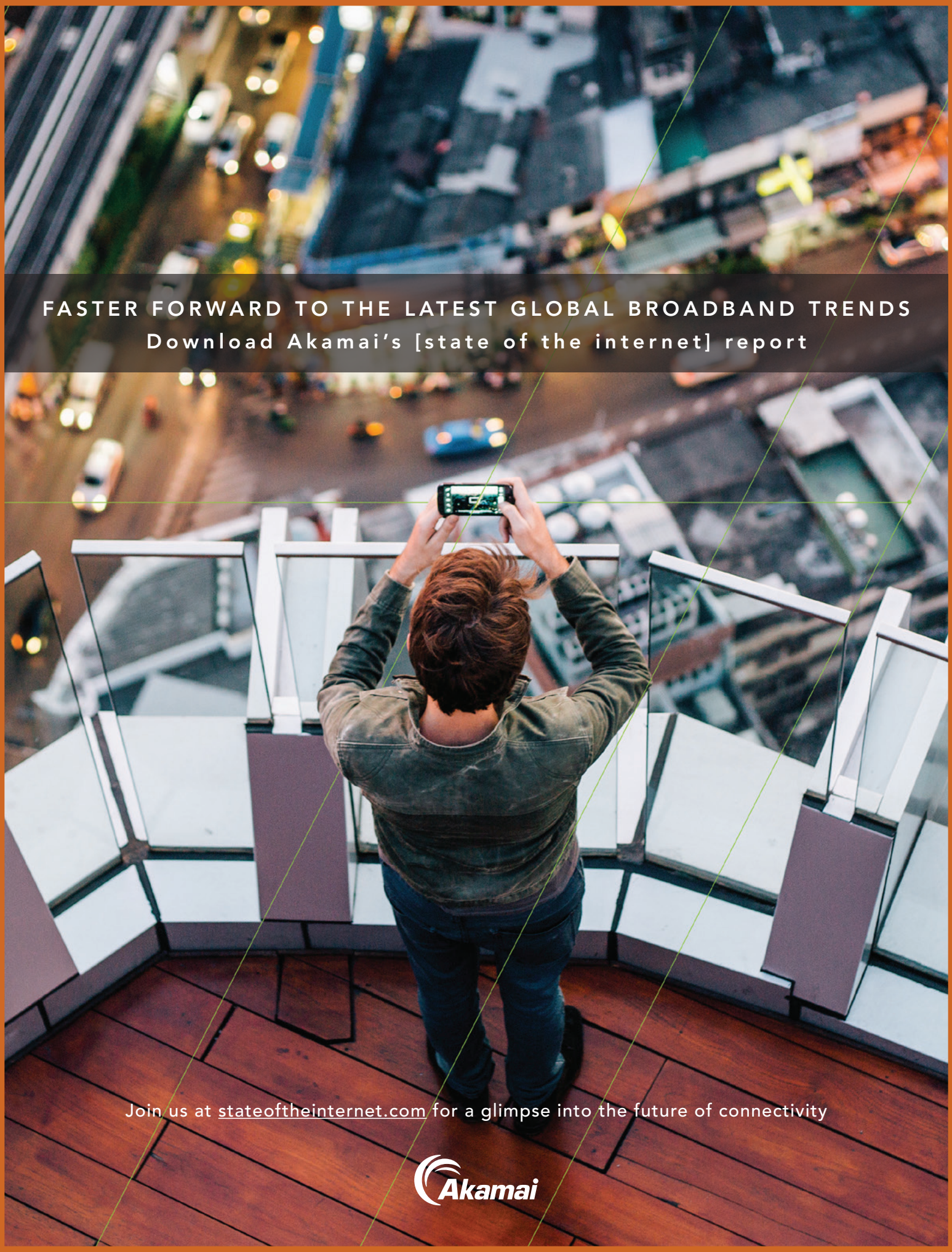
Q1 [2015 Report] Volume 2 Number 1

akamai's [state of the internet] / security


FASTER FORWARD TO THE LATEST GLOBAL BROADBAND TRENDS
Download Akamai's [state of the internet] report
Join us at stateoftheinternet.com for a glimpse into the future of connectivity
Akamai

The Q1 2015 State of the Internet — Security Report marks a significant change from past editions. With this edition, we've combined DDoS attack data previously published in the classic State of the Internet Report with the data previously published in the quarterly Prolexic DDoS Attack Report. The two data sources help form a more holistic view of the Internet and the attacks that occur on a daily basis.

In February 2014, Akamai acquired Prolexic, bringing a powerful influx of security research talent to the table. With the publication of the Q3 2014 report, the Prolexic DDoS Attack Report officially became the State of the Internet — Security Report. With the Q1 2015 edition, we've arrived at an important step in the journey, including contributions from the Computer Security Incident Response Team (CSIRT) and Threat Research, as well as the Prolexic Security Engineering and Research Team (PLXsert).

Readers of the classic State of the Internet Report were accustomed to a report that included a section for security, with contributions from CSIRT and CSI. With this report, the focus is all security, all the time. While DDoS attack information will continue to be the focus of the report, each quarter will include special studies and highlights of attack trends.

The narrative that follows is based on data produced from Kona Web Application Firewall (WAF) and Prolexic DDoS protection technology deployed around the world. Each technology collects a distinct data set that represents a unique view of the Internet. It allows Akamai to compare and contrast different indicators of attack activity.

We explore which industries among our customer base suffered the highest volume of attacks, what the most popular attack techniques and vectors were, and where the attack traffic came from.

It's the most extensive picture we've constructed yet, and hope you find it valuable.

As always, if you have comments, questions, or suggestions regarding the State of the Internet Report, the website, or the mobile applications, connect with us via e-mail at stateoftheinternet-security@akamai.com or on Twitter at @State_Internet.

You can also interact with us in the State of the Internet subspace on the Akamai Community at https://community.akamai.com.

Akamai Technologies

[SECTION][1] ANALYSIS + EMERGING TRENDS

The first quarter of 2015 set a record for the number of DDoS attacks recorded on the Akamai's PLXrouted network – more than double what was reported in Q1 2014. The profile of the typical attack, however, has changed. Last year, high bandwidth, short duration attacks were the norm. In Q1 2015, the typical DDoS attack was less than 10 Gbps and persisted for more than 24 hours.

The most common DDoS attack vectors have shifted as well. This quarter, Simple Service Discovery Protocol (SSDP) attacks made up more than 20 percent of the attack vectors, while SSDP attacks were not observed at all in Q1 2014. The proliferation of unsecured home-based, Internet-connected devices using the Universal Plug and Play (UPnP) Protocol has made them attractive for use as reflectors.

During Q1 2015, the gaming sector was hit with more DDoS attacks than any other industry. Gaming has remained the most targeted industry since Q2 2014 and its share of DDoS attacks has remained steady at 35 percent when compared to the previous quarter.

As was the case in Q4 2014, DDoS attacks were fueled by malicious actors seeking to gain media attention or notoriety from peer groups, damage reputations and cause disruptions in gaming services. Some of the largest console gaming networks were openly and extensively attacked in December 2014, when more players were likely to be affected. This trend continued in the first quarter of 2015, especially during a three-week period in January.

China again topped the list of DDoS attack sources, responsible for roughly 23 percent of Q1 2015 attack traffic. Germany followed with roughly 17 percent of the traffic, and the US came in third with roughly 12 percent of the traffic.

On the Akamai Edge network, seven web application attack vectors were tracked and analyzed for the Q1 report. Local File Inclusion (LFI) attacks were responsible for 66 percent of the attacks tracked, while SQL injection attacks made up more than 29 percent. Each of the five remaining attack vectors combined made up less than 5 percent of the attacks. While the vast majority of web application attacks were over HTTP connections, 8.5 percent of the attacks were over secure (HTTPS) connections, illustrating the fact that secure connections alone cannot guarantee protection.

On the Akamai Edge network, from an application layer attack perspective, the retail sector was hit harder than any other industry — fueled by a massive campaign against two retailers in March. This was followed closely by web application attacks against the media and entertainment industry. This is in contrast to the intense focus on gaming sites seen in the DDoS attack stats.

Additionally, the source IPs for web application attacks varied from what was observed for DDoS attacks. The US was the top source of attacking IPs for web application attacks (52.42 percent), followed by China (11.39 percent). But the vast majority of web application attacks — 80.78 percent — targeted US-based websites.

In Q1 2015, Akamai also tracked a number of new attack techniques and vulnerabilities that warranted the release of threat advisories. These are profiled in more detail in the Cloud Security Resources section. These include:

- DDoS agents targeting Joomla and other SaaS apps
- A heap-based buffer overflow vulnerability in Linux systems
- Attackers using new MS SQL reflection techniques
- Data breaches fueling login attacks

Changes in the mix of Akamai customers by industry, and the prevalence of certain attack vectors used to target those industries, plays a role in the trends we report.

1.1 / DDoS Activity / In Q1 2015, we saw a 35 percent increase in DDoS activity against customers, as compared to Q4 2014. Q1 2015 set a new record for the number of DDoS attacks observed over Akamai's Prolexic network, more than double the number of attacks recorded a year ago.

1.1A / DDoS Attack Bandwidth, Volume and Duration / As the number of attacks continues to increase quarter by quarter, the average (mean) peak attack bandwidth and volume continues to drop. Average peak attack bandwidth was 5.95 Gbps in Q1 2015, slightly down from the 6.41 Gbps average in Q4 2014, and significantly lower than the average peak of 9.70 Gbps seen in Q1 2014.

Compared to Q4 2014

- 35.24 percent increase in total DDoS attacks
- 22.22 percent increase in application layer (Layer 7) attacks
- 36.74 percent increase in infrastructure layer (Layer 3 & 4) attacks
- 15.37 percent decrease in average attack duration: 24.82 vs. 29.33 hours
- China was the top source of attacking IPs

Compared to Q1 2014

- 116.5 percent increase in total DDoS attacks
- 59.83 percent increase in application layer (Layer 7) attacks
- 124.69 percent increase in infrastructure layer (Layer 3 & 4) attacks
- 42.8 percent increase in the average attack duration: 24.82 vs. 17.38 hours

As with bandwidth, the average peak attack volume was down slightly to 2.21 million packets per second (Mpps) in Q1 2015, compared with the average peak of 2.31 Mpps in Q4 2014. Attack volume dropped significantly compared with Q1 2014, when the average peak was a record-setting 19.8 Mpps.

In Q1 2015, the average DDoS attack lasted 24.82 hours — a little more than a day. That represents a 15.37 percent decrease in attack duration compared with Q4 2014 (29.33 hours) and a 42.8 percent increase in attack duration compared with Q1 2014.

The trends of the past two quarters show that malicious actors are favoring lower bandwidth, but more frequent and longer attacks than a year ago.

1.1B / MEGA ATTACKS / In Q1 2015, eight DDoS attacks registered more than 100 Gigabits per second (Gbps), as shown in Figure 1-1. This is down slightly from Q4 2014, when there were nine mega attacks.

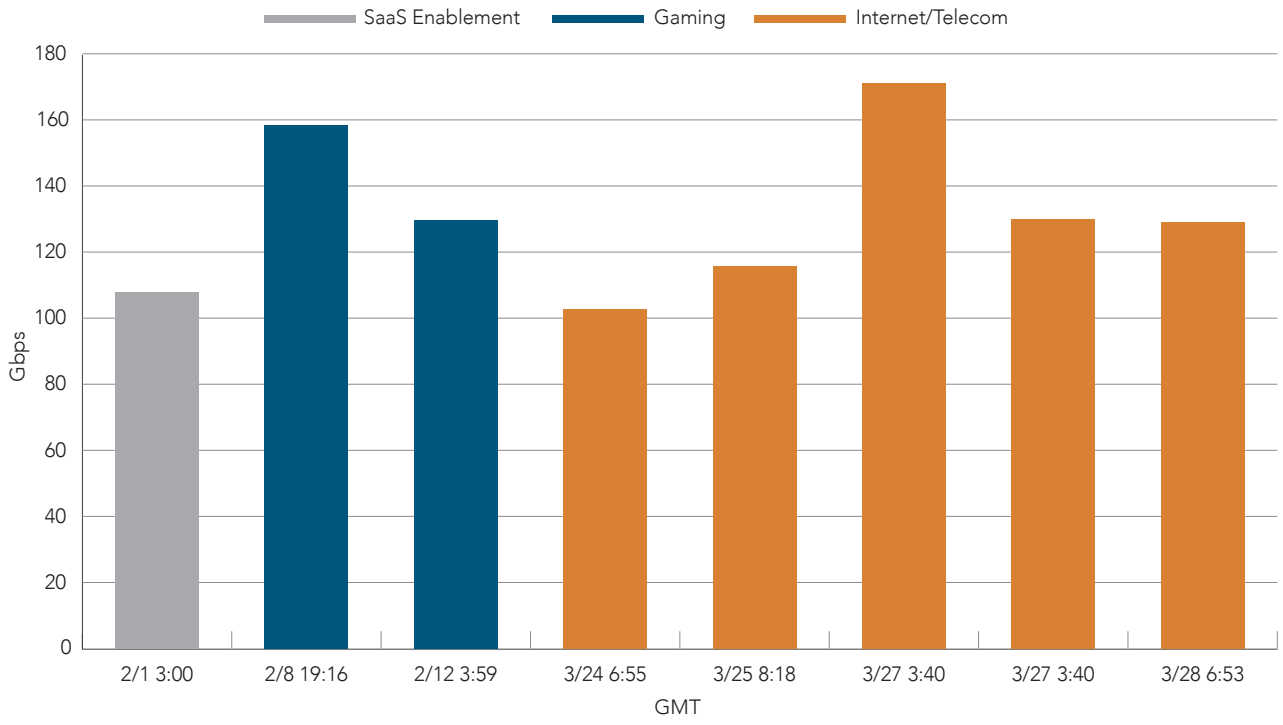


Figure 1-1: Q1 2015 saw eight mega attacks against Akamai customers, with the largest peaking at nearly 170 Gbps

In Q1 2015, the largest attack measured nearly 170 Gbps, an increase in size from the largest (158 Gbps) attack in Q4 2014. Of the eight mega-attacks, gaming received the largest share of attacks, albeit indirectly. The five attacks listed as Internet & Telecom were actually targeting gaming sites hosted on the customer network.

With the exception of the February 8 attack, these mega-attacks all contained SYN floods. The payload that produced the largest attack this quarter is shown in Figure 1-2. In Q3 2014, this same padded SYN payload, coupled with a UDP flood, produced a record-setting 321 Gbps DDoS attack.

```
12:34:04.270528 IP X.X.X.X.54202 > Y.Y.Y.Y.80: Flags [S], seq
1801649395:1801650365, win 64755, length 970
....E.....@...}.6.....6....Pkb......P...c.............
.................................................<sn
ip>.............................................
```

Figure 1-2: Padded SYN payload responsible for Q1 2015's largest DDoS attack

In Q4 2014, the largest attack was 158 Gbps, generated by a multi-vector volumetric attack that used the same padded SYN flood, along with a UDP fragment flood and a UDP flood.

1.1C / DDoS Attack Vectors / In Q1 2015, SSDP attacks represented the top overall infrastructure based attack, bypassing SYN floods, which was the top attack vector in Q4 2014. In Q1, SSDP attacks represented 20.78 percent of all attacks, a rise from 14.62 percent in Q4 2014. This vector first appeared in Q3 2014 and has not been subject to the same cleanup efforts as NTP and DNS, since many sources of SSDP reflection attacks are in-home devices. The victims of SSDP reflection abuse are unlikely to realize that their devices are participating in attacks. Even if these victims notice slowness in their networks, they may not have the expertise to fully troubleshoot and mitigate the cause of the issue.

The chart in Figure 1-3 displays the frequency of observed attack vectors at the DDoS layer.

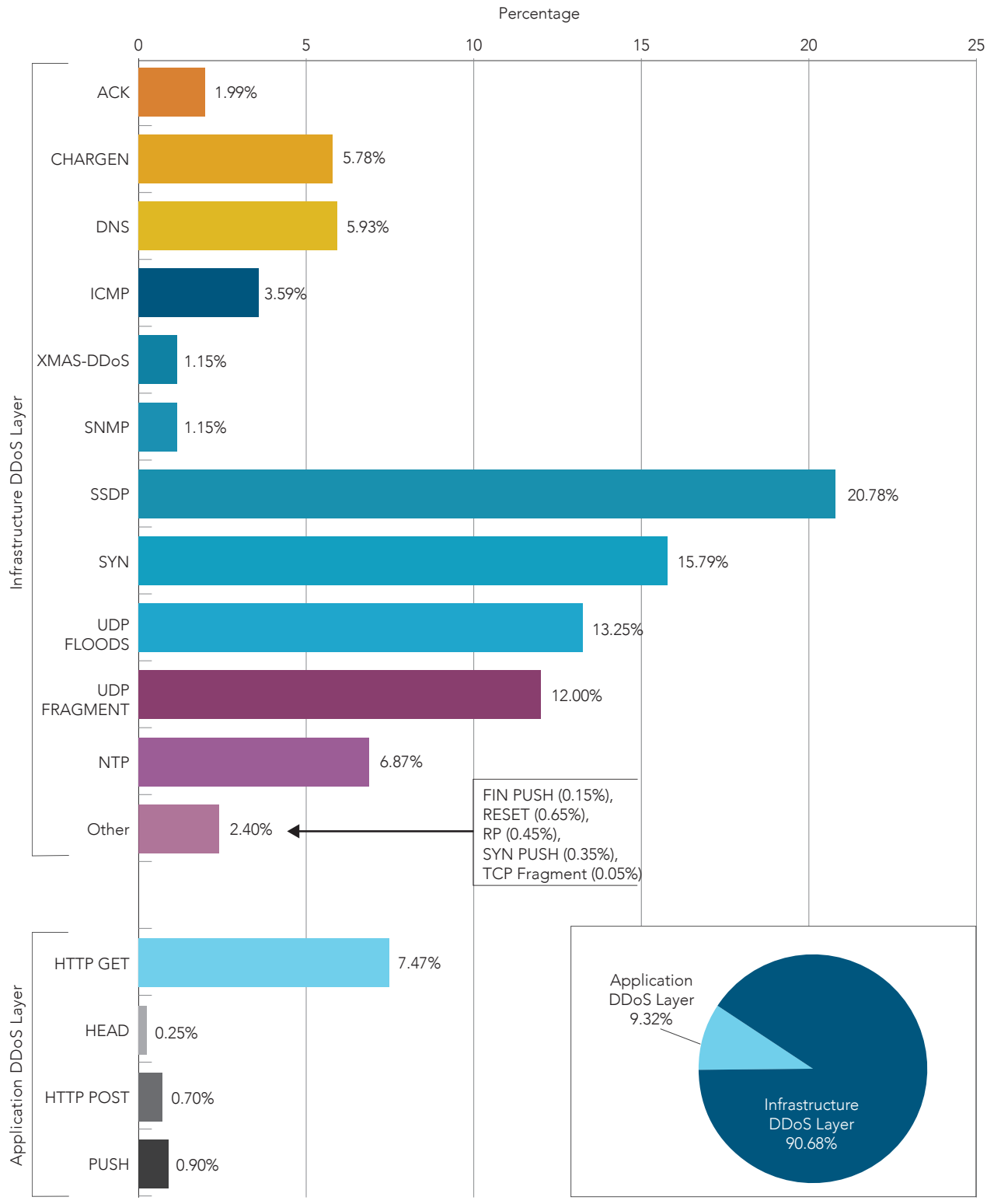


Figure 1-3: DDoS attack vectors, Q1 2015

In the bigger picture, infrastructure-based attacks accounted for the lion's share of DDoS activity in the first quarter. Application layer DDoS attacks accounted for less than 10 percent of all activity, while the infrastructure layer experienced 91 percent of DDoS attacks. This trend of mostly infrastructure attacks has continued for more than one year, as attackers have relied more and more on reflection vectors as primary DDoS attack methods. Not only are these reflection attacks easier to launch, but they also require fewer resources from the attacker.

That said, DDoS attack scripts on the application side have been shifting more towards the use of non-botnet based resources, such as attack scripts that leverage open proxies on the Internet. This trend, along with the continued abuse of WordPress and Joomla-based websites as GET flood sources, may pave the way to an increase in application-based DDoS attacks going forward.

1.1D / Infrastructure Layer vs. Application Layer DDoS attacks / SSDP attacks accounted for more than 20 percent of all attacks, while SYN floods accounted for nearly 16 percent of attacks. As the 100+ Gbps attacks show, the SYN flood attack plays a major role in the larger attacks. UDP floods accounted for 13 percent, while UDP fragments accounted for 12 percent. As stated in previous reports, the fragments are sometimes a byproduct of some infrastructure-based attacks. In particular, UDP-based CHARGEN and DNS reflection attacks together accounted for almost 12 percent of attacks. These attacks are known to produce payloads larger than 1,500 bytes, which in turn produce fragmented UDP flood traffic.

By comparison, in Q4 2014 the most used infrastructure-based attack vectors were SYN floods (17 percent), SSDP floods (15 percent), UDP fragment (14 percent), UDP floods (11 percent) and DNS attacks (11 percent). Additionally, NTP attacks accounted

for 8 percent, CHARGEN for 5 percent, ICMP for 4 percent, ACK floods for 3 percent and RESET flood for 1 percent. SSDP has continued to gain popularity since it was first observed back in Q3 2014.

At the application layer, HTTP GET attacks came in at 7 percent. HEAD, HTTP POST, and PUSH attacks accounted for less than 1 percent each. Many of the GET flood attacks were based on a combination of the Joomla, WordPress and GET flood attacks over proxy. These attacks also came in the form of redirected traffic from Asia. Other application-layer attacks were used less than 2 percent of the time, including HTTP POST (1 percent), HTTP PUSH (0.5 percent) and HTTP HEAD (0.2 percent).

HTTP GET floods have been consistently favored by attackers targeting the application layer. The top application-layer DDoS attack in Q4 2014 was HTTP GET floods at 8 percent of all attacks. Similarly, the top application layer DDoS attack in Q1 2014 was HTTP GET floods, at 9.28 percent of attacks.

A full comparison of attack vector frequency is shown in Figure 1-4.

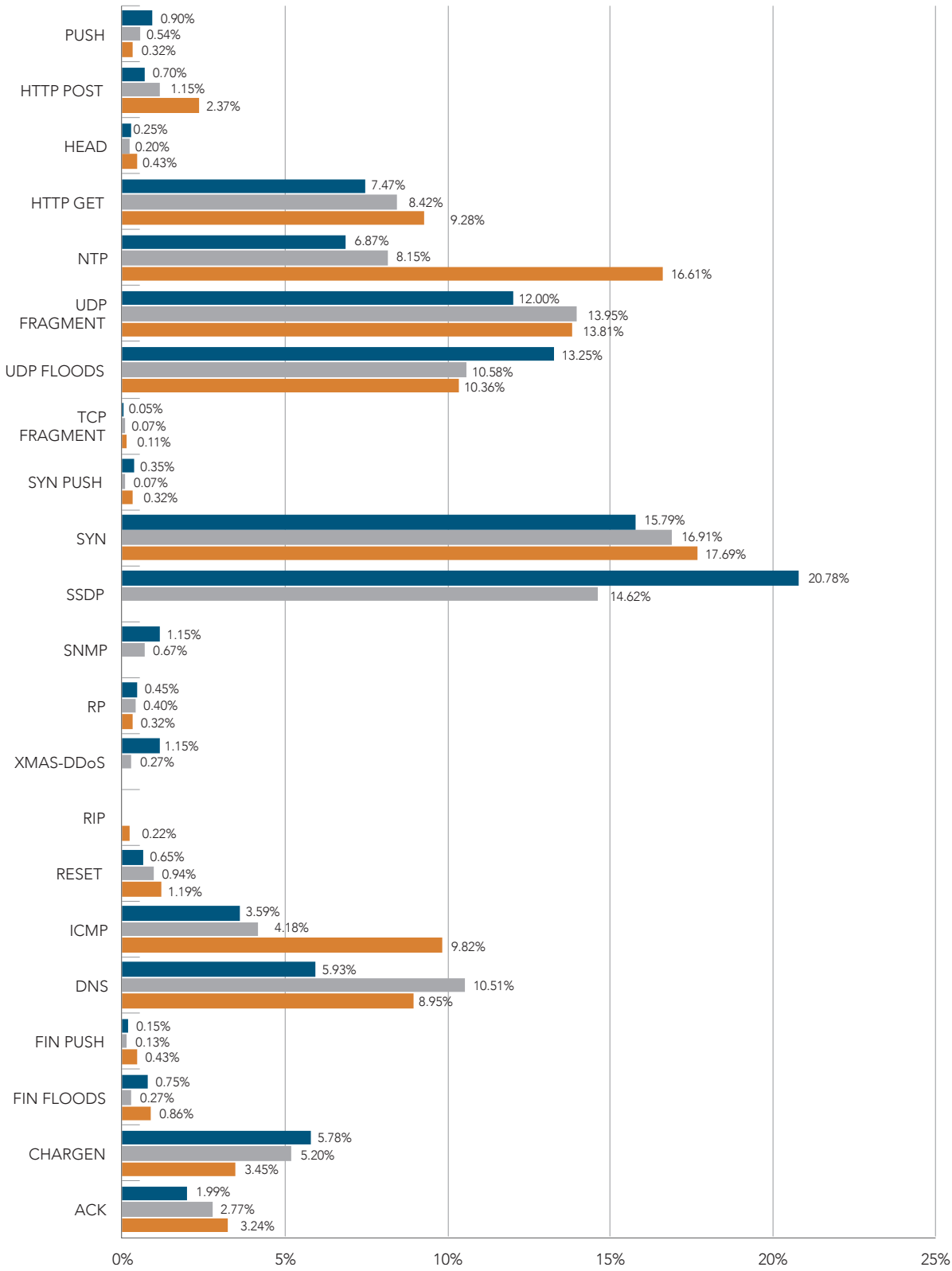


Figure 1-4: DDoS attack vector frequency observed over the Akamai's PLXrouted network

1.1E / Top 10 Source Countries / Looking at the top 10 source countries for DDoS attacks, we see China again topping the list, with roughly 23 percent of Q1 2015 traffic. Germany followed with roughly 17 percent of the traffic, and the US came in third with roughly 12 percent of the traffic, as shown in Figure 1-5. Combined, China, Germany and the US accounted for more than 50 percent of attacking IPs in the quarter.

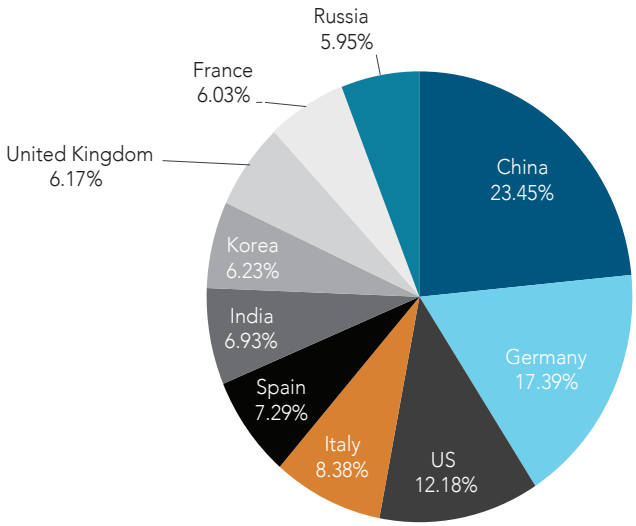


Figure 1-5: Non-spoofed attacking IP addresses by source country, for DDoS attacks mitigated during Q1 2015

The numbers show a drop from Q1 2014 in source country participation percentage, when the US accounted for some 32 percent of all attack traffic, followed by China at 18 percent and Germany at 12 percent. For further comparison, in Q1 2014 the US accounted for 21 percent, China was 18 percent, and Thailand was 15 percent. In Q1 2014, Germany came in at 8 percent. A full comparison is shown in Figure 1-6.

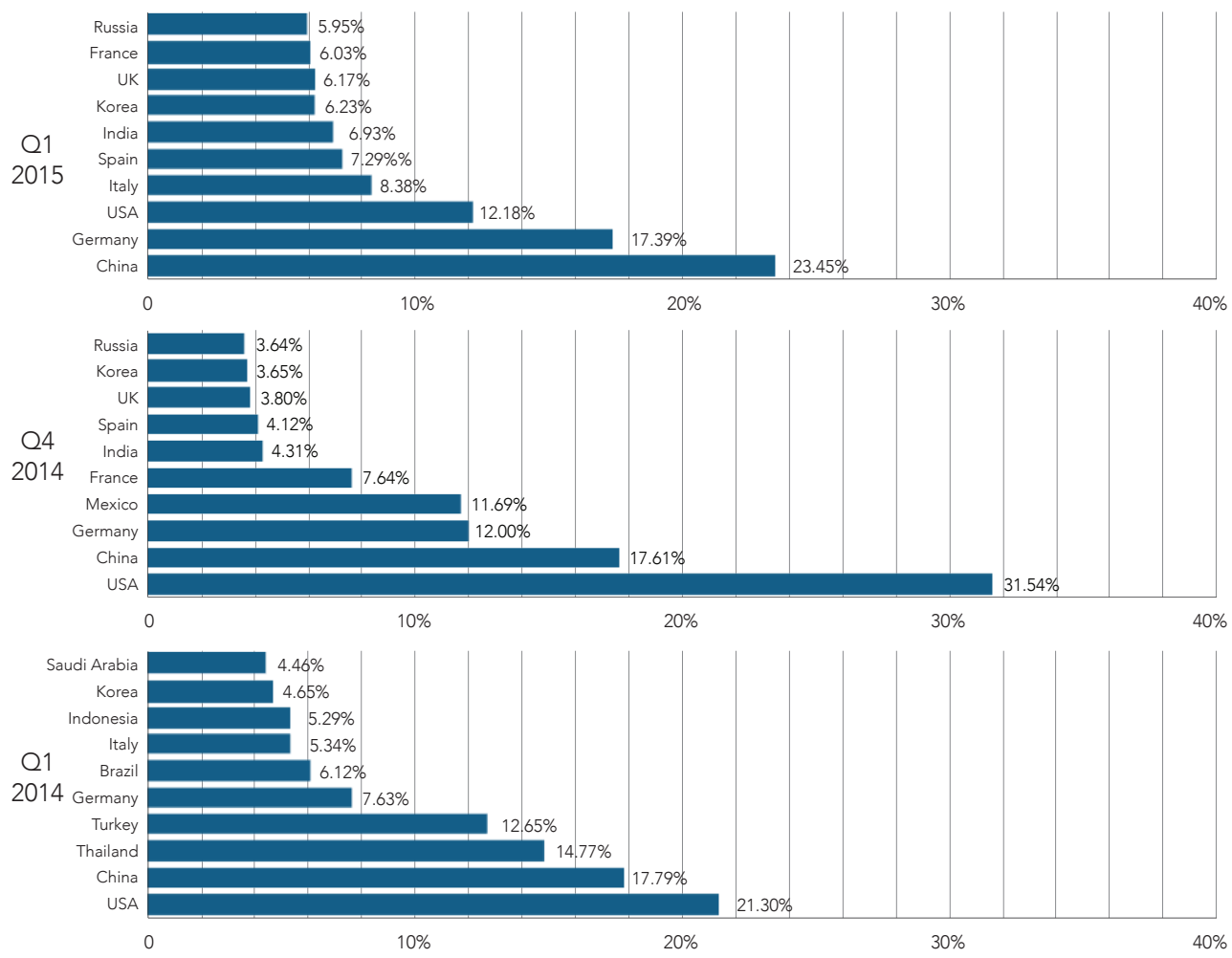


Figure 1-6: China, Germany and the US are consistently among the top 10 sources for non-spoofed attacking IPs

The percentage drop does not indicate a drop in attacks from these countries. Compared to last quarter, DDoS attacks increased 35 percent and have more than doubled in number since Q1 2014. This quarter's top attack source country, China, had a 66 percent increase in attack source IPs compared with last quarter's top source country (US). Some of the increase in attack sources could be attributed to the increase in redirected traffic from Asia. This redirected traffic appeared to be a DDoS attack.

1.1F / Target Industries / The gaming sector was particularly hard hit in Q1 2015, accounting for more than 35 percent of all attacks. Gaming was followed by software and technology, which suffered 25 percent of all attacks, as shown in Figure 1-7. Internet and telecom suffered 14 percent of attacks, followed by financial services (8.4 percent), media and entertainment (7.5 percent), education (5 percent), retail and consumer goods (2.3 percent), and the public sector (2 percent).

Since software-technology, Internet-telecom and media-entertainment all have a hand in gaming products and services, we counted them as indirect attacks on gaming.

Gaming / Gaming has remained the most targeted industry since Q2 2014 and remained steady at 35 percent compared to last quarter. In Q4, attacks were fueled by malicious actors seeking to gain media attention or notoriety from peer groups, damage reputations and cause disruptions in gaming services. Some of the largest console gaming networks were openly and extensively attacked in December 2014, when more players were likely to be affected. This trend continued in the first quarter of 2015, especially in January.

Software and technology / The software and technology industry includes companies that provide solutions such as Software-as-a-Service (SaaS) and cloud-based technologies. This industry saw a slight 1 percent drop in attack rates compared to last quarter.

Internet and telecom / The Internet and telecom industry includes companies that offer Internet-related services such as ISPs and CDNs. It was the target of 14 percent of all attacks, a 3 percent increase over the previous quarter.

Financial services / The financial industry includes major financial institutions such as banks and trading platforms. The financial industry saw a small (1 percent) uptick in attacks from the previous quarter.

Media and entertainment / The media industry saw a slight drop in the percentage of attacks, from 10 percent in Q4 2014 to 7.5 percent in Q1 2015.

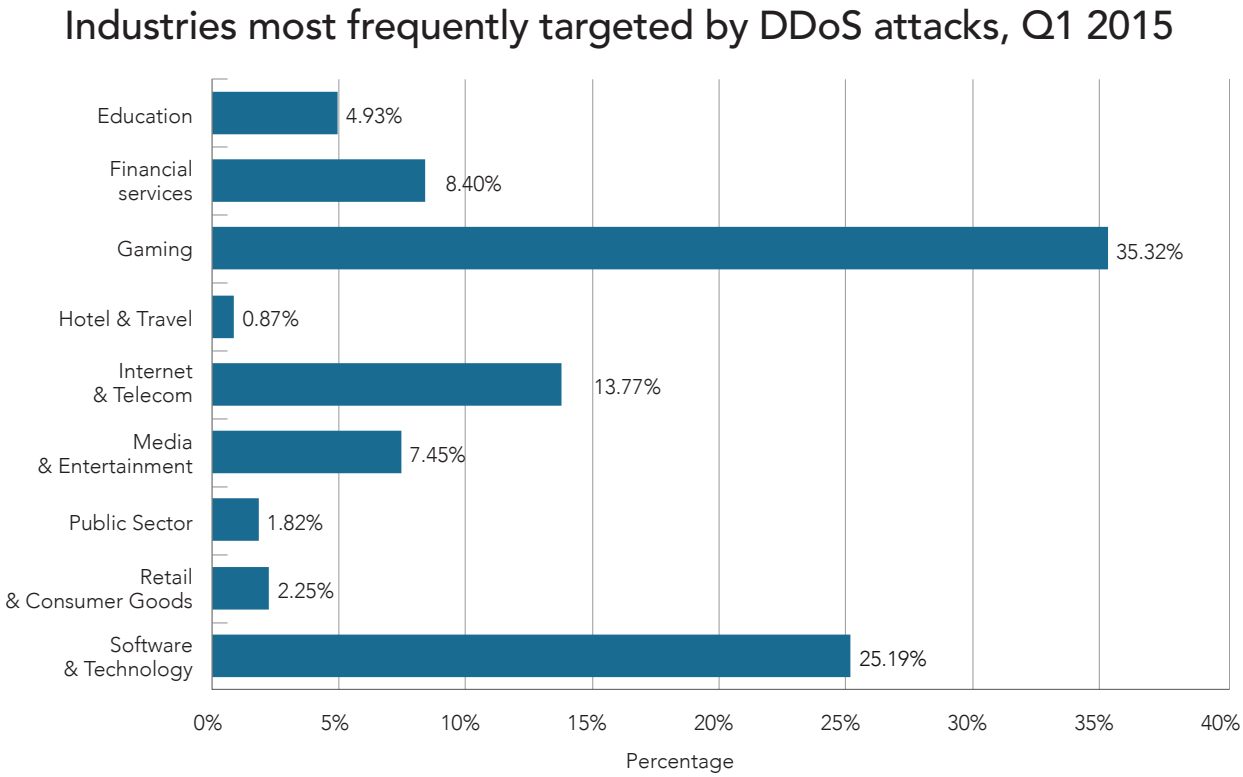


Figure 1-7: In Q1 2015, the gaming and software industries were targeted more than 60 percent of the time

1.1G / Changes in DDoS Attacks Per Week / Figure 1-8 shows the percentage increase and decrease of the total number of attacks per week in Q1 year-over-year. Of the three months of Q1 2015, Akamai mitigated the greatest number of DDoS attacks in January. The third and fifth weeks of January were the busiest. The third week posted a 363 percent increase compared to the same week a year before. Attacks against the gaming industry account for the large spike in traffic.

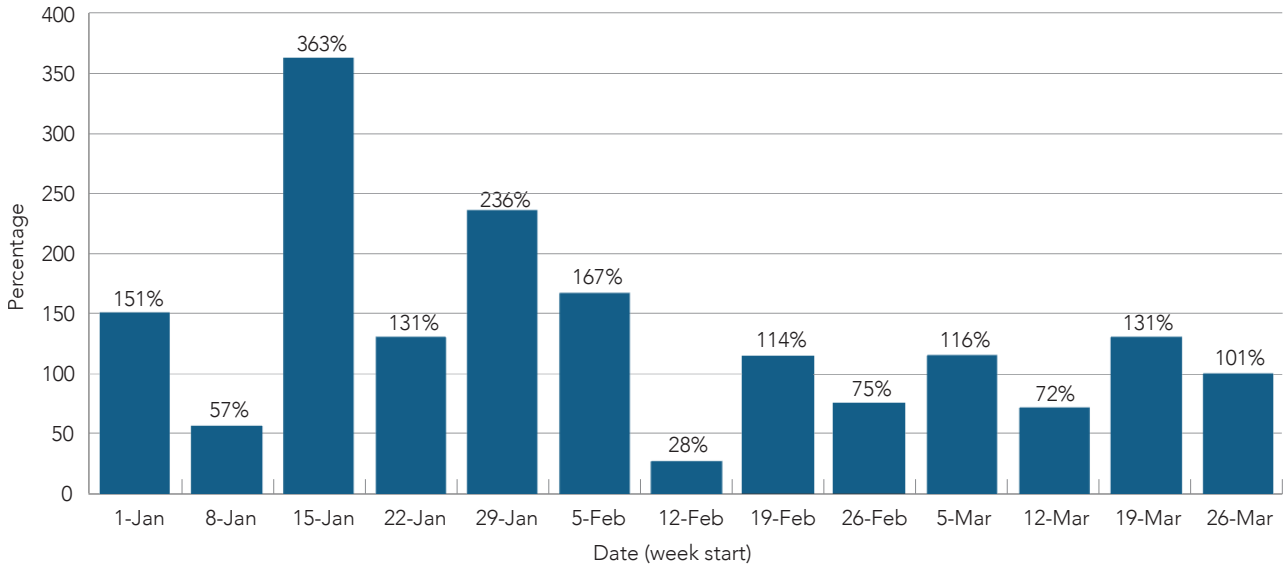


Figure 1-8: The number of attacks in Q1 2015 increased in all but four weeks of the quarter, compared with Q1 2014.

1.1H / DDoS Attacks — A Two-Year Look Back / Figure 1-9 and Figure 1-10 illustrate the measure of central tendency. In Figure 1-9, the black line shows the median attack size, and the outline of the box shows the interquartile range. In other words, the bottom line of the box represents the top of the first 25 percent of attack bandwidth, the middle line represents the median attack size and the top line shows the top bandwidth of the 75th percentile of all attacks. What we see is that half of all DDoS attacks fall within the ranges within the box.

In Q1 2014, there was a sharp uptick in the size of DDoS attacks. This is visible both in the upward shift in the upper bound of the interquartile range (IQR), and an upward shift in the median. In Q4 2014 and continuing into Q1 2015, we see a contraction in the size and number of large attacks. In Q3 2014, there were 17 mega attacks (with bandwidth exceeding 100 Gbps), while in Q4 2014 and Q1 2015 there were eight and nine mega attacks respectively. That said, there has not been a significant shift in the median attack size.

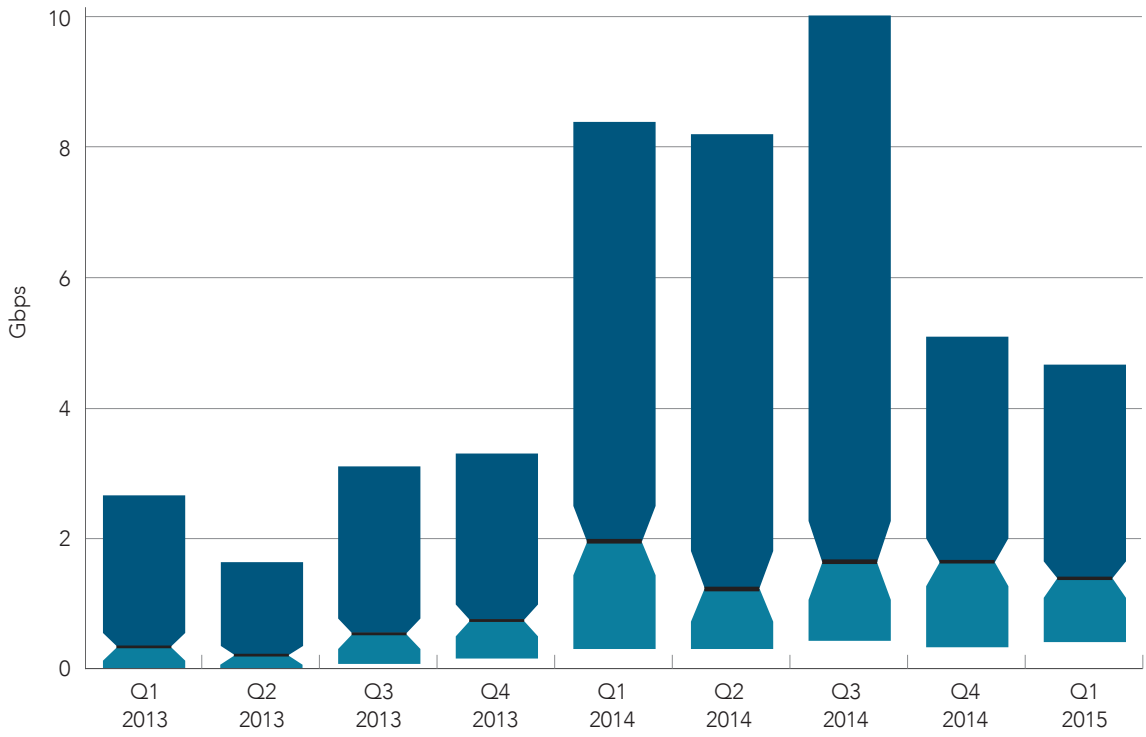


Figure 1-9: The middle 50 percent of DDoS attack data from Q1 2013 - Q1 2015. The top and bottom 25 percent are excluded as outliers.

Figure 1-10 shows the distribution of all DDoS attacks in each of the last nine quarters, including the outliers. From the scatter graph, one can see that the vast majority of attacks are relatively small in size — approximately 10 Gbps or less.

The increases in attack numbers can be seen starting in Q4 2014, where a higher concentration of attacks is populating the bottom portion of the scatter graph.

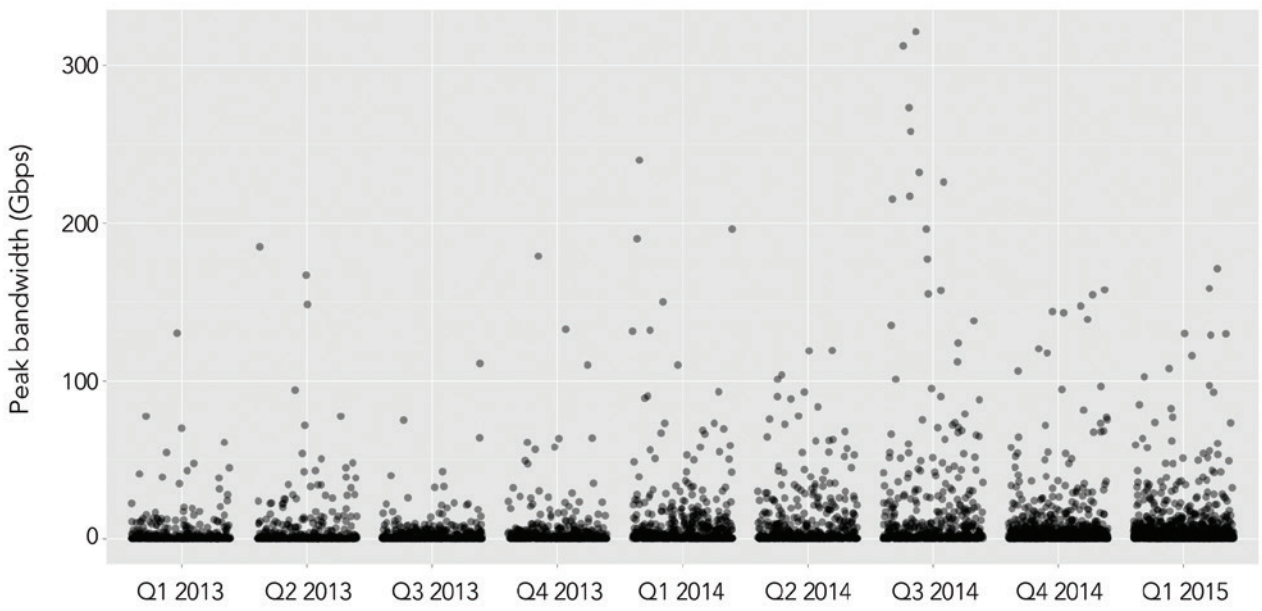


Figure 1-10: The frequency of attacks and average size of campaigns have gradually increased over the course of two years

1.2 / KONA Web Application Firewall Activity / For the Q1 2015 report, we concentrated our analysis on seven popular web application attack vectors. The attack vectors do not entirely cover the OWASP TOP 10, MITRE, CWE/SANS TOP 25 or WASC TC industry vulnerability lists. However, they do represent a cross section of many of the most common categories seen in each of these lists. Akamai's goal was not to validate any one of the vulnerability lists, but instead to look at some of the commonalities among them across a large network. As with all sensors, the data sources used by Akamai have different levels of confidence; for this report, we focused on traffic where Akamai has a high confidence in the low false-positive rate of its sensors.

The following seven attack vectors accounted for 178.85 million web application attacks observed on the Akamai Edge network. Other web application attack vectors are excluded from this section of the report.

SQLi / SQL injection is an attack where user content is passed to an SQL statement without proper validation.

LFI / Local file inclusion is an attack where a malicious user is able to gain unauthorized read access to local files on the web server.

RFI / Remote file inclusion is an attack where a malicious user abuses the dynamic file include mechanism, which is available in many programming languages, and loads remote malicious code into the victim web application.

PHPi / PHP injection is an attack where a malicious user is able to inject PHP code, which gets executed by the PHP sinterpreter.

CMDi / Command injection is a vulnerability where a malicious user has the ability to execute arbitrary shell commands on the target system.

JAVAi / Java injection is an attack where a malicious user injects Java code, abusing the Object Graph Navigation Language (OGNL), a Java expression language. This kind of attack became very popular due to recent flaws in the Java-based Struts Framework, which uses OGNL extensively in cookie and query parameter processing.

MFU / Malicious file upload (or unrestricted file upload) is a type of attack where a malicious user uploads unauthorized files to the target application. These potentially malicious files can later be used to gain full control over the system.

1.2A / Web Application Attack Vectors / During Q1 2015, Akamai observed more than 52.15 million SQLi attacks. This accounted for 29.16 percent of web application attacks. A substantial portion of these attacks is related to attack campaigns against two companies in the travel and hospitality industry. The majority of the attacking source IPs in these campaigns originated in Ireland.

Similarly, there were a substantial number of LFI attack attempts in March, which are attributed to German IPs bombarding two large retailers as a part of a massive campaign. These attacks were volumetric in nature and were found to be trivial attempts to discover an LFI vulnerability targeting the WordPress RevSlider plugin. During week 12 alone, we saw 74.85 million LFI attacks — 16 times as many LFI attacks as in week 11. Overall, LFI attacks accounted for 118.55 million application attacks during Q1 2015, representing 66.29 percent of the analyzed web application attacks for the quarter.

MFU attacks were the third most observed attack vector (2.23 percent). This is related to several published MFU vulnerabilities such as:

- KCFinder file upload vulnerability
- Open Flash Chart file upload vulnerability (CVE-2009-4140)
- appRain CMF (uploadify.php) unrestricted file upload exploit (CVE-2012-1153)
- FCKeditor file upload vulnerability (CVE-2008-6178)

MFU attacks were followed by PHPi attacks (1.23 percent). The rest of the attack types — RFI, CMDi and JAVAi — accounted for less than 1 percent each.

The weekly breakdown of attack vectors is shown in Figure 1-11.

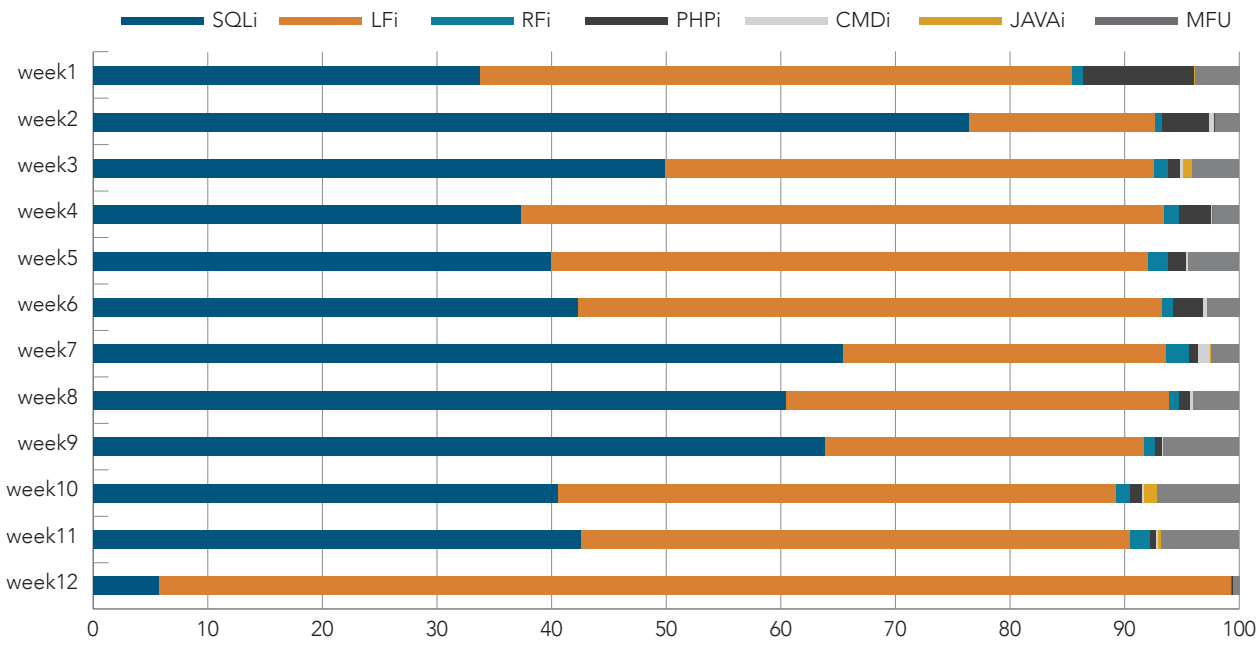


Figure 1-11: SQLi and LFI were responsible for more than 90 percent of web applications analyzed in Q1 2015

1.2B / Web Application Attacks over HTTP vs. HTTPS / Among the application attacks analyzed for the Q1 2015 report, 163.62 million were sent over (unencrypted) HTTP. This represented 91.48 percent of the application attacks.

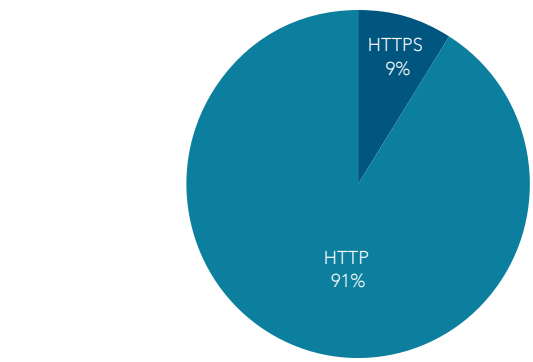


Figure 1-12: The majority of web application attacks anaylzed for the Q1 2015 report were not encrypted

Given that a large percentage of websites either do not use HTTPS for all of their web traffic, or use it only for safeguarding certain sensitive transactions (such as login requests, etc.), the comparison between HTTP vs. HTTPS should be used only for understanding attack trends between these two communication channels.

That said, encrypted connections (over HTTPS) did not automatically safeguard applications from attack. There is no reason to believe that the attackers would not have followed a shift of the vulnerable applications to HTTPS. There were 15.23 million attacks over HTTPS observed during the quarter, making up 8.52 percent of the attacks. Figure 1-12 shows the ratio between HTTPS and HTTP attacks.

Of the 15.23 million attacks over HTTPS, the most prevalent attack vectors were LFI (71.54 percent), and SQLi (24.20 percent). HTTPS-based RFI attacks accounted for 2.12 percent while PHPi attacks accounted for 1.72 percent. CMDi, JAVAi and MFU attacks accounted for less than 1 percent each. The weekly breakdown of attack vectors is shown in Figure 1-13.

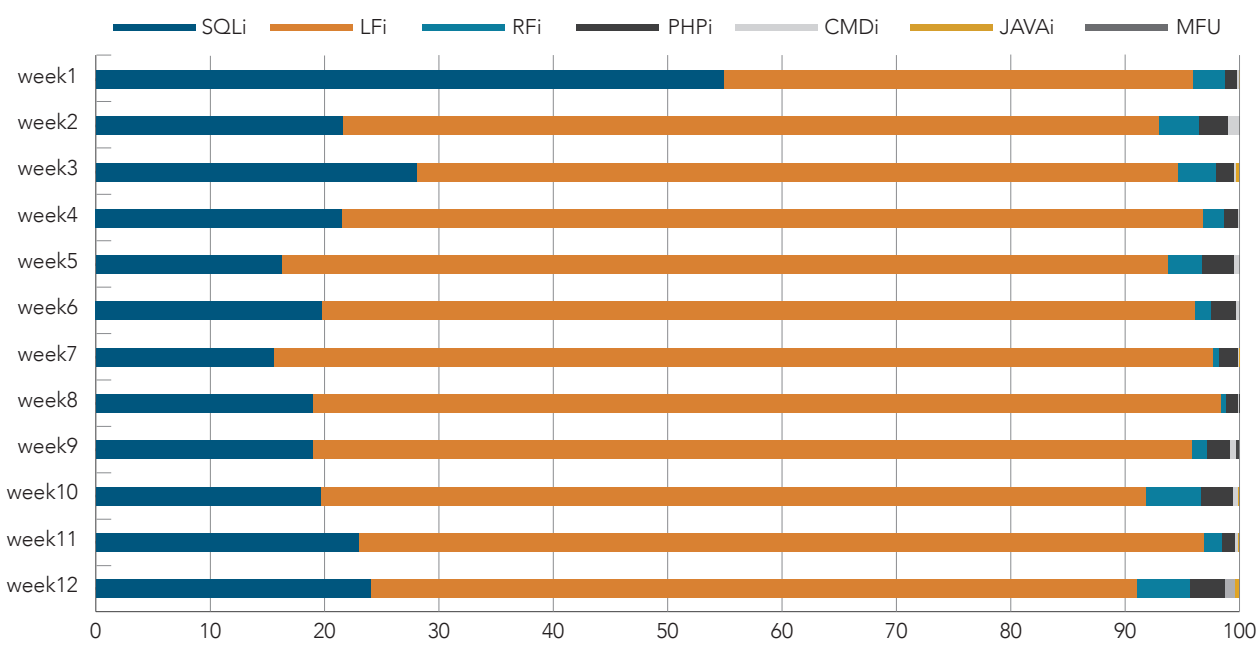


Figure 1-13: LFI and SQLi attacks were the most frequently observed attacks over encrypted connections

When comparing HTTPS-based attacks in each category, against the total of that category, we see that attacks over HTTPS were responsible for 7.07 percent of SQLi attacks, 9.19 percent of LFI attacks, 22.64 percent of RFI attacks, 11.89 percent of PHPi attacks, 14.25 percent of CMDi attacks, 6.44 percent of JAVAi attacks and less than 1 percent of MFU attacks.

1.2C / Top 10 Source Countries / For the attacks analyzed in this report, the US was the top source country of attacking IPs (52.42 percent), followed by China (11.39 percent), Brazil (6.09 percent), India (5.33 percent), the UK (4.55 percent), Canada (4.39 percent), France (4.24 percent), and Russia (4.17 percent). Indonesia and Italy were responsible for less than 4 percent each. The web application attacks analyzed here occur after a TCP session is established. Therefore, the geographic origins of the attack traffic can be stated with high confidence. This is tempered by the possible use of proxy services, which can allow attackers to hide their own locations. See Figure 1-14.

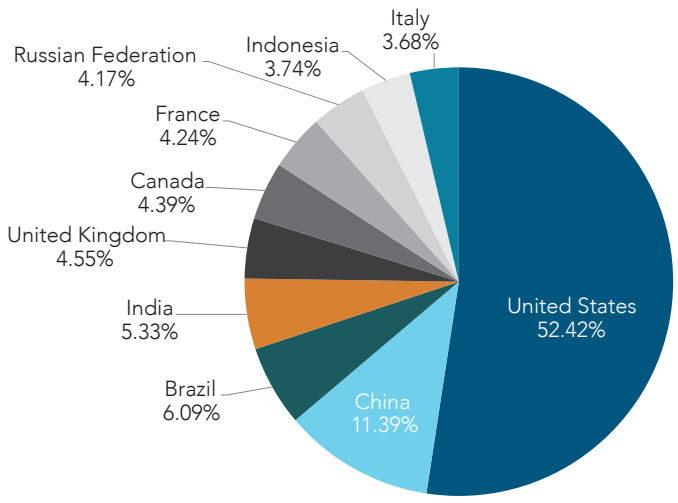


Figure 1-14: The US and BRIC (Brazil, Russia, India, China) countries were responsible for nearly 80 percent of web application attacks analyzed in Q1 2015

1.2D / TOP 10 TARGET COUNTRIES / US-based websites were by far the most targeted for web application attacks in Q1 2015, receiving 81.6 percent of all attacks. Australian-based websites were a distant second with 2.27 percent of attack traffic. UK-based websites were the third most targeted at 2.18 percent, followed by Japan-based sites at 1.80 percent. India, Canada, Korea, Hong Kong, Brazil and Finland-based websites were each targeted in less than 2 percent of attacks, as shown in Figure 1-15.

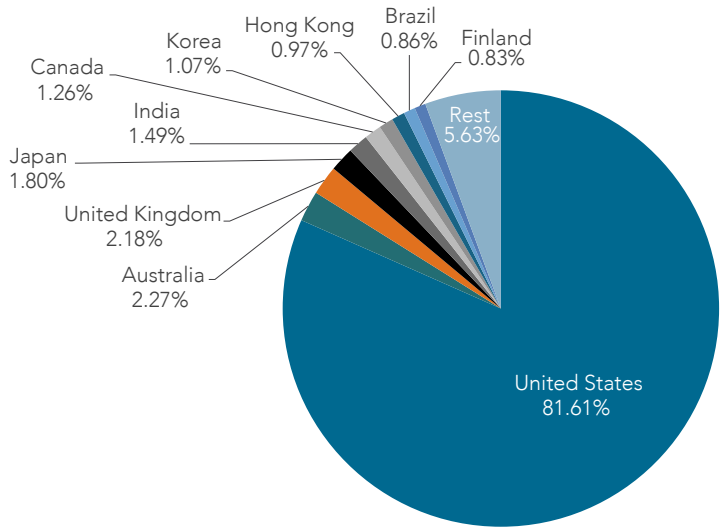


Figure 1-15: The US was targeted with web application attacks in 81 percent of the analyzed attacks

1.2E / A NORMALIZED VIEW OF WEB APPLICATION ATTACKS BY INDUSTRY / Akamai has long tracked DDoS attacks at both the application and network layer, and DDoS attack statistics are typically the most commented on, reprinted, and discussed stats that we produce. Over the years, customers have asked for a similar view into the sneaky application layer attacks that plague enterprises, governments and others; the attacks that hard-working organizations such as the Open Web Application Security Project (OWASP) have typically tracked and ranked according to prevalence and danger.

But figuring out how to give our customers a view of what we see has been a long and arduous challenge. Although Akamai has visibility into 15 – 30 percent of the world's web traffic, the challenge in meeting this goal has been threefold: how to store the data we see, how to query it, and finally, how to report on it meaningfully.

Methodology / In the past two years, we've made great progress in tackling the first two challenges. Storage, for example, has been largely met by the creation of the CSI (Cloud Security Intelligence) platform, which stores more than 2 petabytes (PB) of threat intelligence data (the equivalent of 2,000 terabytes). This allows Akamai to store more than 10 TB of attack data every day, which gives us roughly 30 – 45 days of application layer attack data at any given moment in time. Querying the data has taken a bit more finesse. During the past two years, we've been busy hiring a number of data scientists, analysts and researchers. Today, those researchers make up the Akamai Threat Research team, a team that has set up dozens of heuristics that automatically query the stored data on an hourly basis. The insight they extract from the data, feeds improvements to our Kona Site Defender application protections and our Client Reputation product. The final challenge is reporting on the data. This article represents the first attempt to report on the data that CSI collects.

Our reporting methodology undertook the following assumptions. We divided all Akamai customers into eight verticals. (Note: The verticals we tracked for application layer attacks are slightly different than they are for network layer attacks. This is because the integration of the Prolexic and Akamai customer tracking systems is a work in progress.) For each of the customers in these eight verticals, we tracked the number of malicious requests across the seven categories of attacks featured in this report during a 12-week period. The frequency of these attack vectors and the accuracy of the signatures detecting each of the categories, were both given weight in the selection of categories.

In order to normalize samples, we removed every sample that accrued more than 5 percent of total attacks in a week in any single attack vector. Doing so helped smooth out spikes, and what we consider to be anomalies, in the data. After adding up all attacks per vertical and type, we divided the number of attacks in each vertical by the number of customers in every given vertical. By doing so, we get the average number of attacks per customer in each vertical.

Observations / The industries that were subjected to the greatest number of malicious requests were the retail and media/entertainment verticals. Given that attacks into financial services organizations tend to grab headlines, this was a somewhat surprising finding. The attacks by the Izz ad-Din al-Qassam Cyber Fighters against the financial services sector, from 2012 – 2014, forced many of those organizations to harden their sites significantly. Conversely, the high-profile retail and media attacks and breaches of last year signaled that these industries were softer targets, and many attackers began probing them for vulnerabilities and exploitation. See Figure 1-16.

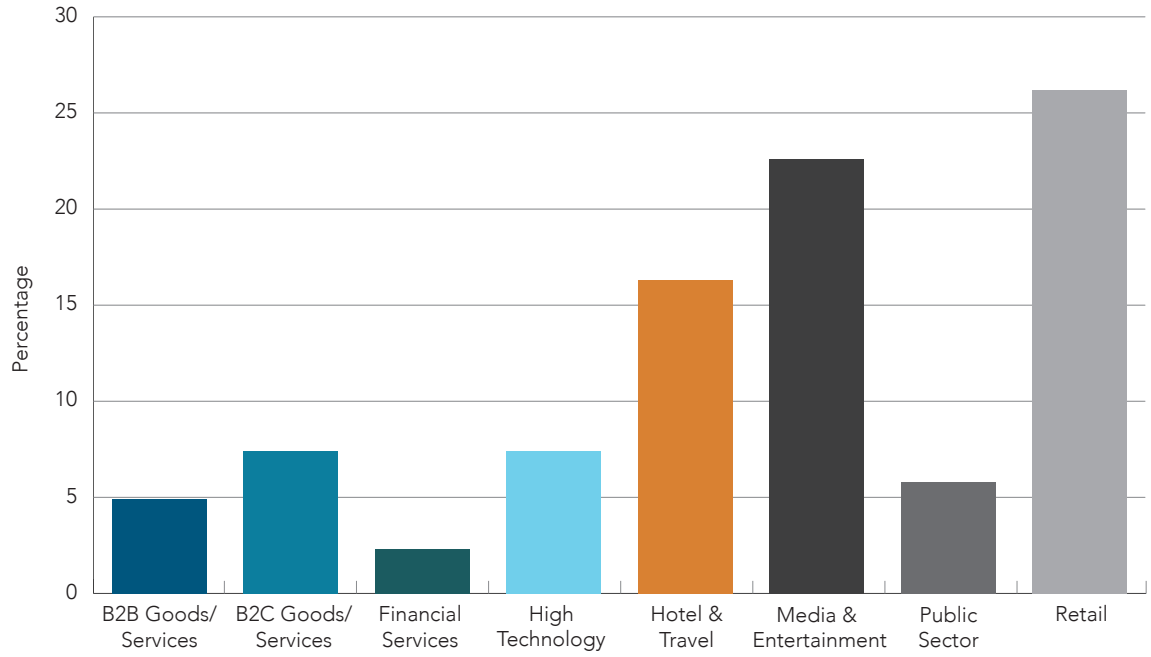


Figure 1-16: Distribution of the seven analyzed web application attack vectors across the most commonly targeted industries

LFI attacks consist of including local files and resources on the web server via direct user input (e.g. parameter or cookie). This attack is possible when a resource include is not properly sanitized or whitelisted, and allows certain manipulations such as using directory traversal techniques. The LFI attack will attempt to read sensitive files on the server that were not intended to be publicly available, such as password or configuration information. LFI attacks were the most frequently observed attack vector in Q1 2015, most often targeting retail and media/entertainment sites.

The second most common attack vector, SQLi, takes advantage of improper coding of your web applications that allows attackers to inject SQL statements into, predefined backend SQL statements such as those used by a login form. This may in turn allow the attacker to gain access to the data held within your database among other potential malicious actions. SQLi and LFI attacks were attempted against

Akamai customers more than any other attack vector, and companies in the retail and media/entertainment space were the most commonly attacked, as shown in Figure 1-17.

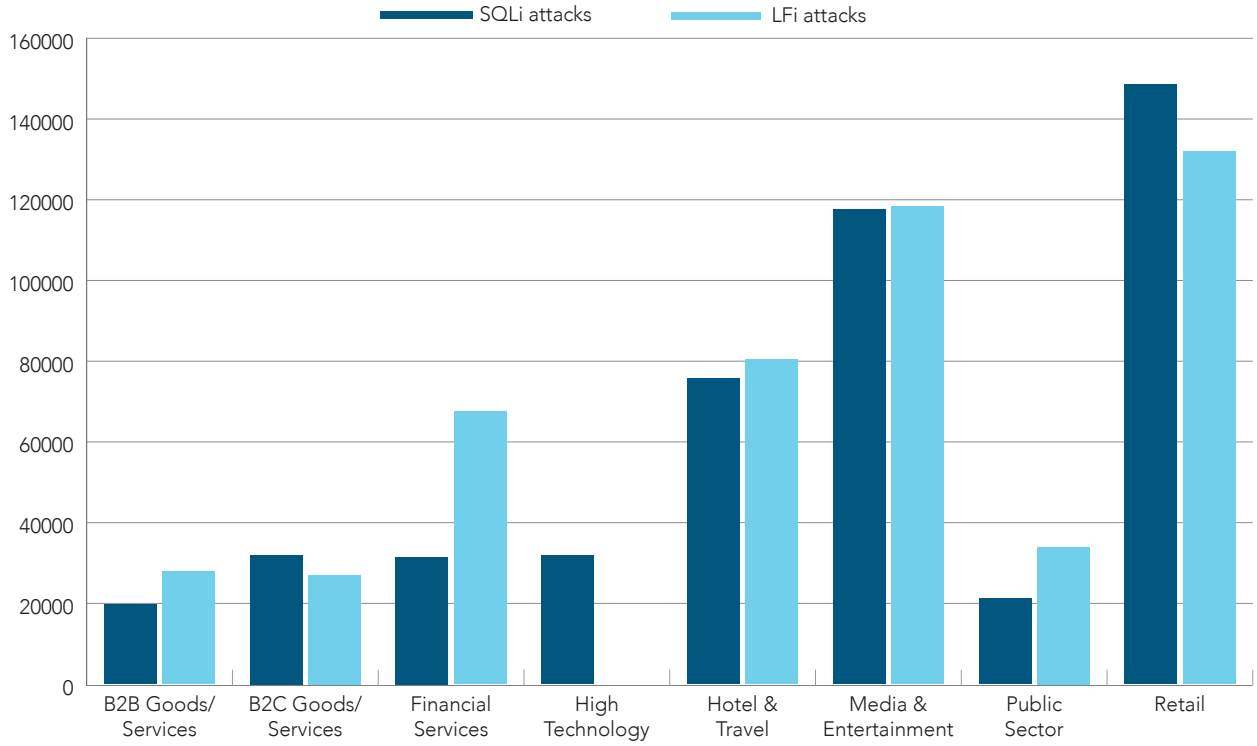


Figure 1-17: Retail and media/entertainment sites were the most popular targets of web application attacks in Q1 2015

The retail sector saw the most SQLi attacks in Q1, although the company that was attacked more than any other company was a travel website. That specific site was particularly hard hit, with five times as many SQLi attempts as the next most attacked site.[1]

MFU attacks were the third most commonly used attack vector. In Q1 2015, Akamai customers saw 3.99 million malicious file upload attempts. MFU attempts were directed at the hotel and travel industry more than any other vertical.

[1] Cross-site scripting (XSS) data was not collected for the Q1 report, but will be collected for the Q2 2015 report

RFI attack were the fourth most commonly employed attack vector in Q1 2015. Media and entertainment, high tech, and retail, were the three industries most often targeted by RFI attacks in Q1 2015.

During Q1 2015, the PHPi attack vector was most commonly performed against the retail sector. The breakdown of MFU, RFI and PHPi attacks is shown in Figure 1-18.

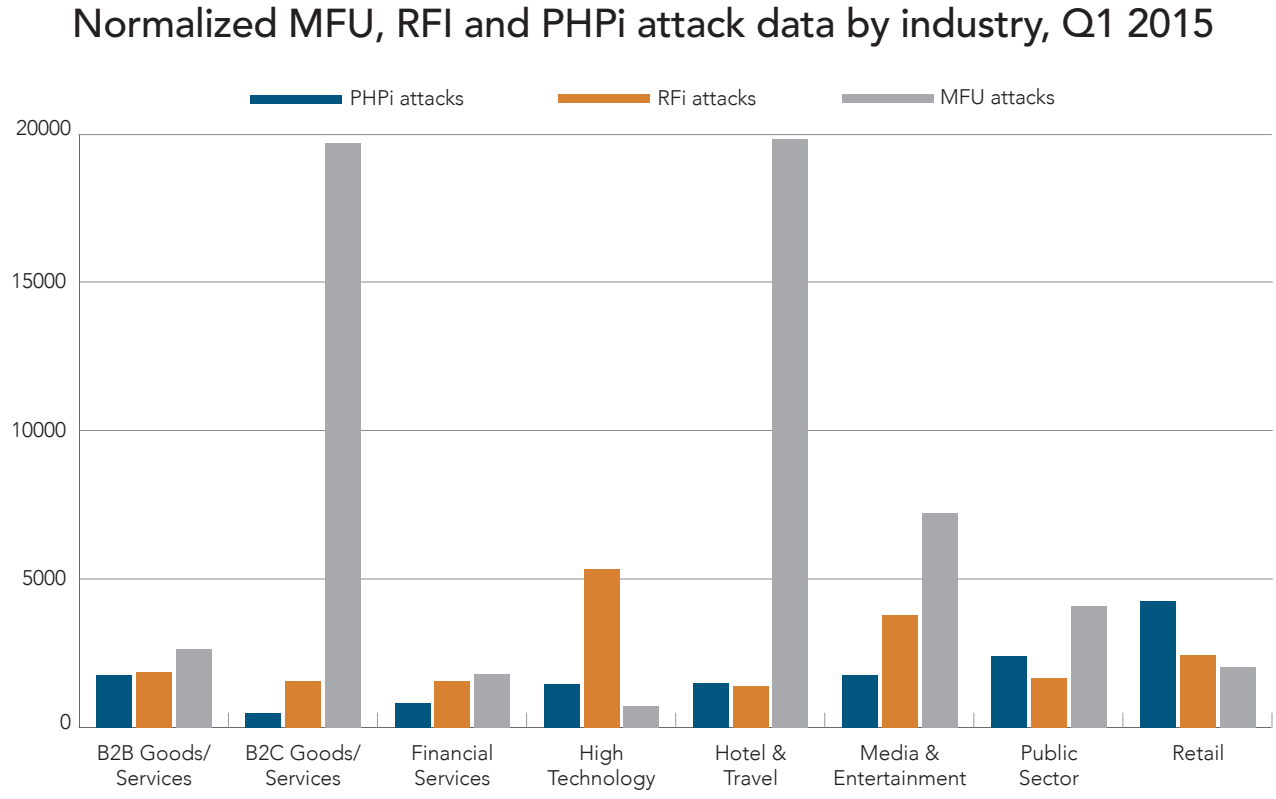


Figure 1-18: Consumer goods and the hotel/travel industries were particularly hard hit by MFU attacks in Q1 2015

In Q1 2015, CMDi attacks most frequently targeted the media/entertainment industry, while JAVAi attacks were the least popular attack vector seen in Q1 2015. The breakdown of CMDi and JAVAi attacks by vertical is shown in Figure 1-19.

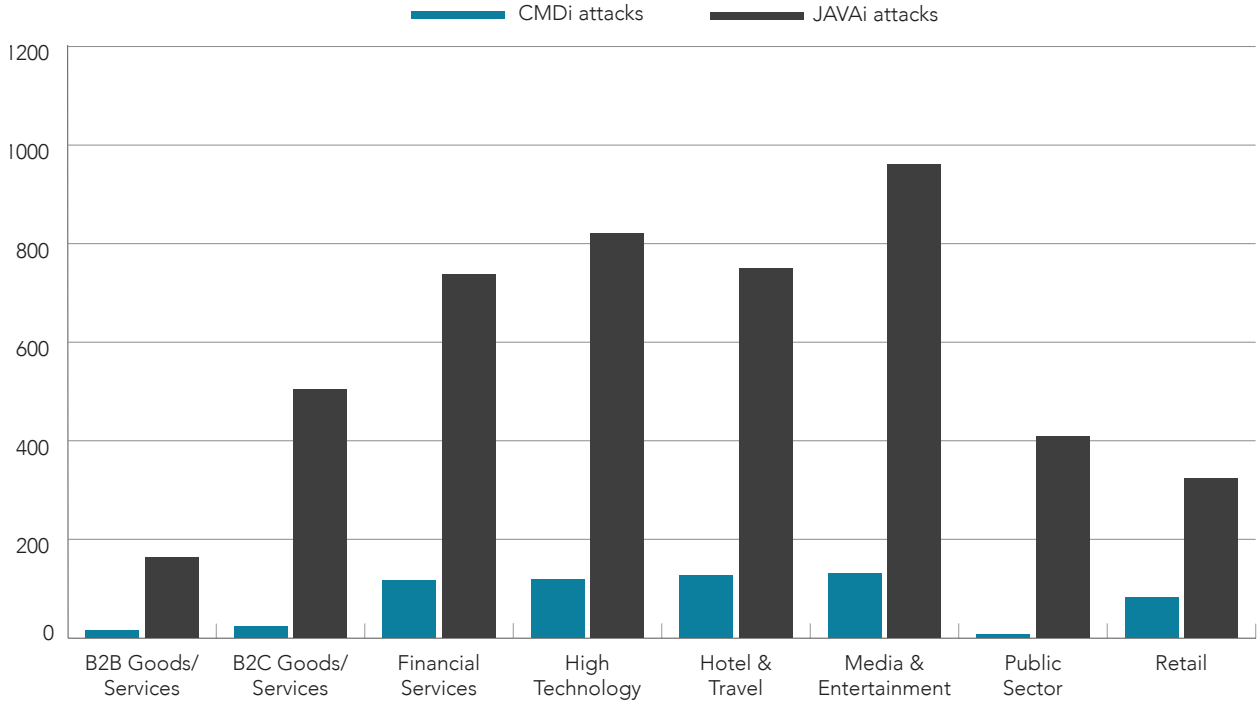


Figure 1-19: The media/entertainment and high tech industries were most frequently targeted by JAVAi and CMDi attacks during Q1 2015

1.2F / Future Web Application Attacks Analysis / As CSI and the capabilities of our Threat Research team grow, we look forward to continuing to report on data such as those listed above, as well as new trends as they develop. Please engage us and let us know which types of data you'd like to see in the next report. As long as we can guarantee the anonymity of our customers, we'll continue to share as much as we can in the most useful way possible.

1.3 / Data Sources / The Akamai platform consists of more than 180,000 servers in more than 100 countries around the globe and regularly transmits between 15 – 30 percent of all Internet traffic. In February 2014, Akamai added the Prolexic network to its portfolio, a resource specifically designed to fight DDoS attacks. This report draws its data from the two platforms in order to provide information about current attacks and traffic patterns around the globe.

The Akamai platform provides protection by being massively distributed, protecting by the use of the Kona Web Application Firewall (WAF) and the ability to absorb attack traffic close to where it originates. In contrast, the Prolexic solution protects by routing traffic to scrubbing centers where experienced incident responders use a variety of tools to remove malicious traffic before passing it to the origin servers. The two types of technology are complementary and provide two lenses through which we can examine traffic on the Internet.

[SECTION]^2 ATTACK SPOTLIGHT The Significance of a 100 Gbps Attack

In Q1 2015, attacks targeting an Akamai property were traced to a group of DDoS attack services found in the DDoS-for-hire market. These booter/stresser sites appear to make use of shared attack scripts found in underground forums.

At first glance, these attack attempts don't appear to be particularly dangerous, with the largest peaking at no more than 100 Gbps. But when compared with the volume from a year ago, we see a potentially dangerous trend emerging.

A year ago, peak attack traffic using these tactics typically measured 10-20 Gbps per second. The 100 Gbps attack we saw in Q1 2015 represents a significant jump, suggesting attackers have been developing ways to maximize impact. At the rate they are progressing, security researchers are concerned about what the attackers may be able to accomplish by this time next year.

Also troubling is the fact that employing the current attack techniques has not required much skill. As more advanced, potent tools enter the picture and available attack bandwidth increases, unskilled adversaries could become capable of much more damaging assaults.

2.1 / Methodology / First, it's important to explain where Akamai's findings come from. Every piece of technology used within our DDoS mitigation platform for these campaigns has correlating activity logs. By reviewing those logs, researchers were able to identify the types of attack techniques being used, as well as attack sizes and other campaign attributes. Research this past quarter points to the heavy use of booter and stresser sites.

2.2 / Booters and Stressers / The words booter and stresser have different etymologies. Booter comes from the online gaming world, where rival factions of players compete against each other, usually in very large multi-player online platforms.

These rivalries can escalate to the point where some players will use DDoS attacks against rivals in order to evict, or boot, them from the game. Malicious actors with a higher skill level built the attack tools into booter sites that offer a la carte DDoS attacks against any desired target. They eventually productized these DDoS attack sites. Some of these booter sites also provide gaming-related services, such as player IP or username discovery, which are then leveraged in the attacks.

The term stresser is a legal artifact used by the malicious actors behind these sites to make them appear to be legitimate. Users of these sites will usually be presented with a legal banner or verbiage on the website that clearly states that the services provided are for stress testing on authorized sites. This provides malicious actors with the appearance of operating within legal limits, since the attacks are directed by customers and not by site owners.

As stated in the Q4 2014 State of the Internet – Security report, there is a remarkable expansion of these sites in the underground DDoS-for-hire market. This proliferation has resulted in a large number of preconfigured and ready-to-go attack tools that often include easy-to-use interfaces and menus. Large DDoS campaigns that match the attack vectors featured at these sites represent a significant portion of observed attacks. Some of them exceed 100 Gbps.

2.3 / Top Attack Vectors / Figure 2-1 depicts the total number of individual attack vectors launched against the targeted Akamai property this quarter and matched attack types available on booter sites.

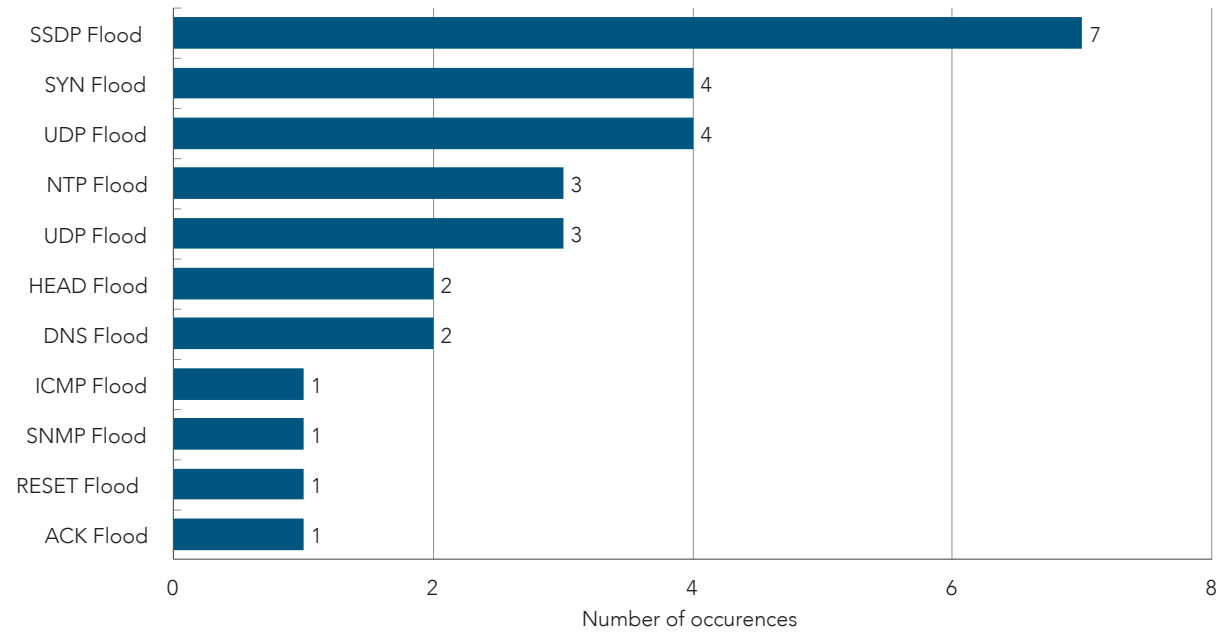


Figure 2-1: Attack occurrence by vector against the Akamai property during Q1 2015

2.3A / SSDP Reflection Attacks / The top attack vector, Simple Service Discovery Protocol (SSDP), comes as no surprise. SSDP reflection attacks have been common during the last few months, as discussed in the SSDP Reflection DDoS Attacks threat advisory. In Q4 2014, SSDP was the top reflection attack vector used against all targeted customers protected by Akamai, according to mitigation data. Not only is this attack easy for malicious actors to execute, but the number of vulnerable reflectors does not appear to be diminishing. There were millions of vulnerable reflectors when PLXsert first released its advisory.

The attack begins with a spoofed `M-SEARCH` request. For example, if an attacker wants to target a host on IP x.x.x.x, using a device on the Internet at IP y.y.y.y, the request will unwittingly force y.y.y.y to reflect the response data back to the target at x.x.x.x, as shown in Figure 2-2.

```
15:17:48.540126 IP x.x.x.x.80 > y.y.y.y1900: UDP, length 90
E..vc.....o...

....P.l.b..M-SEARCH * HTTP/1.1
Host:239.255.255.250:1900
ST:ssdp:all
Man:"ssdp:discover"
MX:3
```

Figure 2-2: A sample spoofed malicious query to an SSDP reflector responds back to the intended target

The source code we analyzed for this attack varies in the order and spacing of the search request and headers. Observed code variations also include two programming languages: Python and C. All variants include the same headers outlined by the UPnP Forum's device architecture documentation for a discovery request. Figure 2-3 shares the required headers and the request method, with descriptions.

Observation of these requests at the network edge are a good indicator of SSDP probe requests, if they appear to target a range of IPs within the network. If the requests appear repeatedly for the same target IP, and from the same unknown source, the target device could be part of a list of SSDP reflectors participating in DDoS attacks.

```
M-SEARCH(request method) *(any resource) HTTP/1.1(HTTP version)
HOST: 239.255.255.250:1900 (multicast address and port)
ST: ssdp:all (search for all devices and all services)
MAN: "ssdp:discover" (defines scope)
MX: 3 (delay response by 3 seconds)
```

Figure 2-3: Request method and header definitions as described by the UPnP Forum

2.3B / SYN FLOOD ATTACKS / Another common attack vector available on a typical booter/stresser site is the SYN flood attack. Among the list of vectors, the SYN flood attack appeared to be one of the more customized attacks. In fact, out of the SYN flood vectors encountered in Q1, two were close — but not exact — matches to the original SYN flood code available on booter sites. One SYN attack was not included in the list because it was too different from the variants depicted here. However, the majority of mitigated attacks against this Akamai property were booter-based.

Recent examination of available SYN flood code reveals at least four common variations. As seen in Figure 2-4, other than the differences in the first script's attack signature, it is difficult to tell them apart. All of the scripts spoof the source IP address based on the randomization logic of the code. One shows how the code can be easily modified to create any combination of TCP flags. These can also be randomized to change during the attack.

```
SYN flood variation 1
23:03:23.724420 IP x.x.x.x.1234 > x.x.x.x.80: Flags [S], seq 0, win
5840, length 0

SYN flood variation 2
23:04:13.330443 IP x.x.x.x.52599 > x.x.x.x.80: Flags [S], seq
3388669952, win 0, length 0

SYN flood variation 3
23:05:07.439638 IP x.x.x.x.50027 > x.x.x.x.80: Flags [S], seq
1616379904, win 0, length 0

SYN flood variation 4
23:05:58.522874 IP x.x.x.x.21304 > x.x.x.x.80: Flags [SEW], seq
943587328, win 0, length 0

SYN flood custom attacks endless variations
13:42:25.234850 IP x.x.x.x.1234 > x.x.x.x.80: Flags [SRP], seq 0,
win 5840, length 0
```

Figure 2-4: The sample SYN flood attack signatures, produced in the lab, were nearly identical

It seems that the four attack vectors share some of the same code. One obvious difference is the first SYN signature has a hard-coded source port of 1234 and destination port of 80 (HTTP). The other difference, seen in the fourth attack variation, includes three flags. The flags set are SYN, Explicit Congestion Notification (ECN), and Congestion Window Reduced (CWR). An expanded view of the flags can be seen when examining the attack traffic using tshark, as shown in Figure 2-5. A *1* indicates the flag is set.

```
Flags: 0x0c2 (SYN, ECN, CWR)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 1... .... = Congestion Window Reduced (CWR): Set
        .... .1.. .... = ECN-Echo: Set
        .... ..0. .... = Urgent: Not set
        .... ...0 .... = Acknowledgment: Not set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..1. = Syn: Set
```

Figure 2-5: The tshark output of the attack traffic identifies which flags were set

Malicious actors will sometimes modify attack code for their own use. This creates the observed variations and, at the same time, the impression that booter site x has more attacks available than another site. The possibility also exists for individuals to make minor modifications to attack code and rebrand the code for profit. In fact, except for the limitations of the first variation of the SYN attack, the other attacks are essentially the same.

2.3c / HEAD Flood Attacks / Among all the booter/stresser attack vectors discussed here, the HEAD flood was the only layer 7 DDoS attack vector used against the targeted Akamai site. This attack consists of a flood of HTTP HEAD requests sent to a server. The HEAD flood script used on booter sites also includes the ability to generate GET and POST requests. The attack signatures are shown in Figure 2-6.

```
GET
GET / HTTP/1.1
Host: target domain
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/31.0.1650.63 Safari/537.36
Connection: close
Accept-Encoding: gzip, deflate

HEAD
HEAD / HTTP/1.1
Host: target domain
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
FunWebProducts; .NET CLR 1.1.4322; PeoplePal 6.2)
Connection: close
Accept-Encoding: gzip, deflate

POST
POST / HTTP/1.1
Host: target domain
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
Gecko/2008120120 Blackbird/0.9991
Connection: close
Accept-Encoding: gzip, deflate
Content-Length: 8
Content-Type: application/x-www-form-urlencoded
```

Figure 2-6: Attack signatures generated from the HEAD flood attack in a lab environment

These scripts are also easily customizable. The most popular change would be to hard code the user-agent list within the script.

This script allows the attacker to leverage a list of open proxy servers. This means that the true IP of the attack source is hidden. This results in more difficult attack attribution to individuals or booter sites.

2.4 / ATTACK TIMELINE / Figure 2-7 shows the Q1 attack activity as it unfolded. Things started quietly enough, with attackers fiddling with different vectors. In February, a spike in attack bandwidth occurred as malicious actors experimented with a wider array of vectors. Towards the end of March, the use of HEAD floods can be noted with a visible drop in bandwidth.

The layer 7 HEAD flood DDoS attacks rely on overwhelming a specific target host site with HTTP requests. In this case, it appears that malicious actors tried to use more targeted attacks after finding that volumetric attacks yielded no result.

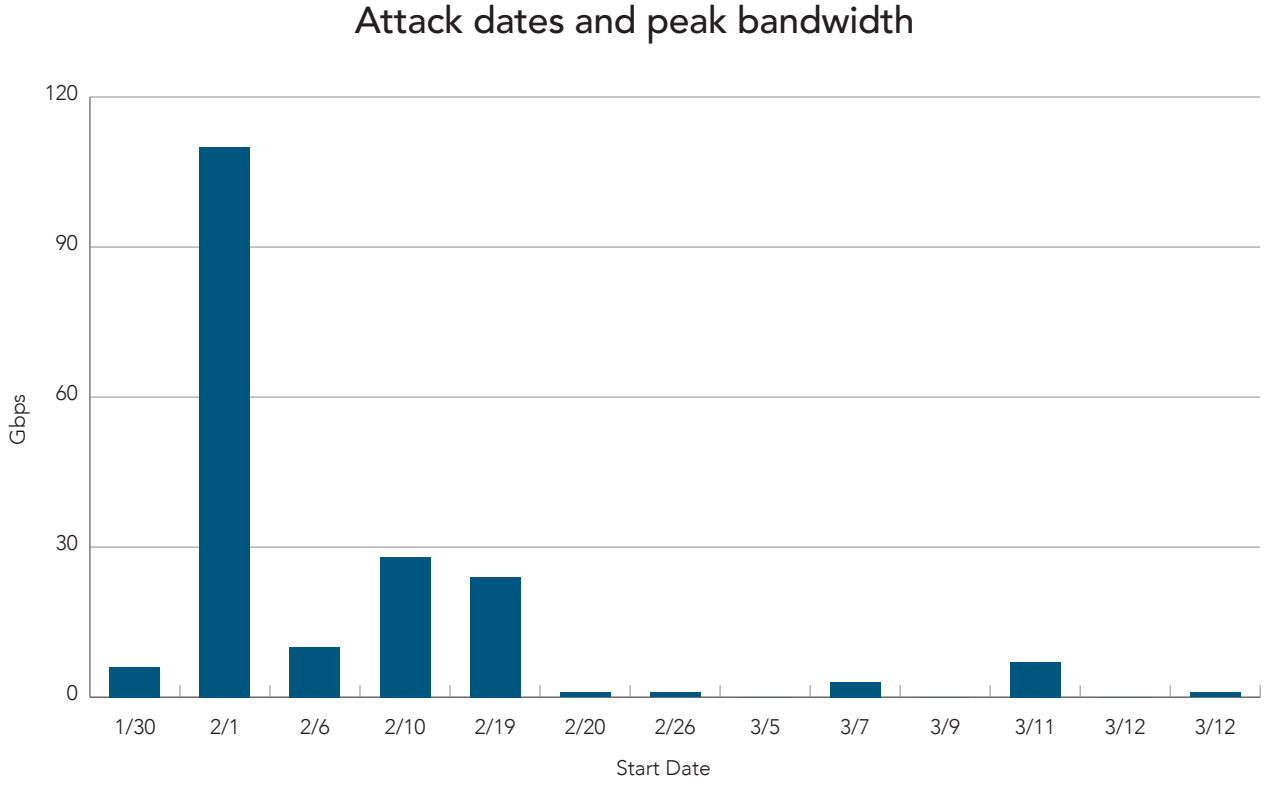


Figure 2-7: Q1 2015 timeline of booter/stresser attacks against Akamai

For the attack on February 1, the bandwidth peaked at 107 Gbps and 29 Mpps. The attack appeared to contain the full suite of booter vectors, including a SYN flood, UDP flood, UDP fragment flood, RESET flood, DNS flood, ACK flood, NTP flood and an SSDP flood. The distribution of the attack is shown in Figure 2-8 and Figure 2-9. Most of the traffic was sourced from Europe. In this case, our Frankfurt scrubbing center absorbed most of the attack.

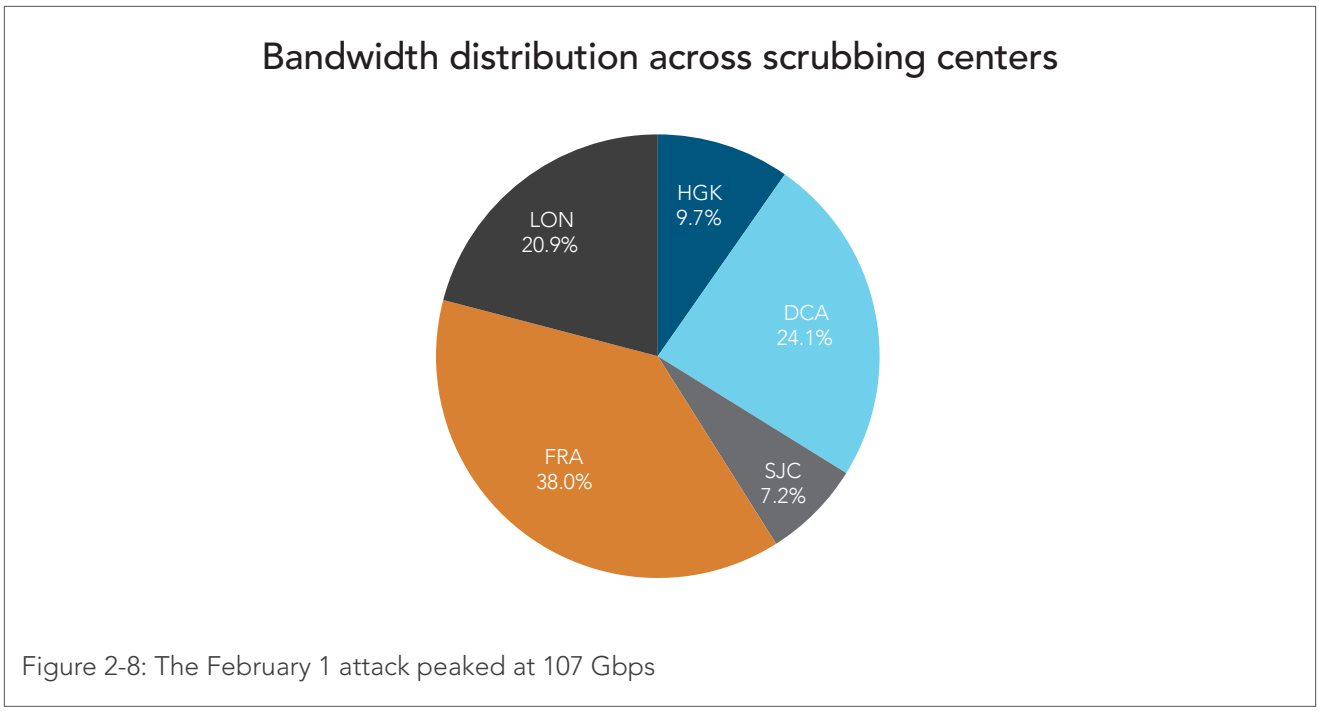


Figure 2-8: The February 1 attack peaked at 107 Gbps

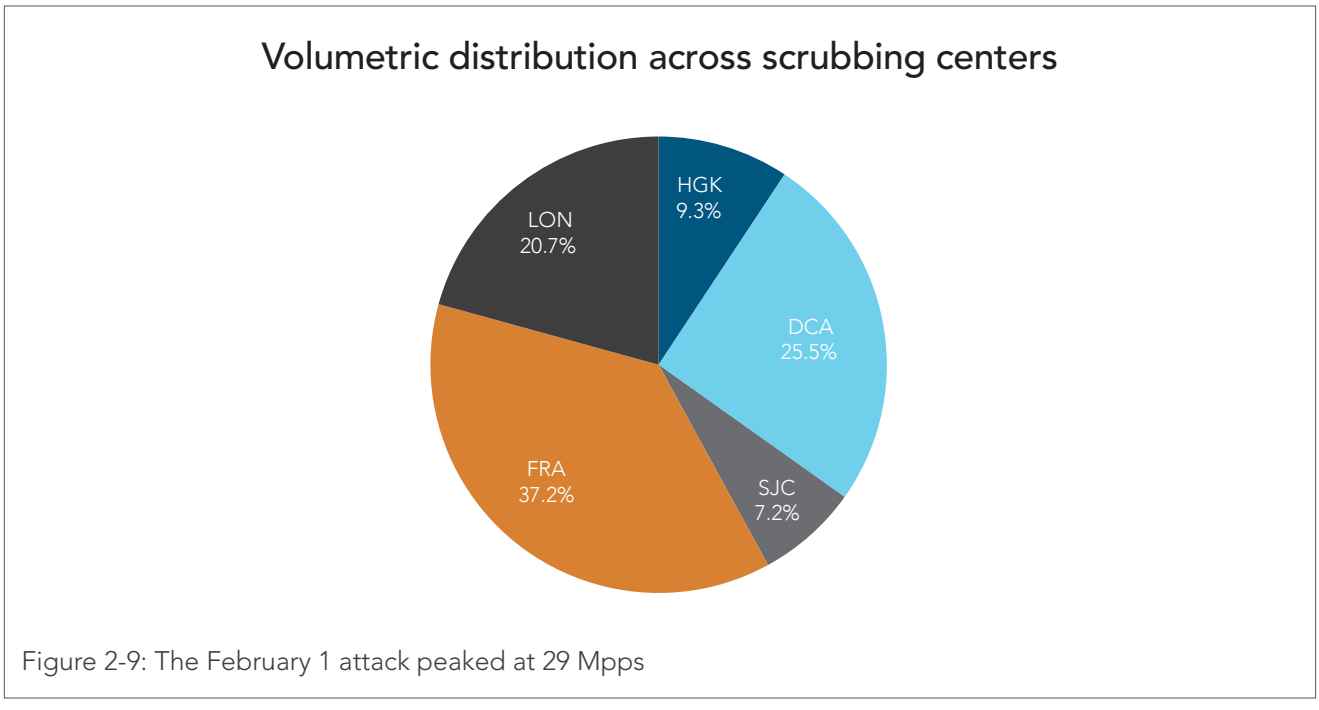


Figure 2-9: The February 1 attack peaked at 29 Mpps

2.5 / Summary / The menu of easy-to-use attack vectors found in the DDoS-for-hire market can make it tempting to dismiss the effectiveness of attackers who use them. While users of these scripts are often branded as *script kiddies*, however, it does not dilute the tools' potential to bring down target sites.

Users of these booter/stresser sites have shown themselves to be clever and persistent, experimenting and tweaking their tactics to get the most harmful results. Attacks originating from these tools have been on the rise. With new reflection attack methods being added continually, such as SSDP floods, the potential damage from these is expected to continue increasing over time. Meanwhile, the tools in these attack kits are being modified and improved constantly. This combination of technological improvement and attacker persistence will be monitored and taken very seriously by Akamai research teams.

[SECTION]3 CASE STUDY Security Implications for IPv6

Widespread expansion of the Internet has nearly exhausted the supply of available Internet Protocol version 4 (IPv4) addresses. This sparked the creation of IPv6, which has several benefits. Besides the extra address space, IPv6 includes some fundamental changes that could have large security implications.

3.1 / IPv6 Security Vulnerabilities / PLXsert gathered a collection of tools and attack vectors developed and used by researchers and malicious actors. Researchers found the following elements driving IPv6 attack vectors:

- Abuse of transitional technologies to bypass security controls
- Use of IPv6 protocol against applications and services that are IPv6 enabled, bypassing IPv4 security controls
- Modification of IPv6 protocol structure, aiming to bypass IPv6 IPS, IDS and firewall technologies
- Adaptation of application layer DDoS attacks to work over IPv6
- Adaptation of exploitation frameworks to work with the IPv6 protocol
- Purpose-built denial of service tools and techniques based solely on the IPv6 protocol architecture

As IPv6 adoption progresses, several researchers have found possible weak points and vulnerabilities in its various implementations. Most of these have been addressed by either deprecating features of the protocol itself — which was the action taken to address the Type 0 Routing Header extension (RH0), or by introducing preventive measures such as IPv6 Router Advertisement Guard (RA Guard).

3.1A / Host Identification / IPv6 presents new challenges for would-be attackers and defenders due to the number of possible IPv6 addresses. For attackers, it will be more difficult to identify hosts as quickly as with IPv4 addresses and makes them more reliant on host naming services, such as DNS and multicast DNS (mDNS). From a defender standpoint, it is also more difficult to track individual hosts, as a single small network prefix can easily generate enough unique addresses to quickly exhaust a system's memory with its massive address space.

There are, however, open source tools and techniques that allow the discovery of live hosts. One popular tool is called Alive6 and is part of the THC IPv6 attack tool. This tool provides the capability of finding active hosts in network segments using Internet Control Message Protocol version 6 (ICMPv6) packets and host enumeration. While this is effective it is also time consuming.

In networks that use dual stack (IPv4/IPv6) implementations, it is also possible to identify neighbors on the IPv4 network using the IPv4 Address Resolution Protocol (ARP). Using the information gathered from this protocol, neighboring MAC addresses can be converted into usable IPv6 local-link, and in some cases, global-link addresses.

3.1B / IPv6 Transition Vulnerabilities / There have been multiple vulnerabilities found in the IPv6 protocol stack, dating from 2002. These vulnerabilities highlight the difficulty with the transition and exposure of multiple devices and applications.

The issue is that many administrators and users overlook IPv6 networking that is enabled by default; services and protocols can be exposed even if security measures were in place for IPv4 implementations, as shown in Figure 3-1. There are also issues with the use of transitional technologies where IPv6 traffic can effectively bypass security filtering by using tunneling protocols such as Teredo. This presents a scenario where malicious actors could attempt leverage these oversights to bypass protections.

For example, Iptables in Linux doesn't work on IPv6 unless expressly configured to do so, while some older versions of the Windows OS firewall do not block IPv6 traffic by default. Additionally, older versions of Snort only support IPv6 if the software was compiled with the `--enable-ipv6` flag. Depending on software distribution channels, feature coverage, and frequency of updates, this might leave some systems unable to inspect, alert on, or drop IPv6 traffic.

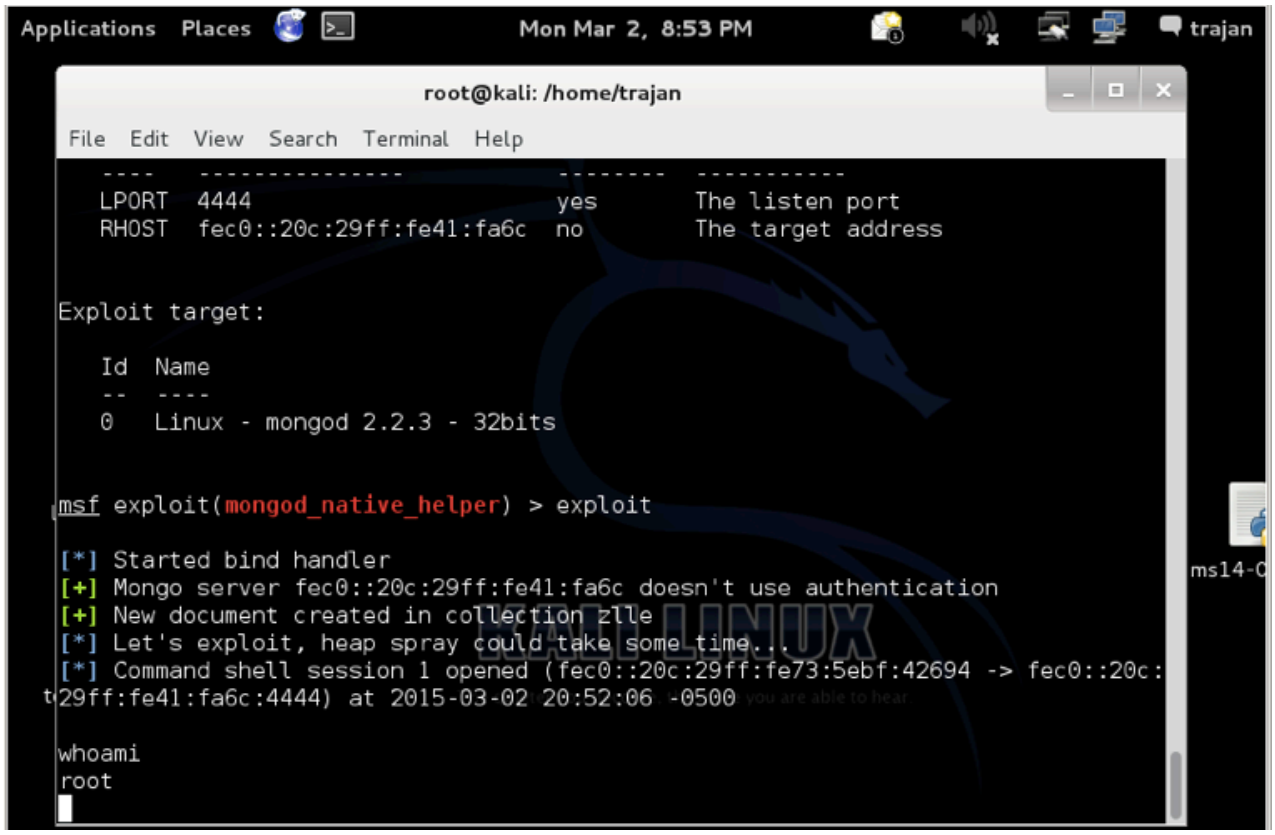


Figure 3-1: Iptables IPv4 was configured to block traffic to the service, but was vulnerable when executed on IPv6-enabled hosts

This ability to bypass firewalls and other security measures opens the potential for old and well-mitigated threats to resurface. Over the past few years, researchers and system administrators have worked together to reduce the effectiveness of techniques that have been popular with attackers, including blocking troublesome services. In most cases, these services are closed to the Internet or filtered via firewalls or IDS/IPS systems. These services may listen on both IPv4 and IPv6, but in many configurations, only IPv4 traffic would be inspected and/or blocked. These old attack vectors may be retooled for operation on IPv6 networks. Metasploit and Slowloris attack kits have already been updated with IPv6 support. There is also the potential for new attack vectors that have been introduced within IPv6 itself.

3.2 / IPv6 Attack Vectors / As part of PLXsert's research, laboratory environments were created internally and on some of the leading cloud provider's platforms. Some of these platforms deployed IPv6 functionality by default, while others required IPv6 to be explicitly enabled. Abuse of IPv4 protected services and systems was possible in most tests using the IPv6 stack. Researchers were also able to leverage dual stack implementations to help gather more information about the IPv6 networks.

3.2A / Reflection / Using a local laboratory environment, researchers were able to use standard UDP reflection techniques against both CHARGEN and NTP services over IPv6, where the packets would normally be ingress-filtered on their way to the reflectors by Iptables (See Figure 3-2). The lack of IPv6 support in the filtering layer allowed access to the services without issue, as expected.

```
01:18:24.069558 IP6 2600:ABC:ABC:X:X:X:X.123 > 2601:3:ABC:AB-
C:Y:Y:Y:Y.31337: NTPv4, Server, length 48
H............b....8.4&.<......<...P..&......4J........{zi.8?.$.
..........KY....-/........EP..............2.
```

Figure 3-2: NTP reflection successfully targeted an IPv6 machine in our lab behind a shared router

3.2B / Spoofing and Hijacking / Many ISPs and hosting providers prevent packet spoofing by deploying anti-spoofing technologies. Due to the nature of Network Address Translation (NAT) networks, a single IP would normally be associated with multiple hosts behind it, making spoofing easier to detect and therefore harder to achieve. To an extent, this is still true with IPv6, but the address space and routing changes allow for a huge spoofable/hijackable address space that could be leveraged by attackers.

A single end-user IP range assigned by an ISP will typically be a /64. This gives an attacker roughly 18 quintillion spoofable/hijackable addresses. Since end-to-end routing is supported and most of those addresses will go unused, even a single machine on a network could easily send traffic that appears to be from millions upon millions of legitimate-looking hosts, as shown in Figure 3-3.

```
05:47:03.700858 IP6 2601:3:ABC:ABC:dead:beef:15:dead.31337 >
2600:ABC:ABC:X:X:X:X.123: NTPv4, Client, length 48
`....8.3&......4........&.<......<...P..zi.{.8SE#.
.........................W.G.............W.L..
05:47:03.701032 IP6 2600:ABC:ABC:X:X:X:X.123 > 2601:3:ABC:ABC:dead-
:beef:15:dead.31337: NTPv4, Server, length 48
k....8.@&.<......<...P..&......4.........{zi.8.e$.
........p.KY....g/.k3...W.L.....W.k.....W.u..
```

Figure 3-3: Spoofed traffic was successfully routed to an IPv6 host via an ISP

A botnet of home computers vulnerable to IPv6 spoofing could generate massive amounts of unique-looking host addresses, far beyond what is possible today using IPv4. Likewise, these same devices can be assigned globally-identifiable addresses that, in effect, bypass NAT. A compromised machine could be leveraged as a malicious server with large numbers of unique addresses for malware distribution, and/or as part of the command-and-control infrastructure.

Another issue arises with blocking. Since quintillions of devices could potentially share a legitimate /64 prefix, one malicious actor could effectively cause massive outages for huge numbers of users if blocking is not accurately applied. With IPv4, blocking a single source might result in a worst-case scenario of a high traffic proxy with tens of thousands of users being blocked. Applying these same techniques to IPv6 could lead to blocking a single prefix and potentially denying service to hundreds of millions of non-malicious users.

3.2C / Local-Link Attacks / PLXsert performed several tests on popular cloud provider networks. It was discovered that one provider did not have Rogue Router Advertisement (RRA) protections in place. Researchers were able to craft RRA packets in Scapy and flood the testing machines over unicast with malformed routing information. These requests directed the targeted machine to use the attacking server as its first hop in the default route. This caused the targeted machine to stop communicating over its global link interface, effectively DoSing its end users. This technique was effective in networks where local-link addresses are shared with neighbors, and protections against RRA are not in place. This technique could also be used to achieve a man-in-the-middle attack.

3.2D / Dual Stacks and IPv6 Address Space / The large address space of IPv6 can be misleading in terms of security. Part of this misconception is the apparent impossibility of scanning such a large address space. However, utilizing IPv4 protocols such as ARP on dual-stack systems, researchers were able to discover neighboring server MAC addresses for the associated IPv4/24 on various cloud platforms, as shown in Figure 3-4. Using this information, researchers could reliably convert those neighboring MAC addresses into IPv6 local-link and, in some cases, global link addresses, as shown in Figure 3-5. These networks, however, did route local-link traffic to services, which could be leveraged to bypass firewall and IDS/IPS measures against the host.

```
f2:3c:91:df:d8:55     64 bytes from ABC:123::f03c:91ff:fedf:d855: icmp_seq=1 ttl=64 time=0.373 ms
f2:3c:91:df:c5:39     64 bytes from ABC:123::f03c:91ff:fedf:c539: icmp_seq=1 ttl=64 time=1.39 ms
f2:3c:91:df:c4:4d     64 bytes from ABC:123::f03c:91ff:fedf:c44d: icmp_seq=1 ttl=64 time=0.516 ms
f2:3c:91:df:b3:90     64 bytes from ABC:123::f03c:91ff:fedf:b390: icmp_seq=1 ttl=64 time=0.404 ms
f2:3c:91:df:a6:5a     64 bytes from ABC:123::f03c:91ff:fedf:a65a: icmp_seq=1 ttl=64 time=1.02 ms
f2:3c:91:df:81:2c     64 bytes from ABC:123::f03c:91ff:fedf:812c: icmp_seq=1 ttl=64 time=0.726 ms
f2:3c:91:df:7f:3f     64 bytes from ABC:123::f03c:91ff:fedf:7f3f: icmp_seq=1 ttl=64 time=0.391 ms
f2:3c:91:df:7a:cb     64 bytes from ABC:123::f03c:91ff:fedf:7acb: icmp_seq=1 ttl=64 time=0.427 ms
f2:3c:91:df:72:83     64 bytes from ABC:123::f03c:91ff:fedf:7283: icmp_seq=1 ttl=64 time=0.489 ms
f2:3c:91:df:6c:7b     64 bytes from ABC:123::f03c:91ff:fedf:6c7b: icmp_seq=1 ttl=64 time=0.420 ms
f2:3c:91:df:66:78     64 bytes from ABC:123::f03c:91ff:fedf:6678: icmp_seq=1 ttl=64 time=0.473 ms
f2:3c:91:df:42:7d     64 bytes from ABC:123::f03c:91ff:fedf:427d: icmp_seq=1 ttl=64 time=0.597 ms
f2:3c:91:df:10:d6     64 bytes from ABC:123::f03c:91ff:fedf:10d6: icmp_seq=1 ttl=64 time=0.436 ms
f2:3c:91:db:db:f1     64 bytes from ABC:123::f03c:91ff:fedb:dbf1: icmp_seq=1 ttl=64 time=0.495 ms
f2:3c:91:db:cf:d2     64 bytes from ABC:123::f03c:91ff:fedb:cfd2: icmp_seq=1 ttl=64 time=0.385 ms
f2:3c:91:db:b3:39     64 bytes from ABC:123::f03c:91ff:fedb:b339: icmp_seq=1 ttl=64 time=0.279 ms
f2:3c:91:db:b0:87     64 bytes from ABC:123::f03c:91ff:fedb:b087: icmp_seq=1 ttl=64 time=0.423 ms
f2:3c:91:db:7a:60     64 bytes from ABC:123::f03c:91ff:fedb:7a60: icmp_seq=1 ttl=64 time=0.495 ms
f2:3c:91:db:4d:79     64 bytes from ABC:123::f03c:91ff:fedb:4d79: icmp_seq=1 ttl=64 time=0.477 ms
```

Figure 3-4: MAC address farmed from ARP responses reply to pings when converted to IPv6 addresses

```
#!/usr/bin/python2

import sys
import bitarray
import binascii

mac=sys.argv[1]
mac=mac.split(":")

local_prefix="fe80::"
global_prefix="ABC:123::"

cnt=0
new_mac=[]
for octet in mac:
    cnt=cnt+1
    if cnt == 1:
        bits = bin(int(octet, 16))[2:]
        while len(bits) < 8:
            bits = "0"+bits
        flip = bits[6]
        if int(flip) == 1:
           flip = 0
```

```
        else:
           flip = 1
        bits = bits[:6]+str(flip)+bits[7:]
        hex = bitarray.bitarray(bits)
        hex =  binascii.hexlify(hex)
        octet = str(hex)
    if cnt == 4:
        new_mac.append('ff')
        new_mac.append('fe')
    new_mac.append(octet)

ipv6 = ""
cnt = 0q
for octet in new_mac:
    ipv6=ipv6+octet
    cnt=cnt+1
    if cnt % 2 == 0:
        ipv6=ipv6+":"

print local_prefix+ipv6[:-1]
print global_prefix+ipv6[:-1]
```

Figure 3-5: Python script to quickly convert MAC address to IPv6 addresses

Using these same methods, attackers could possibly identify patterns in how MAC addresses are generated for virtual machines as they're deployed, and potentially predict which addresses are most likely to occur. Since Stateless Address Autoconfiguration (SLAAC) generates its resulting IPv6 address based on slight modification of the initial MAC address, this greatly reduces the amount of scanning needed, while also increasing the likelihood of finding real targets. Using less predictable methods for address generation will help defeat this technique. Some cloud services providers are already using more advanced global link addressing schemes in the wild.

3.3 / Summary / A new set of risks and challenges associated with the transition to IPv6 are now affecting cloud providers as well as home and corporate network owners. Many IPv4 DDoS attacks can be replicated using IPv6 protocols. Some new attack vectors are directly related to the IPv6 architecture. Many of the features of IPv6 could enable attackers to bypass IPv4-based protections, creating a larger, and possibly more effective DDoS attack surface. These negative effects have remained largely undiscovered or unreported, perhaps due to the slow global adoption of IPv6.

It is important to raise awareness of the potential risks that end users and corporations unknowingly face when deploying IPv6 technology without proper training and security considerations. IPv6 DDoS is not yet a common occurrence, but there are indications that malicious actors are already testing and researching IPv6 DDoS attack methods. These may prove to be effective in combination with transition technologies and dual stack architectures.

Finally, the impending addition of billions of Internet-enabled devices will force IPv6 to be the principal addressing protocol on the Internet in the future. It is imperative for the security community to be ready to address newly-discovered security challenges associated with IPv6. Additional information regarding IPv6 and the associated security challenges can be found at www.akamai.com/ipv6.

A Brief History of Internet Addressing and the Benefits of IPv6

IPv4 is the main protocol version currently used for identification and routing over the Internet. Its address space is relatively small and facing complete exhaustion. To slow total IPv4 address exhaustion, several techniques and technologies such as NAT and Classless Inter-Domain Routing have been implemented.

Widespread adoption of devices commonly referred to as Small Office/Home Office (SOHO) and Customer-Premise Equipment (CPE), such as modems and routers, were a strong driver of these technologies. They've allowed homes and businesses to connect multiple devices to the Internet that wouldn't have been possible otherwise. These technologies, however, will not prevent the exhaustion of all available IPv4 addresses.

A new version of the Internet Protocol was proposed by the Internet Engineering Task Force to simplify networking for Internet-connected devices and to address IP exhaustion. This new version, IPv6, was designed to completely replace IPv4 and provide transitional technologies to migrate from IPv4 addresses and architectures. As with its predecessor, the objective is to provide identification and routing for networked devices on public and private networks.

IPv6 Address Space / IPv6 allows for an almost unimaginably large address space; it is 79 octillion (that's 79 billion billion billion) times larger than the IPv4 address space. If a scanner was able to test 100,000 addresses per-second, a single standard-sized address block allocation given to an ISP customer, referred to as a /64, would take more than 5.5 million years to fully scan. Multiply that by 18 quintillion and you would have the entire address

space possible — about 340 trillion trillion trillion unique addresses. This large address space will be instrumental in preventing the possibility of IPv6 address exhaustion well into the future. However, IPv6 is not interoperable with IPv4 without additional software and configuration. This has been addressed by the development of several transitional technologies such as NAT64, 6to4, 6in4 and Teredo — all of which allow IPv4 and IPv6 interoperation.

IPv6 Security, Routing and Networking / IPv6 provides other benefits besides the large address space. This includes security, routing, and general networking changes.

Routing optimizations are expected to reduce the size of global routing tables and to move some features, such as packet fragmentation and check sums, off routers and onto clients. Network mobility was designed to allow an entire network to migrate without extensive renumbering and give rise to roaming networks. Support for globally unique addresses for each connected client will allow end-to-end connections between devices, simplifying routing and eliminating the need for NAT.

Multicast functionality has been designed with the goal of reducing the number of additional protocols needed to utilize it, simplifying deployment. These features will help optimize application availability and content delivery on local networks, but they will also allow multicast traffic to travel across networks that aren't directly connected.

IP Security (IPsec), a security protocol purposely designed for IPv6 will enable more advanced security features such as encryption and authentication at the IP level. These features will help ensure data cannot be read in transit, that it is from its advertised source, and that it was not modified in transit.

These features could allow larger secure private networks to be interconnected and routed over the Internet without implementing systems such as Virtual Private Networks (VPNs).

Addressing standards, such as SLAAC, allow hosts to configure themselves when connected to IPv6 networks. This will simplify network configuration and allow devices to find each other without a central addressing authority, using multicast networking over local links. There is also the possibility to enable privacy extensions to protect endpoints.

Some IPv6 features, such as multicasting, are similar to IPv4 implementations. Others, such as addressing privacy extensions, are new and have no IPv4 equivalent. IPv6 can use SLAAC at the local link to find its gateways, neighbors, and routers, but can also receive additional configuration via the Dynamic Host Configuration Protocol version six (DHCPv6). Networks and clients can also utilize Cryptographically Generated Addresses (CGA) for more secure and advanced addressing needs.

It's important to notice that the full implementation of IPv6 across the Internet and on every device could technically allow every device the ability to have an end-to-end communication with any other device on the Internet.

Since addressing mechanisms such as NAT are no longer needed, this could expose such devices and their services to the outside world. These changes could effectively allow malicious actors the capability to directly access devices that had been less accessible with IPv4.

As the transition to IPv6 continues, security researchers and malicious actors alike have begun exploring possible vulnerabilities and exploitation techniques. IPv6 has seen adoption across all major operating systems and is now seeing deployment across several large ISPs for consumer use. The U.S. government mandated IPv6 adoption across their networks and services as of September 2012.

Many legacy applications do not bind to IPv6 addresses, making it difficult to support backwards compatibility and interoperation. This has slowed the drive for exploitation of IPv6. That status is likely to change, as more ISPs, device manufacturers and software developers move their products into the IPv6 space. It is now common to see IPv6 local link addresses in home networks, since the latest operating systems implement IPv6. Many websites and organizations with presence on the Internet have implemented it as well; some unknowingly.

[SECTION]4 CRUEL (SQL) INTENTIONS

An analysis of malicious intentions behind real world SQL injection attacks

According to the WASC Threat Classification project, SQL injection is an attack technique used to exploit applications that will construct SQL statements from adversary-supplied input. When successful, the attacker is able to change the logic of SQL statements executed against the database.

Even though SQL injection was mentioned as early as 1998, it still plagues many modern web applications and continues to top industry lists such as the OWASP Top 10 and the CWE/SANS Top 25.

Akamai's Threat Research team set out to develop a technique to categorize attacks by analyzing individual attack payloads and determining the intention behind each one. The team analyzed SQL injection attacks based on data from Akamai's Kona Site Defender web application firewall (WAF). The data included 8,425,489 SQL injection attacks that targeted more than 2,000 unique Akamai customer web applications, during a period of seven days.

4.1 / SQL Injection Attack Types / The original SQL injection exploitation techniques — which attempt to retrieve data from a backend database — are still being used, and new exploitation methods have evolved. In addition, automated injection tools now streamline and simplify some of the more complex methods.

The goals of SQL injection attacks include the following:

4.1A / SQL Injection Probing and Injection Testing / As a first step, malicious actors typically perform an assessment of the web application to determine if it is vulnerable to SQL injection. As a part of the process, the malicious actor will traverse the application, locate all entry points and send certain string sequences to sense whether the application is vulnerable.

Same payloads include sequences of SQL sensitive characters such as apostrophe (') or semicolon (;), etc. Modern approaches to probing make use of techniques commonly associated with blind SQL injection, such as forming Boolean conditions - `AND 1=1`, `AND 1=0`, or using timed attacks with `WAITFOR` or `sleep()` functions. More information on Blind SQL Injection techniques can be found on OWASP.

4.1B / Environment Probing and Reconnaissance / After concluding that the application is vulnerable to SQL injection, the malicious actor will take the attack a step further and try to learn the type and structure of the database, its tables, columns, users and permissions.

4.1C / Database Content Retrieval / With a clear understanding of the type and structure of the database and its tables, the malicious actor can start retrieving contents remotely via techniques such as data extraction using `UNION SELECT` statements or by using blind SQL injection techniques (Boolean expressions).

4.1D / Login Mechanism Bypass and Privilege Escalation / Today, the majority of web application login mechanisms use a back-end SQL query to check whether the given credentials are correct, and if the user is allowed to login to the application. This provides malicious actors with a simple, yet extremely effective method to bypass login mechanisms by using SQL injection attacks.

A common and classic payload would be to send the payload `' OR 1=1` as the user name. When the user name of the administrator (e.g., admin), is known, the malicious actor could attempt to elevate privileges by logging in with the user name: `admin or 1=1--`.

If the application uses that input to complete a query such as:

```
SELECT * FROM user_tbl WHERE user_name='$_POST{username}'
AND password='$_POST{password}'
```

then the final constructed query will be:

```
SELECT * FROM user_tbl WHERE user_name='admin' or
1=1--' AND password=''.
```

This query allows the adversary to modify the syntactic structure of an application's intended query so that it always returns the row for the administrator user, whether or not the adversary knows the administrator's password.

4.1E / Business Logic Subversion / Web application functions that rely on backend SQL queries to fetch data could be subverted to perform unexpected actions, such as presenting data that was not supposed to be visible.

For example, let's assume we have a URL (`/getDocument.php`) that returns the contents of a document, based on the name of the document given in a parameter called `docTitle`:

```
GET /getDocument.php?docTitle=SALARY_TABLE HTTP/1.1
```

A malicious actor could attempt to subvert the logic by issuing the following request:

```
docTitle=SALARY_TABLE OR 1=1
```

4.1F / Credential Theft / SQL injection attacks grant remote actors access to the database and its tables. In some cases, where proper permissions were not applied, they may access the user tables or harvest user names and passwords.

4.1G / Data and File Exfiltration / Certain databases provide facilities and functions to access local files on the database server. In such cases, malicious actors may use SQL injection attacks to extract local files.

4.1H / Denial of Service (DoS) / There are numerous ways SQL injection attacks can generate DoS attacks. For example, SQL injection can overload the database with data, shut down the database or submit too many queries that hang and consume all resources.

4.1I / Data Corruption / If proper permissions are not set, malicious actors may use SQL injection to corrupt data. This might be achieved by data modification such as `UPDATE` statements, data deletion with `DELETE` statements, or using statements to drop tables or even entire databases.

4.1J / Malicious File Upload / Certain SQL databases provide utilities and functions to export query results as a local file. In such cases, a malicious actor could use a SQL injection to write malicious content into local files.

4.1K / Website Defacement and Malicious Content Injection / Many modern web applications store their web content in a database — for example, a content management system (CMS) or blogging platform. Using a SQL injection attack, a malicious actor could write content into relevant tables, which could then be presented to users as legitimate web content. Such data may include link spamming or even links to third-party malware code.

4.1L / Remote Command Execution / This category is perhaps the most pernicious of web application attacks: the ability to execute remote commands on the database server. Not many databases provide facilities for executing shell commands in the default configuration, so this vector is not as prevalent as others.

4.2 / Anatomy of Attacks / The first piece of information we extracted was the distribution between SQL injection attacks over clear HTTP (unencrypted) vs. HTTPS (encrypted). More than 96 percent of the suspected attacks were not encrypted, as shown in Figure 4-1.

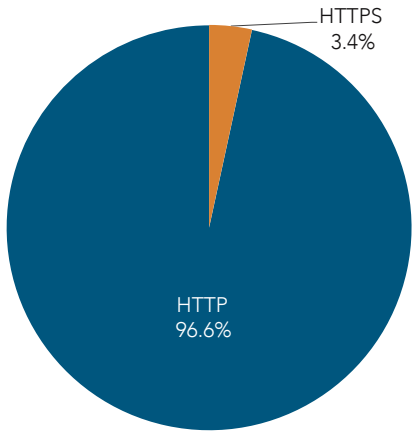


Figure 4-1: The majority of SQL injection attack attempts during the study period were not encrypted

The only SQL injection attack types that we did not observe during the research period were business logic subversion and malicious file upload. Of the 11 SQL injection attack types discussed in this case study, three attack vectors were responsible for more than 98 percent of the detected attack attempts during the study period, as shown in Figure 4-2.

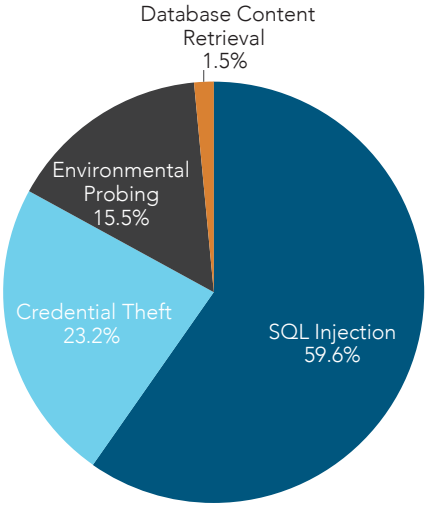


Figure 4-2: SQL injection probing, credential theft and environment probing were the most common attack types during the study period (rounded to the nearest percent). Content Injection, Data Corruption, File Exfiltration, Login Bypass and Remote Command Execution combined account for only 0.2 percent of SQL injection attacks during the research period.

4.2A / SQL Injection Probing and Injection Testing / As a first step, malicious actors will usually perform an assessment of all the entry points of the web application to see if any are vulnerable to SQL injection. As a part of the process, the attacker will send a wide range of characters that have syntactic meaning in SQL, such as semicolon (;) and apostrophe (‘), as well as blind-injection related Boolean sequences or timed queries. This will naturally result in large volumes of traffic. Nearly 60 percent of HTTP transactions (5,021,240) were attributed to these probing attempts.

If we consider the fact that every SQL injection attack is preceded by probing or testing for feasibility of injection, then we can extract 5,021,240 transactions from the statistics below, and analyze the distribution of all other payload classes out of the new total of 3,404,249.

4.2B / Environment Probing and Reconnaissance / Since most targeted SQL injection attacks require the malicious actor to probe the database environment and extract its table names and columns, it’s no surprise that we saw 1,306,681 malicious transactions (15.5 percent of the total or 38.3 percent of the non-probe payloads) attempting to carry out such actions. The most common payloads used the following techniques:

- Extraction of information from the MYSQL `INFORMATION_SCHEMA` table `UNION SELECT group_concat(COLUMN_NAME) FROM INFORMATION_SCHEMA.COLUMNS—`
- Extraction of database environmental variables. A common example is the use of the MS-SQL scalar configuration functions (`@@`) such as `@@VERSION`, which returns system and build information for the current installation of SQL Server.

4.2C / Database Content Retrieval / In total, 129,814 malicious transactions (1.54 percent or 3.8 percent of the non-probe payloads) were related to content retrieval unrelated to user credential theft. The main approach was through SQL `UNION SELECT` statements.

4.2D / Login Bypass / Of the malicious HTTP transactions, 5,467 unique attacks (0.06 percent or 0.16 percent of the non-probe payloads) contained payloads which might bypass the login mechanisms of the targeted web applications.

4.2E / Credential Theft / There were 1,950,749 attempts to steal user credentials through sql injection attacks. This represented 23.15 percent (or 57.3 percent of the non-probe payloads) of the malicious transactions. These attacks included attempts to retrieve data from tables and views such as:

- `mysql.user` (MYSQL)
- `master.syslogins` (MS-SQL)
- `master.dbo.sysxlogins` (MS-SQL)
- `ALL_USERS` (Oracle)

While this category can be considered a subset of content retrieval, we felt it was unique and large enough to merit its own focus. It appears that the main target for data extraction is user credential theft.

4.2F / Data File Exfiltration / During the period of this research, only 24 malicious requests (0.0003 percent or 0.001 percent of the non-probe payloads) were attributed directly to file extraction attempts from the database server. All requests tried to access the `/etc/passwd` file. We assume that these are automated requests made by vulnerability scanning tools.

4.2G / Denial of Service / Only 326 malicious requests (0.0039 percent or 0.010 percent of the non-probe payloads) attempted to overload or shut down the database server. The main two vectors were:

- Excessive use of timed queries with time intervals much higher than expected when timing is used for blind SQL injection.
- Attempts to use the MS-SQL `SHUTDOWN` management command

4.2H / Data Corruption / In this analysis, there were 2,238 unique HTTP transactions (0.02 percent or 0.066 percent of the non-probe payloads) with SQL injection payloads that attempted to corrupt database information. These payloads included:

- `DROP` statements
- `DELETE` statements
- `TRUNCATE` statements

A simple, logical explanation for this low rate is the fact that the data could be worth more to the malicious actors intact, than it would be if destroyed. In addition, most of the automated SQL injection scanning tools we observed do not corrupt the data.

4.2I / Website Defacement and Content Injection / This category was not very popular during the research period but still yielded 8,156 unique malicious transactions (0.09 percent or 0.24 percent of the non-probe payloads) that attempted to inject content into the database, mainly using the SQL `INSERT` and `UPDATE` statements.

Among the most popular content injection attempts were link spamming attempts, which tried to inject HTML links and content pointing to certain blogging sites, as shown in Figure 4-3.

```
...id=someValue';declare @c cursor ; declare @d varchar(4000);set
@c=cursor for select 'update ['+TABLE_NAME+'] set ['+COLUMN_
NAME+']=['+COLUMN_NAME+']+case ABS(CHECKSUM(NewId()))%7 when 0 then
''''+char(60)+''div style="display:none"''+char(62)+''click here
''+char(60)+''a href="http:''+char(47)+char(47)+''community.some.
site''+char(47)+''someValue''
```

Figure 4-3: An example content injection query string (URL decoded)

4.2J / Remote Command Execution / In this category, we observed 794 malicious transactions (0.009 percent or 0.023 percent of the non-probe payloads). The majority attempted to use the MS-SQL extended stored procedure `xp_cmdshell`, which enables remote users to run Windows shell commands on the database server. The use of this stored procedure may be attributed to automated tools, rather than manual hacking attempts. The commands that were executed using this stored procedure were (in order of occurrence):

- Windows `ver` (version) shell command
- Windows `dir` shell command
- `ping` command
- `nslookup` command

There were no detected attempts to perform more sophisticated commands, which strengthens the assumption that these were automated scanning/probing attempts.

4.3 / Summary / This case study shows malicious actors use a variety of SQL injection techniques to accomplish many different tasks. The effects of these malicious queries can extend well beyond simple data exfiltration, potentially causing more damage than a data breach.

When generating threat models for web applications, never assume that SQL injection attacks lead only to data theft. These attacks can be used to elevate privileges, execute commands, infect or corrupt data, deny service, or otherwise harm your business.

[SECTION]5 EMERGING THREAT Website Defacements and Domain Hijacking

In Q1 2015, Akamai tracked and provided defensive measures to mitigate mass website defacement and domain hijacking attempts. During this time, there were multiple media reports where a group claimed to hack hundreds or thousands of websites in a single night. The intent was to instill widespread unease in the casual observer.

5.1 / The Common Element / As we looked closer, we saw that there was more to the story. One could assume that many of these attacks had an element of automation. But when we looked more closely, we saw something else interesting about the attacks.

While reviewing the list of affected websites, we noticed that many had the same IP address. This led us to believe the sites were hosted on the same server.

Figure 5-1: One defaced website served up pro-ISIS materials

Hundreds of hosting companies provide hosting for as little as a few dollars a month, hosting many paying accounts on the same server. This can result in hundreds of domains and sites running under the same server IP address.

One way to determine which sites are co-hosted at an IP address is to use Bing's IP operator (see Figure 5-2). Bing will reply with a list of known sites hosted at that IP address.

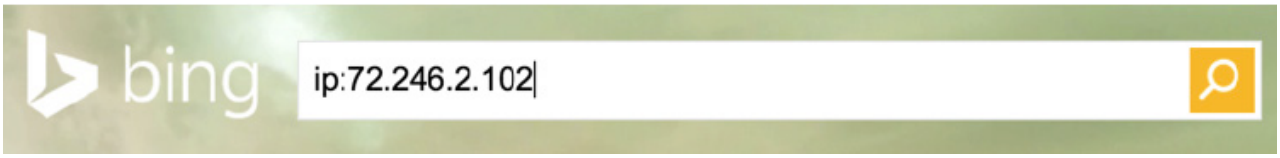


Figure 5-2: The Bing IP operator allows users to identify websites co-hosted at a single IP address

5.2 / Mass Website Defacements with Symlink / Mass defacements are often accomplished by combining the use of a directory traversal vulnerability and the deployment of a symlink to the defaced content. This vulnerability exists when the hosting server does not properly prevent accounts from accessing files outside their assigned directory structure.

As a result, a malicious actor can traverse the server's directories, potentially reading username and password lists, and access files from other customer accounts. This could include website database credentials. With this information, attackers could gain the ability to change files on every site on the server.

First, the attacker must get a foothold on the server. With hundreds of sites running on a single server, it's likely that at least one of them will be vulnerable to an attack that provides the attacker with the ability to upload files. Attackers often find vulnerable sites through Google hacking (see Figure 5-3) to identify vulnerable software, such as third-party content or content management system (CMS) plugins.

Figure 5-3: Google hacking involves using search queries to find vulnerabilities

The attacker will use the vulnerability — whether it is SQL injection, an insecure file upload or remote file execution — to implant scripts on the server. The attacker will often start with a shell script, such as the c99madshell, which provides visibility into the server structure and assists in gathering files such as account and password lists.

With the list of accounts, the attacker will upload a mass defacement script. This script uses the account names, which often match the web root for each customer on the server, and then accesses the desired files. The defacement script uses the credentials to overwrite the home page for each website, replacing it with the attacker's own file.

5.3 / Multi-Purpose Joomla and WordPress Defacement Script / One multi-purpose defacement script observed during Q1 focuses on Joomla and WordPress configuration files. This works similarly to the defacing script in the way it can access files in other account directories. With this deface script, it will access the WordPress `wp-config.php` or the Joomla `configuration.php` file to extract database credentials. Once it has the site owner's username and password, it will attempt to insert a new administrator account into the database.

Next, the script attempts to access the WordPress theme editor. The theme editor, shown in Figure 5-4, provides administrators with access to edit template files. If successful, the defacement script will then overwrite the home page template's code with the attacker's own defacing page.

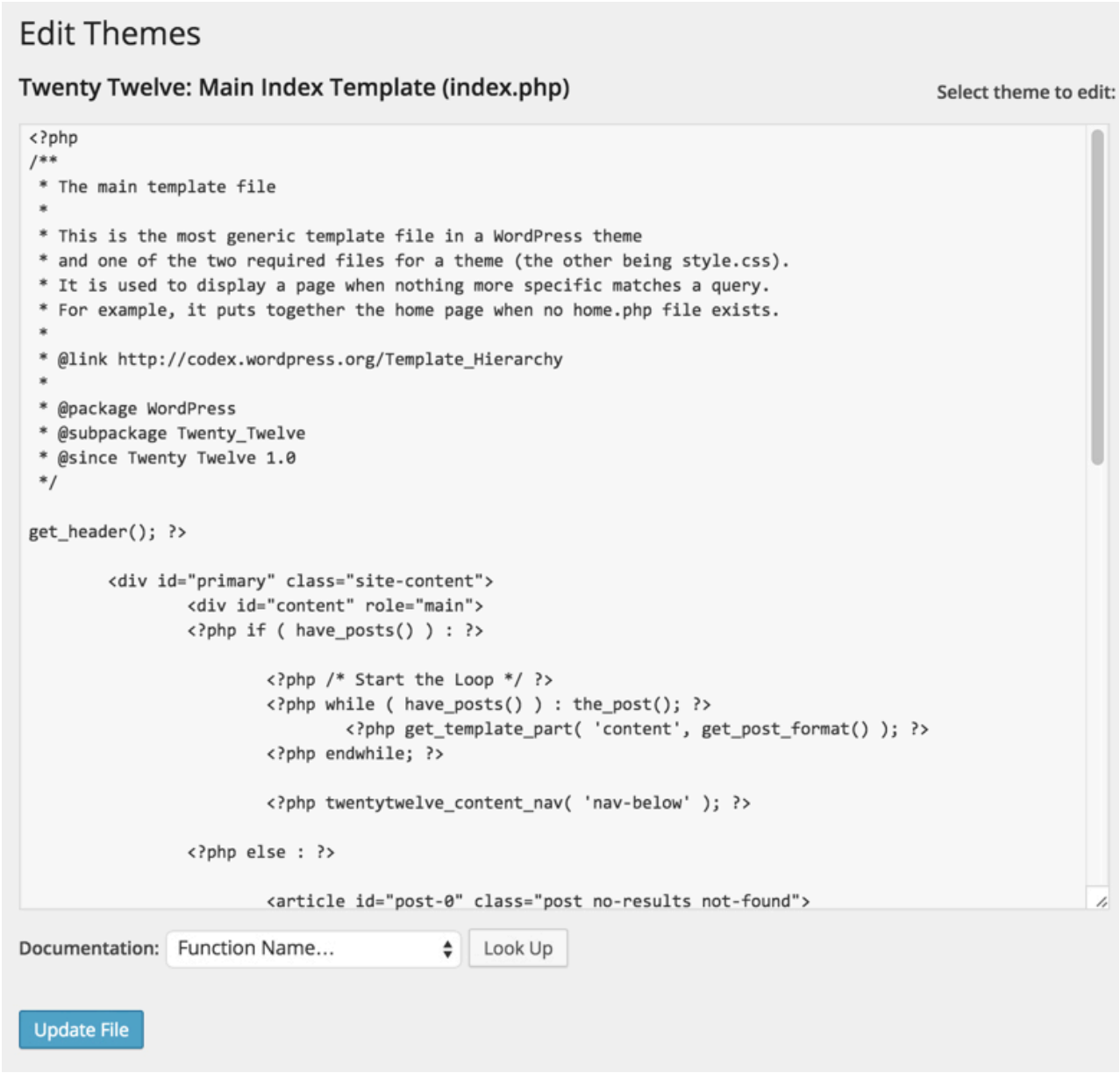


Figure 5-4: The WordPress theme editor allows users to change a website's template

If the script detects a Joomla installation, it will perform similarly. It uses the symlink attack described above to steal database credentials and then uses the Joomla `com_installer` plugin to overwrite the home page of the website.

On the Joomla side, the script will actually attempt to change the admin user's password. The real Joomla administrator will then be locked out of the system.

In the WordPress instance, the script would simply install an additional account with administrator privileges. The WordPress administrator would still be able to login to the system, but without careful forensics, may not know whether the attacker still has access to the CMS, even after the defacement is reversed.

5.4 / INDICATORS / This attack is one of the easiest to identify. The home page of your website, or possibly a secondary page, has been replaced or edited to have unwanted content. The message will often proclaim support for a cause or simply brag about defacing the site.

5.5 / DEFENSIVE MEASURES / If you are using a hosted service and you're affected by this attack, there may not be much that can be done, other than moving to another hosted service.

The Bing IP search can help you identify other accounts on your host IP. With this information, you can check those websites to determine whether they bear the hallmarks of compromise.

If the hosting provider allows for it, you may also test this vulnerability by trying to view the web space of other accounts on the same provider. First, check the hosting provider's policy to see whether it would be allowed. Akamai does not condone violating the hosting provider's Terms of Service.

If your hosting provider allows shell access, first examine the directory listing one step above your home directory. If you see other accounts listed there, then it is likely that an attacker who compromises one of those accounts will gain similar visibility into your files. You may use tools like `find` or `ls` on UNIX or `dir` on Windows to identify files in other users' directories. On UNIX, you might use a command like this to find directories under your home directory which are writeable by other users: `find ~ -perm -2 ! -type l -ls`

You will most likely want to address the permission issue to prevent other users from modifying your files.

Often, the host will not allow shell/command line access but will allow FTP client access. FTP clients often have a function that lets one traverse the directory structure. If you're able to view other accounts and their files, your server is more likely to be vulnerable to this attack.

5.6 / Domain Hijacking: Dangers and Defenses / Early in Q1, Akamai observed attacks that could bypass even the best web server security protections. Domain hijacking occurs when attackers gain access to a domain registrar account and change the DNS resource records to point to another server under the attacker's control.

5.6A / Nature of the Threat / Unlocked domain registration records expose a threat which has high repercussions. Spear-phishing attempts often target IT, Finance and Human Resources staff who may have access to domain registration accounts. Very often, this access is gained by phishing email credentials from a site's domain administrator. With the credentials, the attacker can perform a password reset on the registrar's site.

With the new password and administrative access, the malicious actor can login to the registrar and make changes to name server (NS) records for web servers and mail servers.

When the NS records are under the attacker's control, web and email traffic for the compromised domain can be redirected to any IP address controlled by the attackers. Often the NS record updates have a 24 – 48 hour time-to-live (TTL), so the damaging effects of a compromised registrar attack can be lengthy.

We have seen instances where the attackers modified the entire zone file, including mail exchange (MX) records, providing the attackers with access to any mail sent to the target of the attack. In addition to information leakage via email, the attackers can also use this access to trigger password resets on other services and compromise them as well. With the ability to intercept password reset attempts, the attackers could maintain control over all administrative accounts for a given domain name.

5.6B / Defensive Measures / Companies need a good awareness program to protect against phising. Wherever possible, please enable two-factor authentication (2FA) on hosted email services.

Additionally, protect against domain hijacking by employing the registrar locks available for domains. The first type of lock is a client lock, which prevents unauthenticated changes to a DNS record.

The client locks include:

- `clientUpdateProhibited`
- `clientTransferProhibited`
- `clientDeleteProhibited`

Most registrars offer these locks at no charge. Some registrars even turn them on by default. You can check to see if these locks are in place for a domain by running a whois command at a terminal prompt, as shown in Figure 5-5.

```
Domain Status: clientTransferProhibited
Domain Status: clientUpdateProhibited
Domain Status: serverDeleteProhibited
Domain Status: serverTransferProhibited
Domain Status: serverUpdateProhibited
```

Figure 5-5: A whois query will return domain lock status

If an attacker successfully obtains credentials to a registrar account, client locks will not prevent changes to the DNS records. The attacker can login to the registrar and turn off these locks to make the changes. For a higher level of protection, registrars offer server locks.

Server locks follow the same format:

- `serverUpdateProhibited`
- `serverTransferProhibited`
- `serverDeleteProhibited`

These locks offer a type of two-factor authentication. If these locks are in place, and someone tries to make DNS changes, even valid ones, the registrar will confirm the change with a previously agreed upon contact. The only drawback is the registrar may take up to a few days to turn these locks off. If rush changes need to be made, you will need to account for the time needed.

5.7 / Web Portals / Web portal accounts are at risk for password reuse attacks. These attacks leverage previously compromised credentials to check for accounts that reuse the same password for different Software-as-a-Service (SaaS) providers. Akamai has introduced two-factor authentication and other restrictions for logging into the Luna portal site for making configuration changes.

Some of the options that can be enabled include:

- IP restricted login
- Two-factor authentication
- SAML-based login
- User management APIs

In addition, practice safe password handling for all users who have access to external Internet infrastructure accounts, such as domain registrars and Akamai portal administration.

Also, do not use the same password for multiple sites. Store passwords for critical infrastructure offline in a secured location.

5.8 / Summary / The threats outlined in this section are certainly not new but as Akamai's researchers have found, the tactics remain popular.

A properly configured web application firewall has been able to protect Akamai customers from web application attacks such as defacements. Domain hijacking will continue to be a problem as long as proper controls are not in place at the registrar level. Organizations can heed the defensive advice above and share any additional defensive techniques with the larger security community.

[SECTION]6 CLOUD SECURITY RESOURCES

Throughout the year, Akamai security researchers track myriad threats. When necessary, Akamai releases advisories outlining the individual dangers, explaining how Akamai is protecting customers and offering defensive measures customers and the larger industry can take to minimize the impact. Meanwhile, the company keeps a close eye on vulnerabilities affecting its technology. When a vulnerability is found to affect Akamai, communications are launched to explain what the company is doing to address the issues. Additionally, as we continue to include more data sources in our report, we provide explanations of how we obtain and use this data to inform our research.

6.1 / Q1 2015 Advisory Recap: Attack Techniques and Vulnerabilities / What follows are the attack techniques and vulnerabilities Akamai tracked and addressed in Q1 2015.

6.1A / Retiring SSL / In Q1 2015, Akamai aggressively moved away from the use of Secure Socket Layer (SSL) in favor of Transport Layer Security (TLS).

Readers of the Q4 2014 report saw that the quarter began with researchers disclosing Padding Oracle on Downloaded Legacy Encryption (POODLE), a severe vulnerability affecting SSLv3. As that quarter progressed, attackers used Universal Plug and Play (UPnP) devices and DNS flooder tools to amplify their DDoS activity and employed Yummba webinject tools to commit banking fraud.

The POODLE vulnerability was the latest in a string of severe vulnerabilities in 2014, including Shellshock and Heartbleed.

At POODLE's core was a vulnerability in SSLv3 that attackers could exploit to calculate the plaintext (cleartext) in secure connections, effectively defeating SSL protection.

The SSL protocol was designed to protect communications on the Internet by wrapping them with encryption to preserve the confidentiality and integrity of communications. It is often used for banking transactions, shopping, secure messaging, instant messaging, and email. The vulnerability affected SSLv3 and did not affect newer encryption protocols such as TLS.

In an advisory, Akamai offered actions that organizations could take to mitigate the impact of POODLE. Recommendations included:

- Disabling SSLv3 wherever possible
- Applying patches and updates from vendors, especially in cases where SSLv3 could not be disabled
- Accelerating deprecation of SSLv3, as well as earlier versions

To help protect customers, Akamai deployed support for the TLS Signaling Cipher Suite Value (SCSV). SCSV prevents downgrading or fallback attacks to SSLv3 or earlier versions in case of a man-in-the-middle attack.

This cipher suite encodes the best protocol version that the client would have liked to use. Servers that support SCSV don't actually treat it as a cipher (what cipher suites normally list). Instead, if the value carried in the SCSV is worse than the best protocol version that the server supports, this connection is treated as an attack and fails the connection. A client only sends an SCSV value if it has already been forced to version downgrade; it's a way of signaling, "I tried to connect with a better protocol than I think you support. If you did support it, then somebody is messing with us."

Rich Salz, Akamai senior architect and member of the OpenSSL Development Team, was heavily involved in Akamai's work to move away from SSL. This is his description of the sequence of events:

- POODLE made it clear that Akamai had to protect the platform, and move from SSL *enabled by default* to *disabled*.
- We communicated closely with the Google team during our work.
- Akamai worked with OpenSSL to get a safe patch implemented into OpenSSL that could go into the release.
- Meanwhile, we prepared customer communications letting them know we were disabling SSL and asking them to let us know if they needed it.
- Akamai had the SCSV protection deployed on our network very quickly. This let us notice and prevent downgrade attacks. Customers who still wanted, needed, or thought they wanted/needed SSL3 remained vulnerable. But those customers who could use TLS were now protected.

- SCSV isn't perfect. Things like network hiccups and browser re-try happen a lot.
- By the end of Q1 2015, Akamai still supported SLSV3, but only for those customers who explicitly ask for it. It is otherwise disabled by default.

6.1B / DDoS Agents Target Joomla, Other SaaS Apps / A new attack threatened enterprises and Software-as-a-Service (SaaS) providers: chaotic actors using Joomla servers with a vulnerable Google Maps plugin installed as a platform to launch DDoS assaults. Akamai researchers working alongside researchers from PhishLabs' Research, Analysis, and Intelligence Division (R.A.I.D) discovered the attack technique.

Following a series of vulnerability disclosures throughout 2014, attackers began targeting the popular content management framework Joomla, specifically:

- Attack campaigns are designed to hijack large numbers of servers or SaaS providers that are then used to distribute malware and phishing campaigns. Hijacked systems are used as zombies in DDoS botnets.
- In the joint investigation we observed traffic signatures from Joomla distributions with a vulnerable Google Maps plugin used in DDoS attacks, as described in the Joomla Reflection DDoS Attacks Threat Advisory.
- The DDoS campaigns contained traffic signatures matching sites known for providing DDoS-for-hire services and matched attacks staged using tools developed specifically to abuse XML and Open Redirect functions, which then produce a reflected response that can be directed to targeted victims.
- PLXsert identified three distinct attack signatures produced by the DAVOSET and UFONet tools.

- The new DDoS attack type uses compromised Joomla servers with a vulnerable Google Maps plugin as zombies or proxies to stage denial of service GET floods.

Cloud-based DDoS attack mitigation can combat this problem to protect organizations from malicious traffic. Edge-based security and scrubbing centers stop DDoS attack traffic long before it affects a client's website or data center.

Specific actions to blunt this threat include:

- Blocking HTTP GET/1.0 request traffic, if support for legacy clients is not needed
- Blocking HTTP requests with a PHP-based User-Agent string, if they are not needed
- Using the three Snort rules provided in the threat advisory. The signature can be adapted to other mitigation techniques in order to detect or block these DDoS attacks

6.1C / CVE-2015-0235: Heap-Based Buffer Overflow Vulnerability in Linux Systems / In late January, researchers disclosed a vulnerability in the GNU C Library that could be exploited to take remote control of vulnerable Linux systems. Specifically, the problem was a heap-based buffer overflow in the glibc's `__nss_hostname_digits_dots()` function used in `gethostbyname()` and `gethostbyname2()` `glibc` function calls. The vulnerability became known as Ghost in the media:

- According to the Red Hat Bugzilla advisory, an attacker could remotely exploit this condition to make an application call to either of these functions. In the process, the attacker could launch malicious code with the permissions of the user running the application.

- Threatpost published a report on the vulnerability, stating, "The vulnerability, CVE-2015-0235, has already been nicknamed GHOST because of its relation to the `_gethostbyname` function. Researchers at Qualys discovered the flaw and say it goes back to `glibc` version 2.2 in Linux systems published in November 2000."

- The issue was first reported by security vendor Qualys. In a separate advisory, Qualys researchers said they stumbled upon the vulnerability during an internal code audit. "We discovered a buffer overflow in the `__nss_hostname_digits_dots()` function of the GNU C Library (glibc)," Qualys said in the advisory. "This bug is reachable both locally and remotely via the `gethostbyname()` functions, so we decided to analyze it — and its impact — thoroughly, and named this vulnerability GHOST."

Akamai engineers examined the primary software components that power the Akamai platform and found no exposure to this flaw. Regardless, Akamai exercised caution and patched older deployments of glibc.

6.1D / Attackers Use New MS SQL Reflection Techniques / Malicious actors used a fairly new reflection-based DDoS tactic to tamper with the Microsoft SQL Server Resolution Protocol (MC-SQLR) and launch DDoS attacks in Q1 2015.

Akamai first spotted attackers using the technique in October. But in Q1, researcher Kurt Aubuchon studied another such attack and offered an analysis. Akamai researchers replicated this attack by creating a script based on Scapy, an open-source packet manipulation tool, and published its findings in the MS SQL Reflection Threat Advisory.

The attack manifests in the form of MS SQL Server responses to a client query or request via abuse of the Microsoft SQL Server Resolution Protocol (MC-SQLR), which listens on UDP port 1434.

MC-SQLR lets clients identify the database instance with which they are attempting to communicate when connecting to a database server or cluster with multiple database instances. Each time a client needs to obtain information on configured MS SQL servers on the network, the SQL Resolution Protocol can be used. The server responds to the client with a list of instances.

Attackers abuse SQL servers by executing scripted requests and spoofing the source of the query with the IP address of the intended target. Depending on the number of instances present in the abused SQL server, the amplification factor varies.

An attack presented a specific payload signature and produced an amplification factor of nearly 25x. In this case, the attacker's request totaled 29 bytes, including IP and UDP headers, and triggered a response of 719 bytes including headers. Some servers may produce a larger or smaller response depending on their configuration.

Other tools publicly available on the Internet could reproduce this attack as well. Replicating this attack does not require a high level of technical skill. A scripted attack would only require a list of SQL servers exposed on the Internet that respond to the query. Attackers could use a unicast client request 0x03 or a broadcast request 0x02. Both are requests with a data length of 1 byte that will produce the same type of response from SQL servers.

PLXsert identified a tool on GitHub on January 26, 2015, that weaponizes this type of attack for mass abuse.

Server hardening procedures should always be applied to servers that are exposed to the Internet. As a rule, services and protocols that are unnecessary should be disabled or blocked.

This attack can only be performed by querying SQL servers with exposed SQL Server Resolution Protocol ports to the Internet.

The following best practices can help mitigate this type of DDoS attack. These recommendations are by no means exhaustive and affected organizations should refine and adapt them further based on specific infrastructure and exposed services.

- Follow Microsoft Technet Security Best Practices to Protect Internet Facing Web Servers.
- Apply ingress and egress filters to SQL server ports at firewalls, routers, or edge devices to help prevent this attack. If there is a business case for keeping UDP 1434 open, it should be filtered to allow only trusted IP addresses.
- Block inbound connections from the Internet, if ports are not needed for external access or administration.
- Disable SQL Server Resolution Protocol service if there is only one database instance. This has been disabled by default since Microsoft SQL Server 2008. It is not disabled in earlier or desktop engine versions. Disable this service to prevent the abuse of SQL server for this type of attack.
- If the use of SQL Server Resolution Protocol service is needed, add an additional layer of security before the service is accessed, such as authentication via secure methods (SSH, VPN) or filtering as described above.

6.1E / Data Breaches Fuel Login Attacks / Public dumps of compromised data from several high-profile attacks fueled an increase in automated and systematic attempts to reuse stolen credentials at multiple websites.

The requests showed that user agents were systematically randomized. One of the most targeted sectors was online financial services. Other industries targeted by these brute force attacks were online entertainment, high-tech consulting and SaaS. Specifically, malicious actors:

- Harvested ID and password combinations from public dumps of compromised data
- Used automated tools to systematically attempt to gain access to other sites using the available credentials
- Exploited the availability of numerous high-profile and large data dump leaks
- Used brute force login attempts more frequently following compromises and disclosures of big data dumps

To prevent these attacks, we recommended enterprises follow the same best practices around passwords and access control that have been standard for years:

- Enforce password complexity requirements
- Enforce account lockout threshold limits
- Monitor suspicious traffic and activity following login attempts or successful logins
- Use tools such as CAPTCHA or RECAPTCHA
- Use multi-factor authentication
- Use randomized URLs for login to mitigate automated tools
- Lockout IP addresses that have made multiple login attempts, providing they are not proxies
- Use rate control rules

Learn more in the Data Breaches Fuel Login Attempts Threat Advisory.

[SECTION]7
LOOKING FORWARD

The volume and frequency of DDoS attacks will fluctuate from one quarter to the next, but the innovation of attackers will continue, and with it, an ever-expanding palette of threats to organizations large and small.

Expect the heavy barrage of attacks in the gaming industry to continue, as players keep looking for an edge over competitors, and security vulnerabilities in gaming platforms continue to attract attackers looking for low-hanging fruit.

The security implications of IPv6 outlined in this report will come into clearer focus in the coming quarters, and Akamai will continue to piece together the full risks and suggest best practices to neutralize threats that may materialize over time.

SQL injection and local file inclusion (LFI) attacks will remain popular in the coming months, for the simple reason that chaotic actors have enjoyed so much success through these methods.

As malicious actors continue to innovate and perfect older techniques, it's very likely that the frequency of attacks exceeding 100 Gbps will increase in the coming months.

We will also continue to see malware in ads, third-party service attacks, and the attackers continue to find security holes in the many widgets and plug-ins used across myriad platforms.

DDoS trends will include more attacks, the common use of multi-vector campaigns, the availability of booter services and low-cost DDoS campaigns that can take down a typical business or organization. The expansion of the DDoS-for-hire market may result in the commoditization of DDoS attacks, where availability drives down prices, which grows the market. DDoS may become a common tool for even non-technical criminals.

On the Akamai Edge network, we expect to continue seeing an escalation of LFI and SQL injection attacks, due to malicious actors' history of success with these methods.

Web application attacks over both HTTP and secure (HTTPS) connections will continue in future quarters, particularly LFI and SQLi attacks.

The retail and media sectors will continue to take heavy fire, as attackers continue to find profit there.

Collaboration is imperative for the software and hardware development industry, application and platform service providers, and the security industry in order to break the cycle of mass exploitation, botnet building and monetization.

About Prolexic Security Engineering & Research Team (PLXsert)

PLXsert monitors malicious cyber threats globally and analyzes these attacks using proprietary techniques and equipment. Through research, digital forensics and post-event analysis, PLXsert is able to build a global view of security threats, vulnerabilities and trends, which is shared with customers and the security community. By identifying the sources and associated attributes of individual attacks, along with best practices to identify and mitigate security threats and vulnerabilities, PLXsert helps organizations make more informed, proactive decisions.

About Threat Research Team

The Threat Research Team is responsible for the security content and protection logic of Akamai's cloud security products. The team performs cutting edge research to make sure that Akamai's cloud security products are best of breed, and can protect against the latest application layer threats.

About Customer Security Incident Response Team (CSIRT)

The Akamai Customer Security Incident Response Team (CSIRT) researches attack techniques and tools used to target our customers and develops the appropriate response – protecting customers from a wide variety of attacks ranging from login abuse to scrapers to data breaches to DNS hijacking to distributed denial of service. It's ultimate mission: keep customers safe. As part of that mission, Akamai CSIRT maintains close contact with peer organizations around the world, trains Akamai's PS and CCare to recognize and counter attacks from a wide range of adversaries, and keeps customers informed by issuing advisories, publishing threat intelligence and conducting briefings.

Contact

Twitter: @State_Internet
Email: stateoftheinternet-security@akamai.com

Akamai FASTER FORWARD

As the global leader in Content Delivery Network (CDN) services, Akamai makes the Internet fast, reliable and secure for its customers. The company's advanced web performance, mobile performance, cloud security and media delivery solutions are revolutionizing how businesses optimize consumer, enterprise and entertainment experiences for any device, anywhere. To learn how Akamai solutions and its team of Internet experts are helping businesses move faster forward, please visit www.akamai.com or blogs.akamai.com, and follow @Akamai on Twitter.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 40 offices around the world. Our services and renowned customer care enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers and contact information for all locations are listed on *www.akamai.com/locations*.
