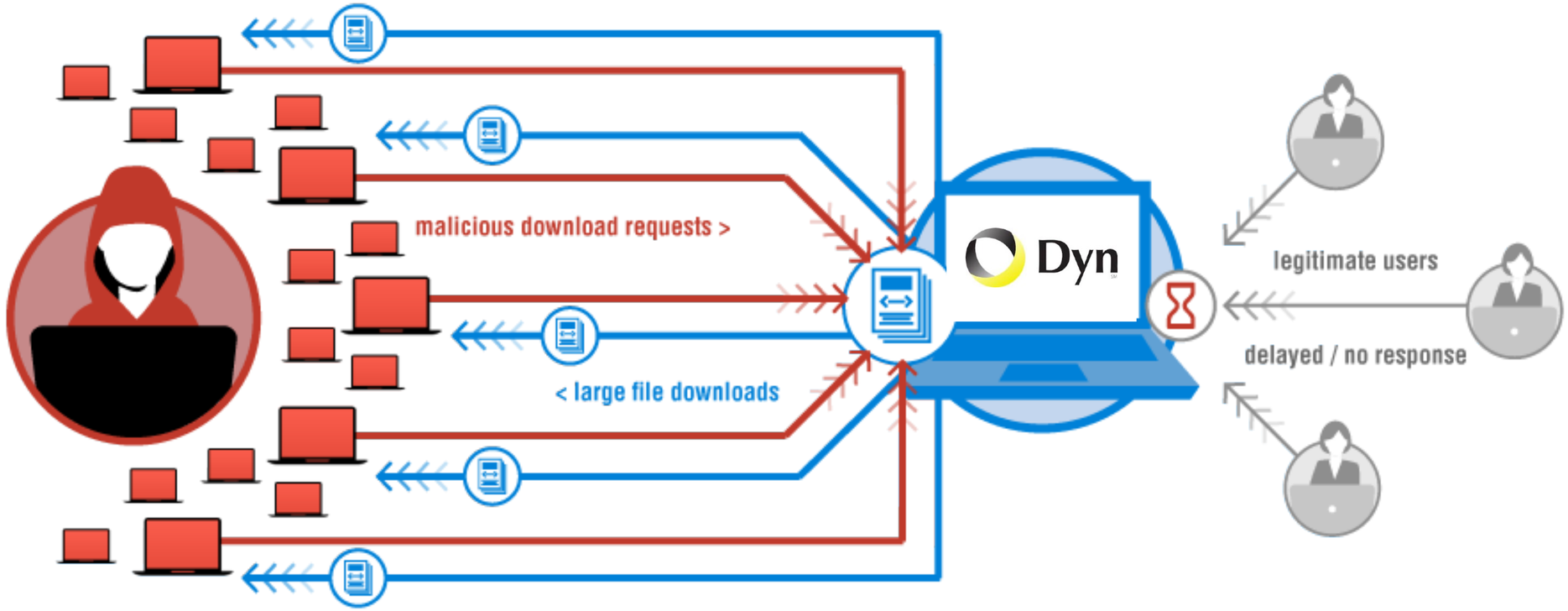
SOUNDCLOUD

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

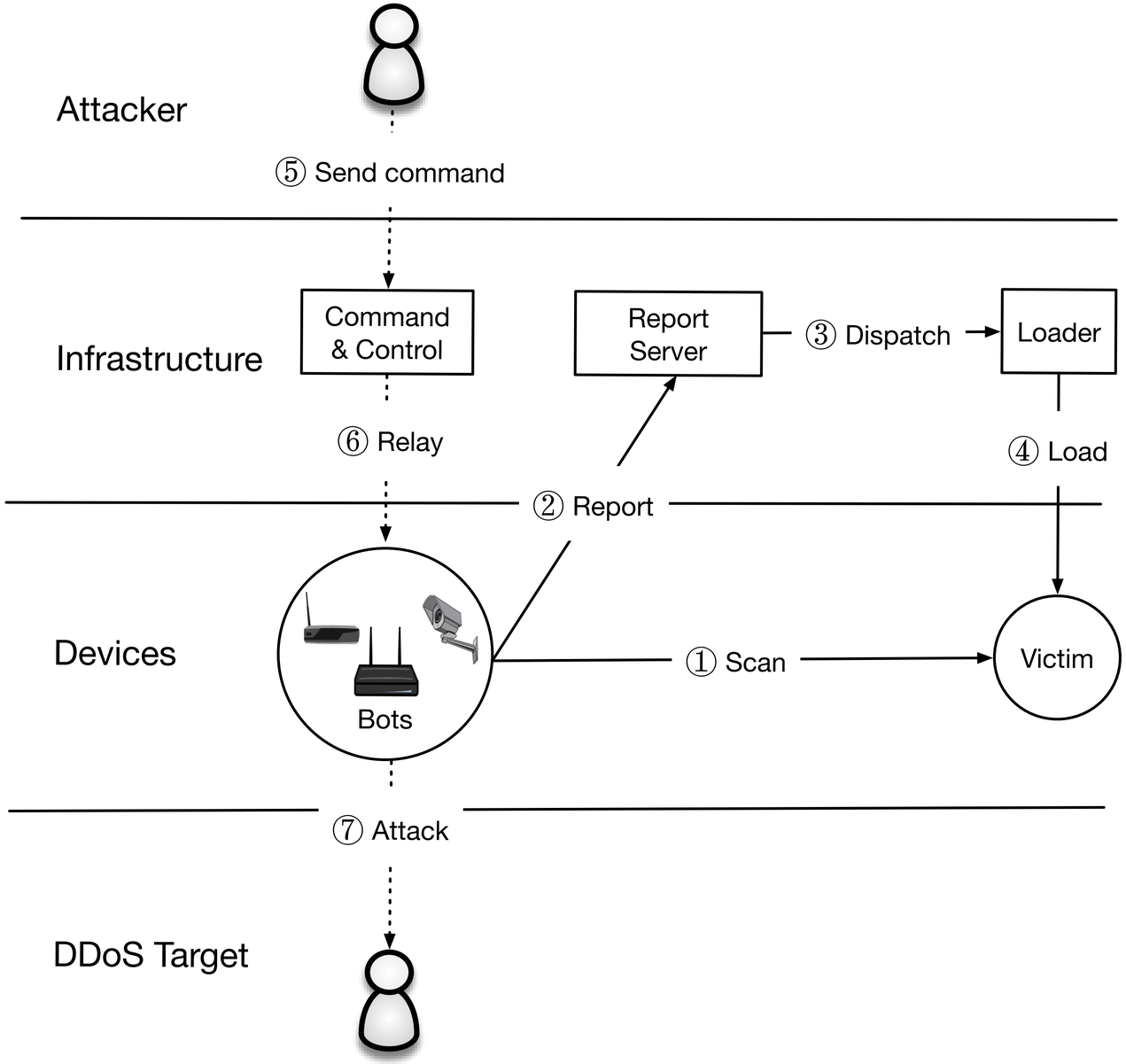
Not Amplification.
Flood with SYN, ACK, UDP, and GRE packets

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

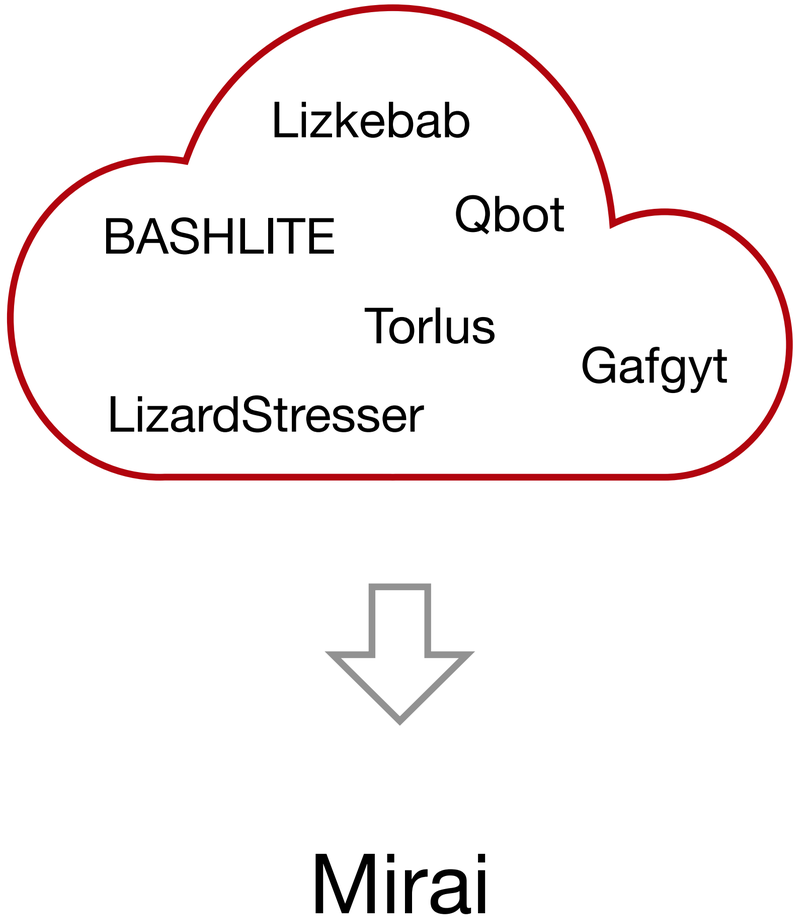


What made Mirai Successful?

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

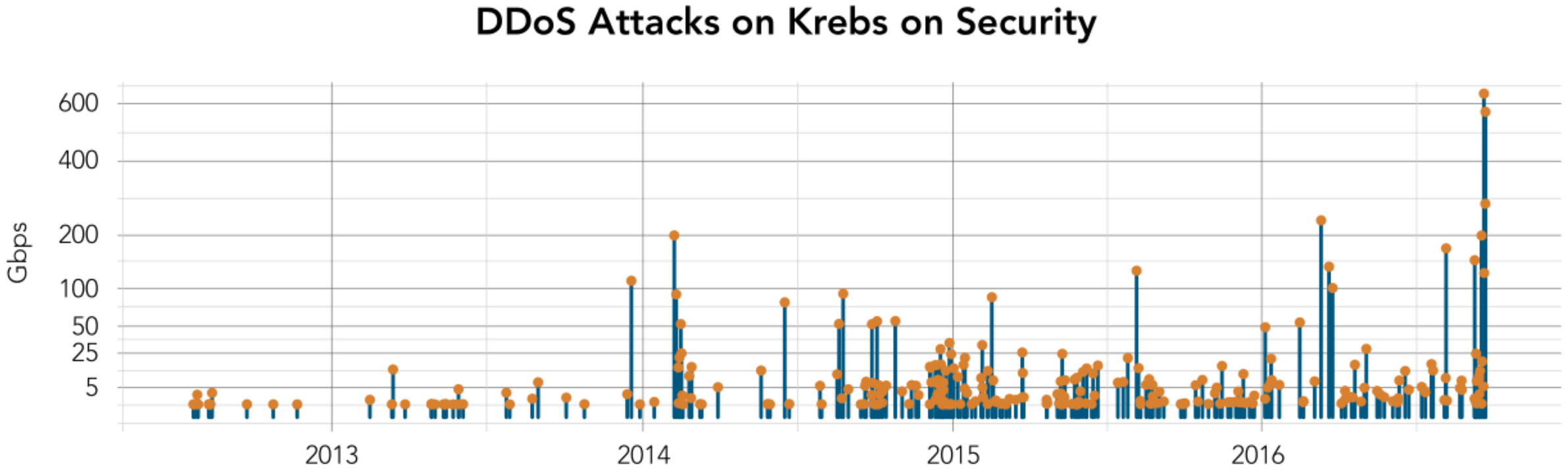
Mirai was successful because:

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

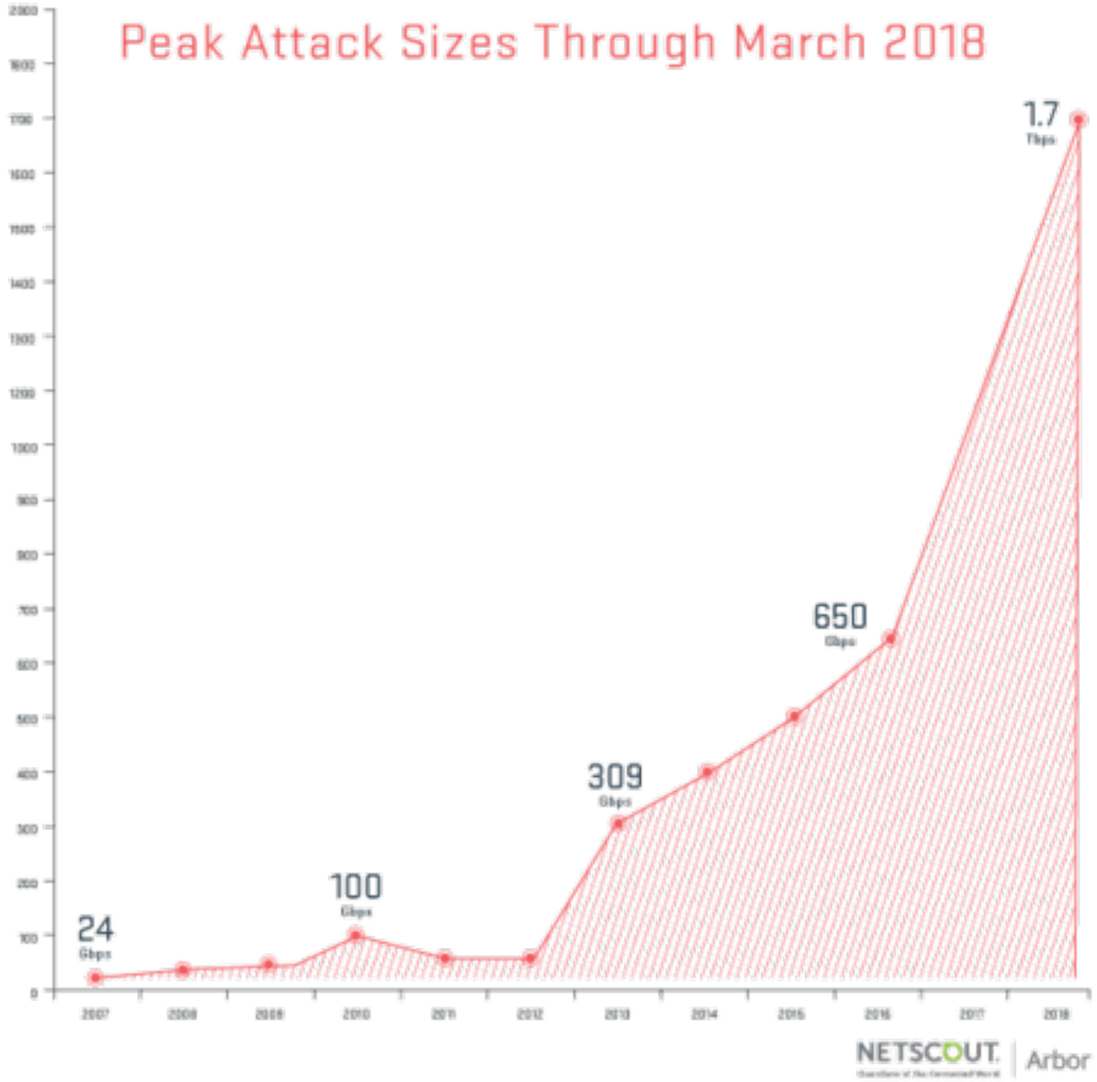
Password	Device Type	Password	Device Type	Password	Device Type
123456	ACTi IP Camera	klv1234	HiSilicon IP Camera	1111	Xerox Printer
anko	ANKO Products DVR	jvbzd	HiSilicon IP Camera	Zte521	ZTE Router
pass	Axis IP Camera	admin	IPX-DDK Network Camera	1234	Unknown
888888	Dahua DVR	system	IQinVision Cameras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

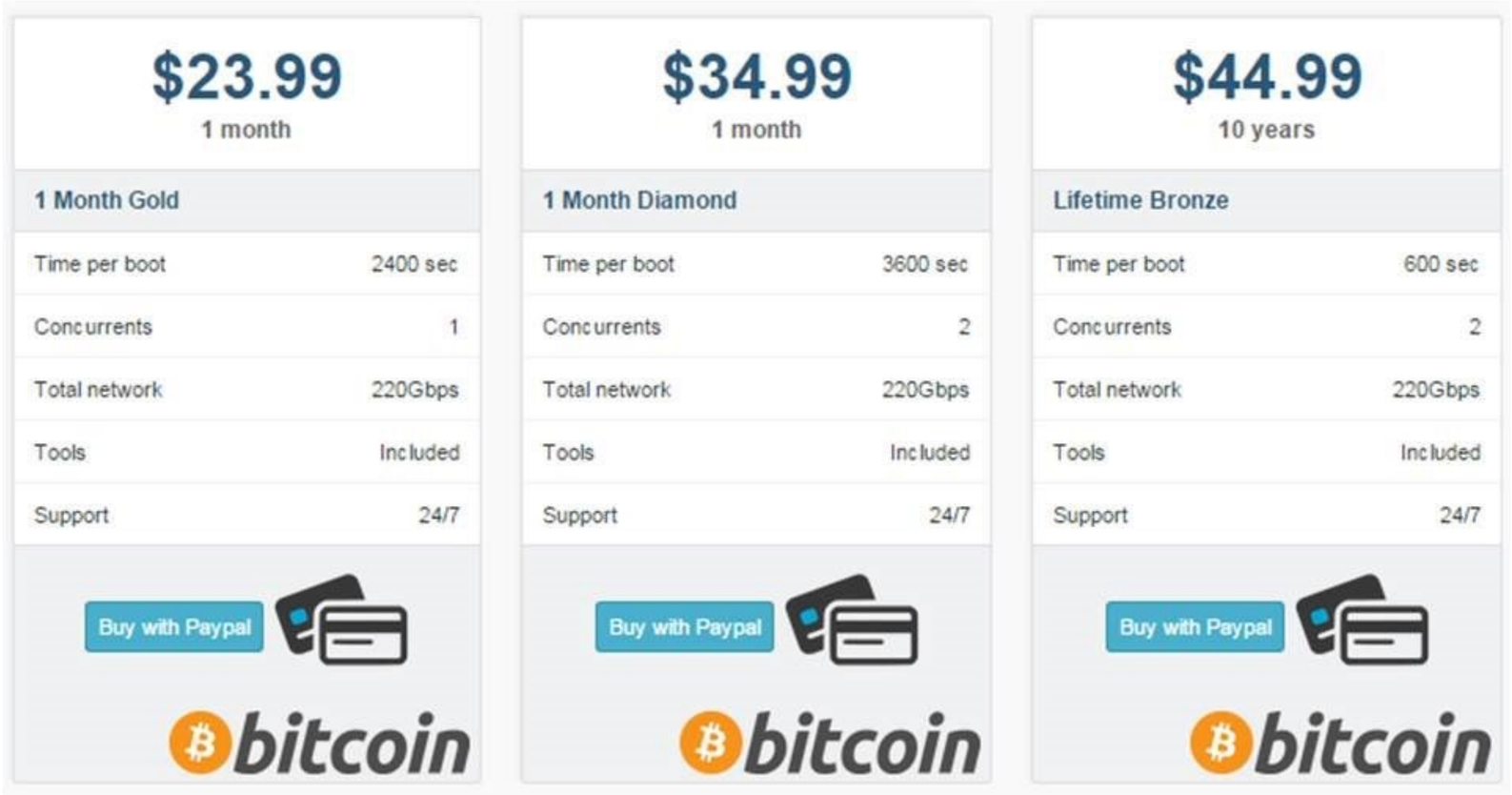


Memcache: retrieve large record

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.

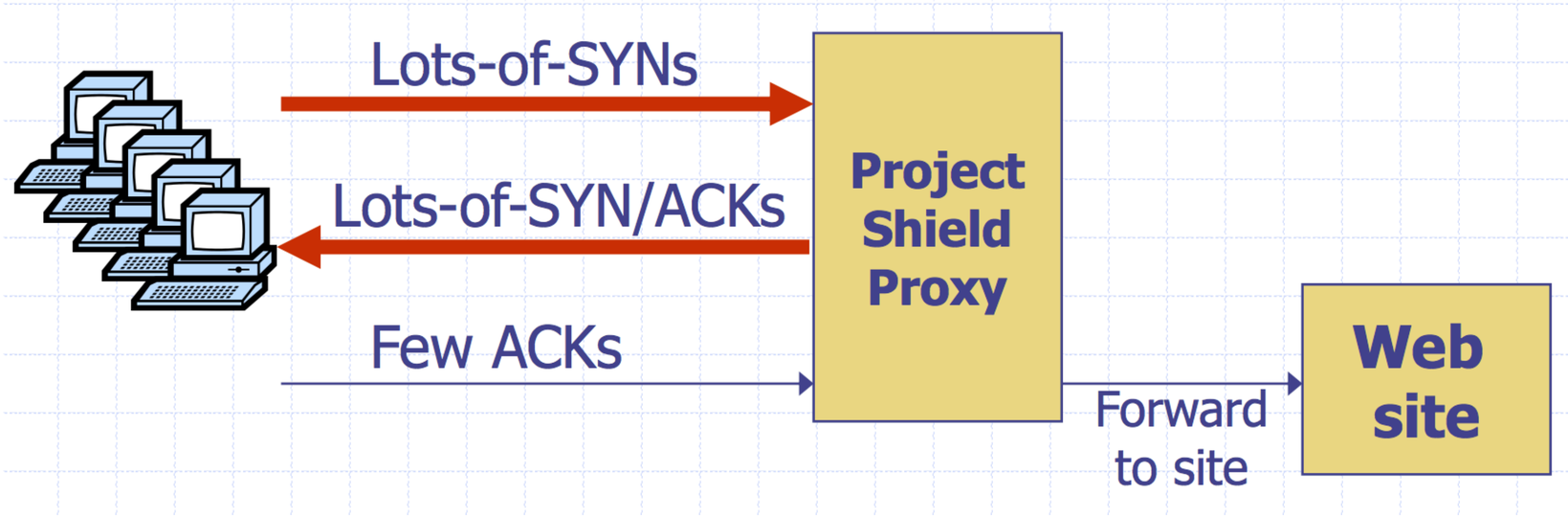
Booter Services



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

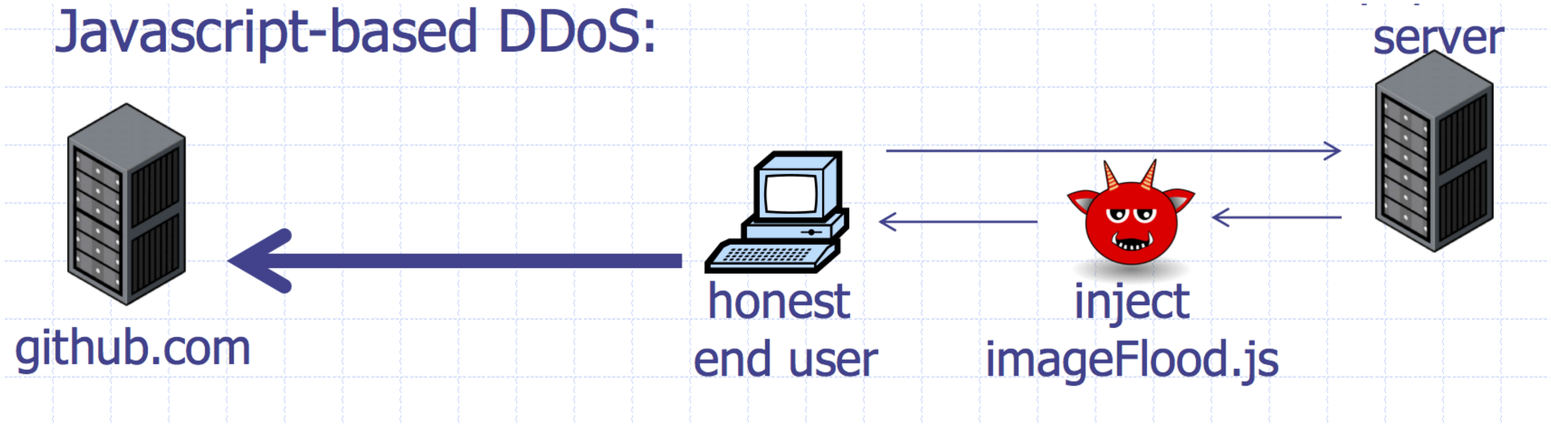
Will bypass flood protections…. but attacker can no longer use random source IPs

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 always use HTTPS!

Network Defenses

Local Network Services

Review: Popular TCP and UDP services live on standardized ports. HTTPS servers listen on TCP/443. SSH on TCP/22.

Some services you don’t want listening on the public Internet.

Recursive DNS Resolvers: allows attackers to mount DDoS attacks

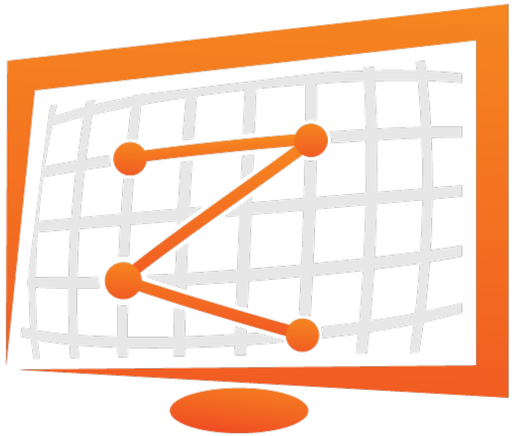
Windows File Sharing: historically full of vulnerabilities. What if a local machine doesn’t have a secure password on it?

Port Scanning

Send a SYN or application-specific UDP packet to a port to see if any service is listening

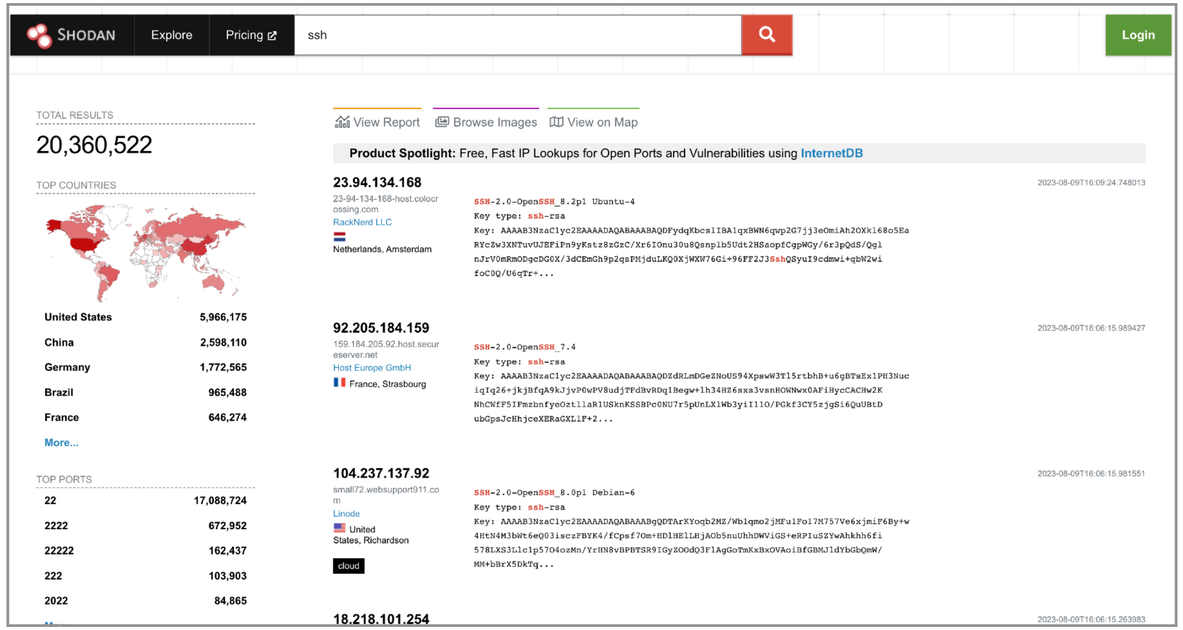
Vertical Scan: Try large number of ports on a single host. Typically use Nmap.

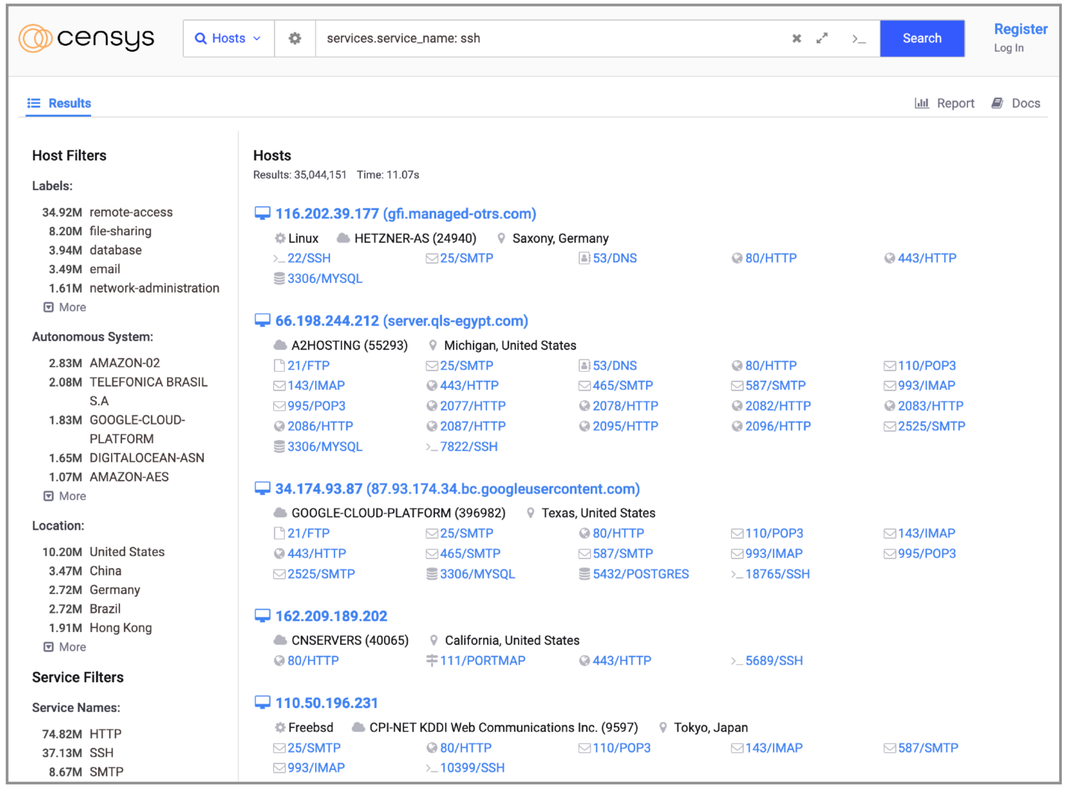
Horizontal Scan: Try a single port on a large number of hosts. Typically ZMap.



Service Search Engines

Public services like Shodan and Censys index all of the publicly available services on the Internet





Attacks Against Internet Services

MOVEit is a piece of software that allows file transfer between organizations

Vulnerable to multiple login-field SQL injection vulnerabilities

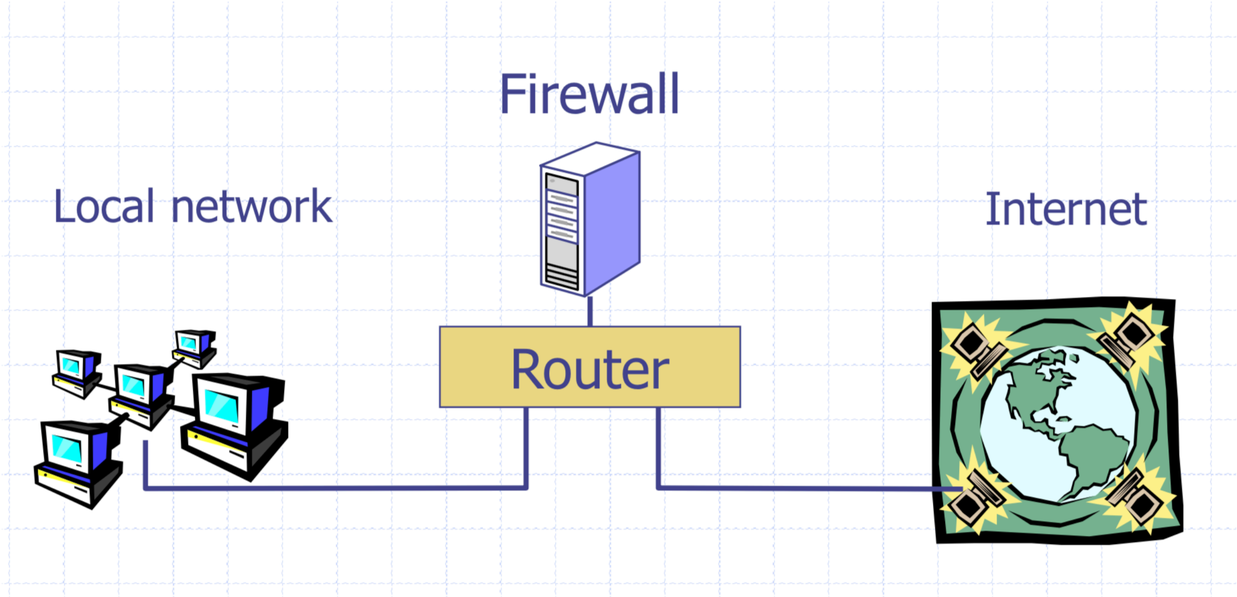
Ransomware’d/Extorted Companies based on the data on their Internet MOVEit Servers

Progress®
MOVEit®

Firewalls

Separate local area network (LAN) from the Internet. Only allow some traffic to transit.

Sometimes rules on a router. Sometimes a standalone device.



Basic Packet Filtering

Uses transport and IP layer information only

- IP Source Address, Destination Address
- Protocol (TCP, UDP, ICMP, etc.)
- TCP and UDP source and destination ports

Examples:

- “Do not allow external hosts to connect to Windows File Sharing”
  -> DROP ALL INBOUND PACKETS TO TCP PORT 445

What's the rule?

What if you have a network with lots of servers but only want outsiders to be able to access a web server?

DROP ALL INBOUND PACKETS IF DEST PORT != 80

All outbound connections also have a source port! Their responses will blocked!

IANA Port Numbering

System or Well-Known Ports [1,1023]:

Common services, e.g., HTTP -> 80, SSH -> 22

User or registered ports [1024, 49151]

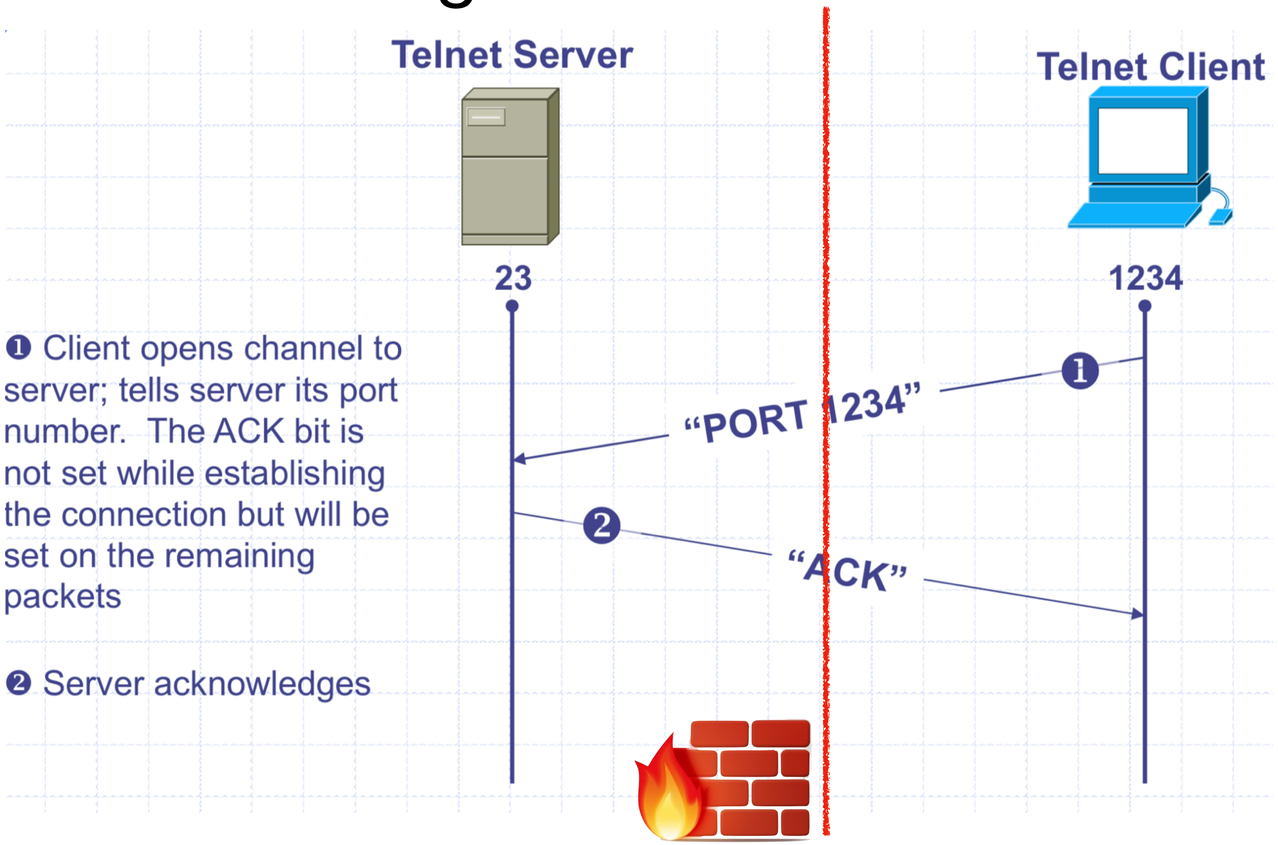
Less well-known services

Ephemeral/Dynamic/Private Ports [49152, 65535]

Short lived connections

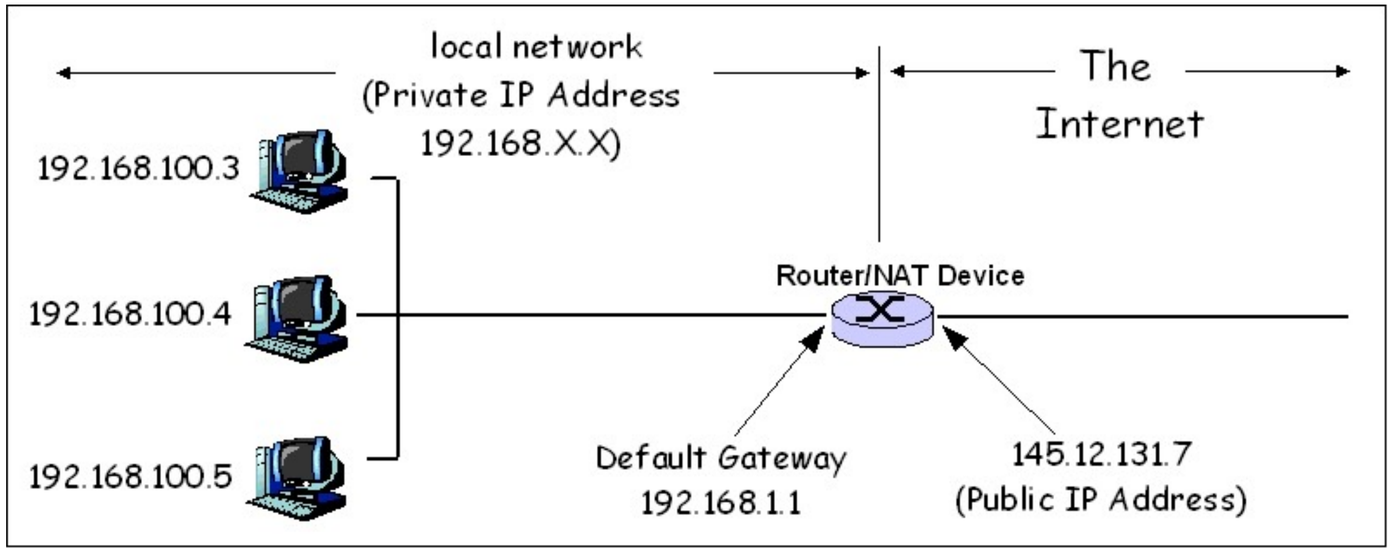
Stateful Filtering

Firewall tracks outgoing connections and allows associated inbound traffic back through



Network Address Translation (NAT)

NATs map between two different address spaces. Most home routers are NATs and firewalls.



Private Subnets

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Local vs. Network Firewall

Firewalls we've discussed so far have all been network firewalls. Most have lived at the edge of the organization.

Firewalls also run on individual hosts. Linux servers use **iptables**.

Typically have a combination of network and host firewalls

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

Local vs. Network Firewall

Organizations typically have a combination of network and host firewalls

- Border (Network) Firewall will block malicious traffic from the outside and limit inbound traffic to accessing only servers intended to be accessed by the public
- Host Firewalls protect hosts from other hosts (e.g., protect against internal compromise and malicious insiders)

Think of firewall rules in terms of “Defense in Depth"

Next Generation Firewalls (NGFW)

So far, firewalls operate by allowing access to a specific host or protocol — but what about malicious application traffic?

Next Generation Firewalls (Industry term for Application-Layer firewall)s protect for attacks *within* L7 traffic

For Example:

- Virus scanning for SMTP
  - Need to understand protocol, MIME encoding, ZIP files, etc
- Look for SQL injection attacks in HTTP POSTs
- Look for a large number of authentication attempts or malformed requests

Intrusion Detection Systems (IDS)

Software/device to monitor network traffic for attacks or policy violations

Violations are reported to a central security information and event management (SIEM) system where analysts can later investigate

Signature Detection: maintains long list of traffic patterns (rules) associated with attacks

Anomaly Detection: attempts to learn normal behavior and report deviations

Open Source IDS

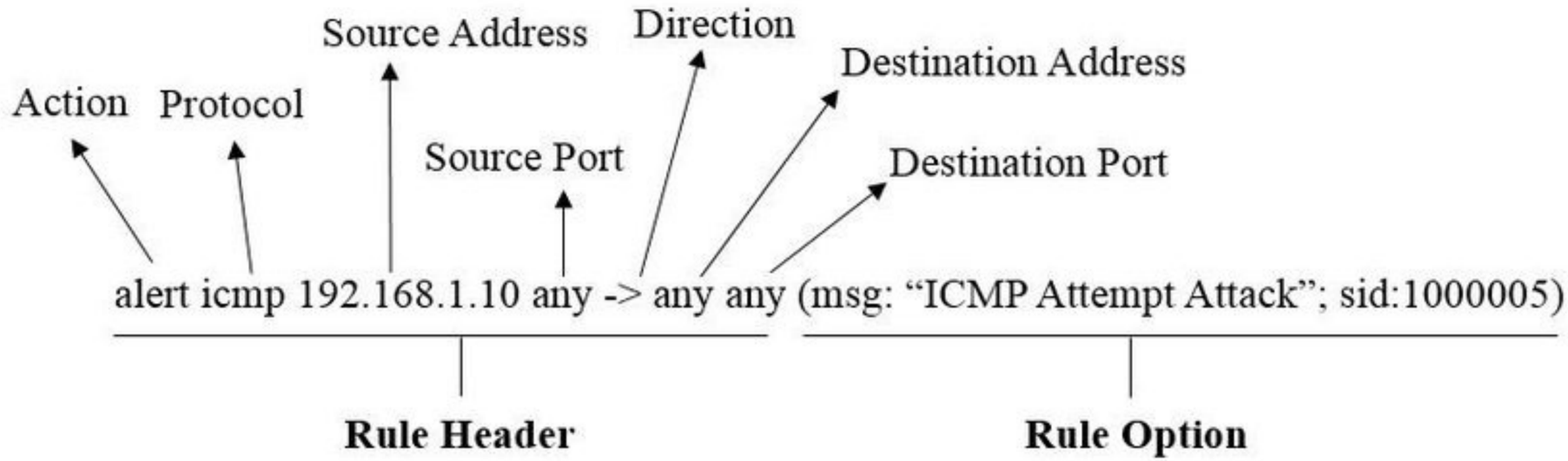
Three Major Open Source IDS (and a *tremendous* number of commercial products)

Snort

~~Bro~~ Zeek

Suricata

Example Snort Rule



Outbound Too!

Organizations will often inspect outbound traffic as well

- Block access to sites with known malicious behavior
- Prevent exfiltrating data
- Block services like bit torrent

Be careful on enterprise networks! Sometimes companies will even install their own root certificates on employee workstations to monitor TLS traffic.

Remote Access

Virtual Private Networks (VPNs)

Problem: How do you provide secure communication for non-TLS protocols across the public Internet?

VPNs create a fake shared network on which traffic is encrypted

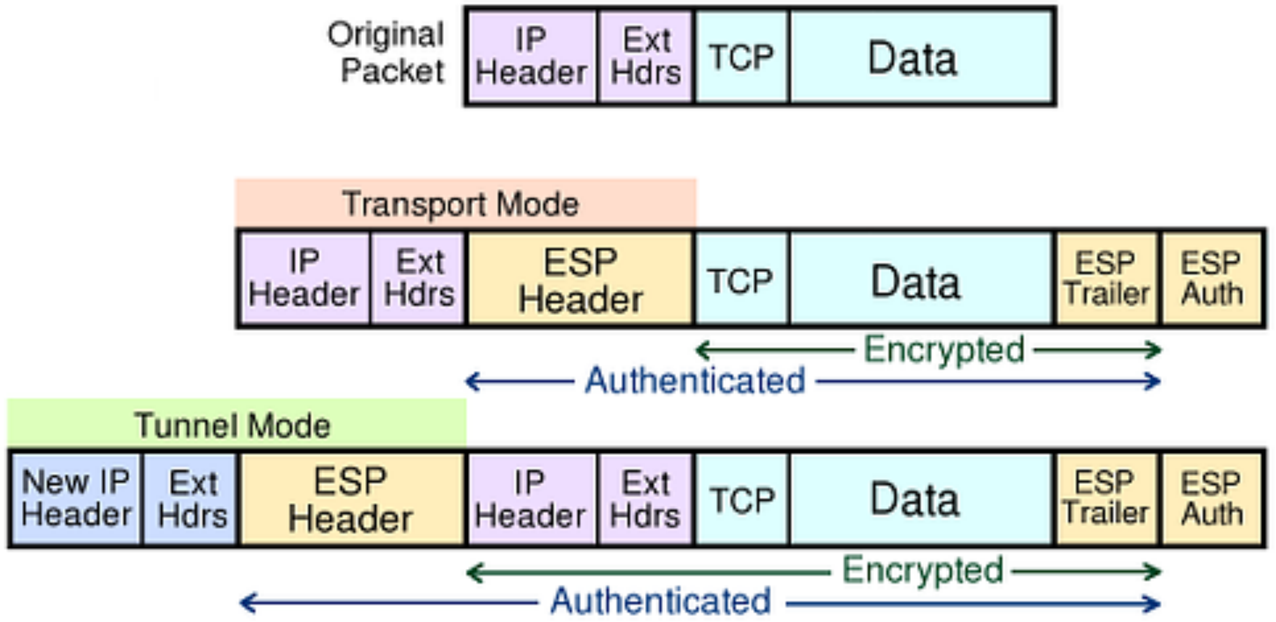
Two Broad Types:

- Remote client (e.g., traveler with laptop) to corporate network
- Connect two remote networks across Internet

IPSec

Several VPN protocols exist (PPTP, L2TP, IPsec, OpenVPN)

Most popular is IPsec. OpenVPN is open source.



Wireguard

Recently introduced VPN that has gained significant following in the past 5 years over options like OpenVPN:

- Simpler protocol and much more performant than OpenVPN. Relatively few configuration options reduces opportunity for error
- Utilizes modern cryptographic primitives like Noise protocol framework, Curve25519, ChaCha20, Poly1305,

Cisco AnyConnect

Stanford and many other organizations use Cisco AnyConnect

Encapsulates traffic in TLS! Initial handshake uses normal TCP-based TLS for initial handshake and then DTLS (UDP-based TLS) to transport data

Gooey Middle

VPNs support the idea of having a secure internal network and untrusted public Internet. Unfortunately, attacker has a ton of access once the network perimeter is breached.

Unfortunately, internal networks aren't *that* secure. Computers are compromised all the time and attackers have free rein.

Zero Trust Security (BeyondCorp)

Google: assume internal network is *also* out to get you. Remove privileged intranet and put all corporate applications on the Internet.

Access depends solely on device and user credentials, regardless of a user's network location

Protect applications, not the network