Imperva's Web Application Attack Report

Edition #1 - July 2011

Protecting the Data That Drives Business®

Table of Contents

① Abstract

As a part of its ongoing Hacker Intelligence Initiative, Imperva's Application Defense Center (ADC) observed and categorized attacks across 30 applications as well as onion router (TOR) traffic, monitoring more than 10 million individual attacks targeted at web applications over a period of six months. The analysis shows:

- Due to automation, web applications, on average, are probed or attacked about 27 times per hour or about once every two minutes. At the apex of an attack, web applications experience nearly 25,000 attacks per hour or 7 per second.
- Four dominant attack types comprise the vast majority of attacks targeting web applications: Directory Traversal, Cross-Site Scripting, SQL injection, and Remote File Inclusion.
- The United States is the main source of application attacks. Applications are attacked by infected computers, or bots, with most located in the US.

We provide a list of technical recommendations for security teams as well as nontechnical ones for corporate executives.

② Executive Summary

Web applications are a primary target for hackers. The recent of spate of high profile breaches, including PBS, Sony, Epsilon and more, were largely application attacks. Applications have become major targets for a simple reason: they transact and access large amounts of personal and corporate data that hackers hope to monetize on black markets.

As of July 2011, it is estimated that there are 357,292,065 websites on the internet today.[1] Estimates indicate that most applications had an average 230 vulnerabilities during 2010.[2] Assuming just the top 1% of websites drive commerce and require a high degree of protection, this means they have a total 821,771,600 vulnerabilities in active circulation. But which will be exploited?

To answer this question, Imperva's Application Defense Center (ADC), as a part of its ongoing Hacker Intelligence Initiative, studied actual malicious web application attack traffic over a period of six months, December 2010 through May 2011. The ADC monitored and categorized more than 10 million individual attacks across the internet, including attacks witnessed via onion router (TOR) traffic as well as attacks targeting 30 different enterprise and government web applications. The ADC studied the frequency, type and geography of origin of each attack to help security professionals better prioritize vulnerability remediation and web application security projects based on real hacker activity.

Most notably, the ADC found that attack automation is prevailing. Modern botnets scan and probe the Web seeking to exploit vulnerabilities and extract valuable data, conduct brute force password attacks[3], disseminate spam, distribute malware, and manipulate search engine results. These botnets operate with the same comprehensiveness and efficiency used by Google spiders to index websites. As the recent Lulzsec episode highlighted, hackers can be effective in small groups. Further, automation also means that attacks are equal opportunity offenders; they do not discriminate between well-known and unknown sites or enterprise-level and non-profit organizations.

Learning from Lulzsec

The Lulzsec attacks took place largely during the month of June and our report cut off was May. Consequently, we didn't directly witness any attacks from Lulzsec. However, it is interesting to note the incredible similarity between our observations and the attacks used by the half-dozen (or so) hacker team. As we explained in our blog, "Lulzsec was a team of hackers focused on breaking applications and databases."[4] To achieve this, they employed three of the four attack methods we describe in this report: SQL injection, RFI and Cross Site Scripting.

Key Findings

The ADC study yielded the following key findings:

1. **Automation is prevailing.** According to the study, websites experience an average of 27 attacks per hour or about once every two minutes. However, 27 attacks per hour is only an average. When sites come under automated attack, the target can experience up to 25,000 attacks per hour or 7 per second.

How many attacks occur?

24158 Peak attacks per hour

27 Average number of attacks per hour

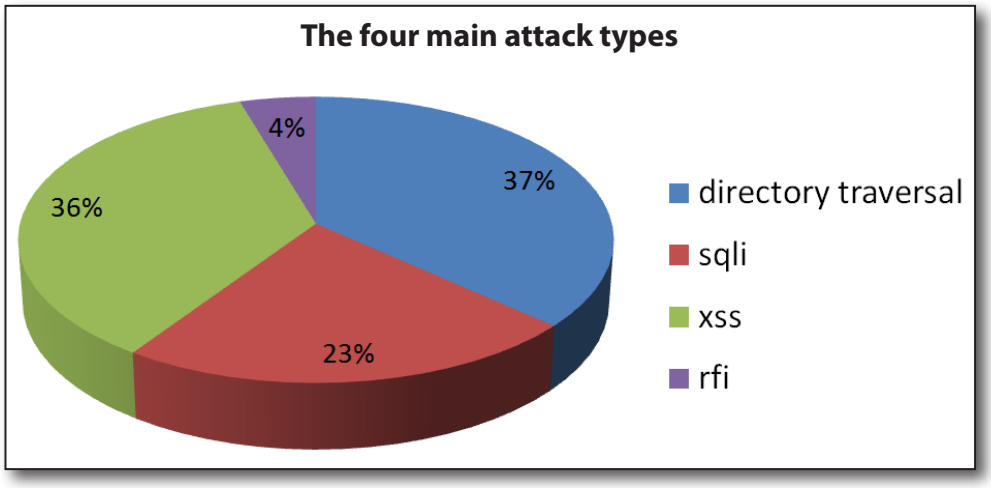
Attack traffic during the six month period was characterized by short peaks of high activity followed by longer periods of lighter activity. During the peaks of activity, attack volume (number of attack vectors in the attack) was quite large, in the thousands, and attack rate (number of requests per second) was rapid, both common indicators of automated activity. Additionally, each of the primary attack methods observed was executed in high-volume bursts or waves, also confirming the existence of automation. The ADC analyzed attacks against Alexa rankings and found that site popularity was not a factor that helped determine targets. Detecting and stopping automated application attacks will be an essential security skill. In a world of automated attacks, applications of all sizes are – or will be – targeted.

[1] http://news.netcraft.com/archives/2011/07/08/july-2011-web-server-survey.html
[2] https://www.whitehatsec.com/home/resource/stats.html
[3] http://www.nytimes.com/2010/01/21/technology/21password.html
[4] http://blog.imperva.com/2011/06/analyzing-the-lulzsec-attacks-.html

2. **The Unfab Four:** The ADC monitored all web traffic, marking 10 million events as suspicious. Four dominant attack types comprised the vast majority of attacks targeting web applications: Directory Traversal, Cross-Site Scripting, SQL injection, and Remote File Inclusion (RFI). Directory Traversal and Cross-Site Scripting were the most widely used attack types, making up almost 75 percent of all attack traffic. SQL injection was also prominent, making up 23 percent of attack traffic, while RFI attacks made up 4 percent. Other events were noticed as well but these four comprised the bulk and therefore deserve intense focus.

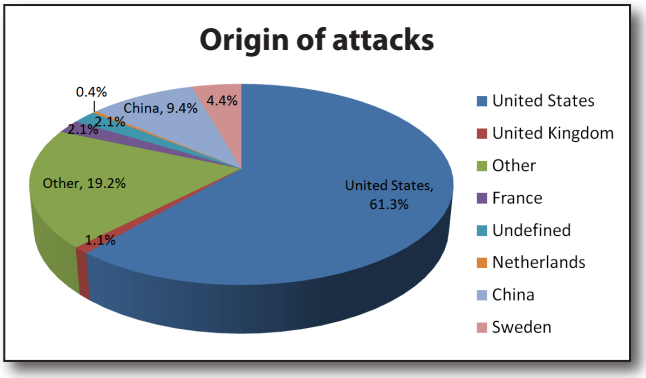
   Most automated attacks typically consist of two phases, scan and exploit, and most often hackers use these attacks in a compound fashion. For example, a hacker may use Directory Traversal during a reconnaissance phase of the combined attack to identify the directory structure of an attacked server before sending an additional effective exploit vector, such as an RFI.

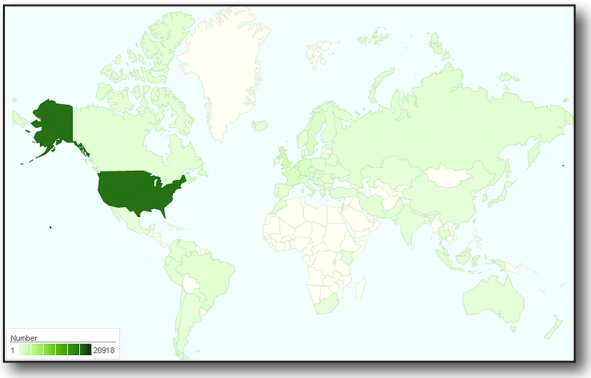
   Other web application attack types were observed during the six months, but these were the four most significant. In fact, the only significant attack type not covered in this document are the coordinated search engine scans which the ADC covered in a separate report.[5]



3. **Most attacks come from within the United States.** Over 61 percent of the attacks monitored originated from bots located in the United States, although conclusions cannot be made regarding from which geography these bots are controlled. Attacks from China made up almost 10 percent of all attack traffic, followed by attacks originating in Sweden and France.

   Additionally, the ADC found that 29 percent of the attack events originated from the 10 most active attack sources. While filtering based on geography is far from reliable, sorting traffic based on reputation is viable.





[5] http://www.imperva.com/docs/HI_Search_Engine_Poisoning_SEP.pdf

③ Background

In 1989, headlines across the world warned consumers about apples sprayed with Alar, a pesticide suspected of being a carcinogen. The Alar problem was even featured on 60 Minutes. Dr. Bruce Ames, then the chairman of UC Berkeley's biochemistry department, vigorously dismissed the Alar issue, writing "that a glass of the suspect Alar-contaminated apple juice posed only 1/10th the possible carcinogenic hazard of the average peanut butter sandwich and 1/50th that of a mushroom…"[9] Dr. Ames went on to itemize a list of dozens of everyday products, such as bananas and broccoli, that contained natural carcinogens while pinpointing a few chemicals which posed the strongest danger. Consequently, the public had a better idea of what to fear.

Today's software security industry is experiencing something similar. As of July 2011, it is estimated that there are 357,292,065 websites on the internet today.[10] WhiteHat, who has conducted the most comprehensive research on vulnerabilities, estimates that most applications had an average 230 vulnerabilities during 2010.[11] Assuming just the top 1% of websites drive commerce and require a high degree of protection, this means they have a total 821,771,600 vulnerabilities in active circulation. But which will be exploited?

Carcinogens and vulnerabilities are a real concern and need to be well understood. However, like the confused consumer buying apples in 1989, anyone worried about security vulnerabilities is easily overwhelmed. In the spirit of Dr. Ames, Imperva's Application Defense Center sought to quantify and answer a key question: "What vulnerabilities matter most to attackers?" Instead of seeing the world through the application's eyes, we decided to look at the world through the attacker's eyes. To do so, we deployed several "weather balloons" on the internet, where we could observe polluted traffic. In addition, we monitored about 30 different web applications during the past 6 months (December 2010-May 2011). The applications experienced about 10 million individual attacks during this period.

Web applications are a major target for hackers. This year's Verizon Data Breach report showed that of all successful breaches, 50% were a result of hacking.[12] The recent spate of high profile breaches, including PBS, Sony, Epsilon and more, were application attacks. Applications have become major targets for a simple reason: they transact and access the data hackers hope to monetize on black markets. In the past, hacking was about taking a site offline to publicly embarrass or blackmail the operator. Today, the attack method is much more about "hide and seek", i.e., take data without detection and sell it. Mature online exchanges exist that resemble eBay in structure, only their focus is selling personal and corporate data. For example, a few months ago, on a hacker forum, one hacker offered to sell full administrative rights to several government, military and educational websites for $499[13]:

Advanced Persistent Threat (APT) and the Conundrum of Geography

Much has been written on what comprises APT and who is responsible. This has become very important considering the US government's recent declaration that cyber attacks are an act of war and will be met with kinetic retaliation.[6] One of the most notable examples of APT was China's suspected attack against Gmail accounts belonging to US government officials.[7]

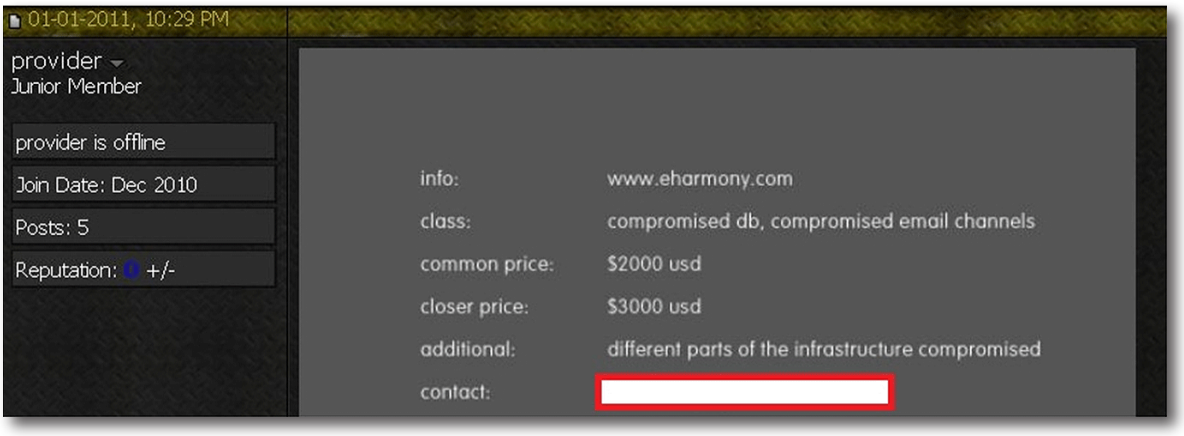
Although IP address is the most common tool for determining someone's location, it is unreliable. Individual attacker addresses are most probably masked today using various proxy services. However, as most of the attack traffic is attributed to compromised machines, we were able to identify that 30% of attacks came from ten identifiable IP addresses. This is significant for governments hoping to stop cyber crime. Notably, the UK government announced an initiative to stop cyber crime using a global network of vendors and law enforcement. Our research indicates that reputation-based services should be a key aspect of the initiative. Real-world examples help provide further color. Brian Krebs, in describing the attack against RSA breaches, wrote:[8]

…according to interviews with several security experts who keep a close eye on these domains, the Web sites in question weren't merely one-time attack staging grounds: They had earned a reputation as launch pads for the same kind of attacks over at least a 12 month period prior to the RSA breach disclosure.

[6] http://online.wsj.com/article/SB10001424052702304563104576355623135782718.html
[7] http://abcnews.go.com/Politics/us-probe-alleged-chinese-hack-senior-officials-gmail/story?id=13744049
[8] http://krebsonsecurity.com/2011/05/rsa-among-dozens-of-firms-breached-by-zero-day-attacks/
[9] http://www.fortfreedom.org/n16.htm
[10] http://news.netcraft.com/archives/2011/07/08/july-2011-web-server-survey.html
[11] https://www.whitehatsec.com/home/resource/stats.html
[12] http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf
[13] http://blog.imperva.com/2011/01/major-websites-govmiledu-are-hacked-and-up-for-sale.html

http://cecom.army.mil/	The United States Army | CECOM	Full SiteAdmin Control/SSH Root access	unknown	$499

So, for the price of an iPad, you could have purchased the ability to control a US Army web site. Earlier this year, a hacker tried to sell access to dating site eHarmony for $2,000:



Cumulatively, McAfee estimate sizes this market at $1 trillion.[14]

Another factor making applications attractive is the fact that many organizations haven't deployed proper defenses. Hackers prefer the path of least resistance. Although media reports focus attention on large, well-known organizations, the path of least resistance has made smaller, poorly resourced enterprises prime targets. For example, when the small town of Pittsford, New York, was hacked and lost $139,000, the town supervisor said, "We have good firewalls and anti-virus software, and we weren't at all lax in our security systems. We thought we were pretty secure."[15] This statement helps summarize part of the problem. Today, most organizations spend security dollars on network firewalls and anti-virus leaving applications ripe for attack. Enterprises spend nearly $27 billion on security[16] with less than $500 million on protecting applications.[17]

Data, and the applications that transact data, mean that cyber security can't be separated from business operations. Enterprises aren't just setting up firewalls to keep the bad guys out. For this reason, how CEOs must view cyber security has changed dramatically. In the past, security was the technician's realm – setting up firewalls and deploying anti-virus. Today, the challenge is to know where data resides, who touches it, where it moves and how to protect it. This requires business process experts, not just technicians, to keep the bad guys repelled while keeping the good guys productive. More and more, CEOs are feeling duty bound to protect data. In a June 2011 article on security, the consultancy McKinsey wrote:

> *At leading organizations, cybersecurity should be a constant item on the agendas of CEOs and boards. To stay ahead of the threats, executives must engage in an ongoing dialogue to ensure their strategy continually evolves and makes the appropriate trade-offs between business opportunity and risks.*[18]

For this reason, in our conclusion, we itemize a list of actions CEOs should take help bolster cyber security.

[14] Unsecured Economies Report, http://resources.mcafee.com/content/NAUnsecuredEconomiesReport (registration required).
[15] http://krebsonsecurity.com/2011/06/fbi-investigating-cyber-theft-of-139000-from-pittsford-ny/
[16] IDC, Worldwide and U.S. Security Services 2011–2015 Forecast and Analysis, May 2011
[17] Gartner, Market Share Analysis: Security Software, 20 May 2011.
[18] https://www.mckinseyquarterly.com/Meeting_the_cybersecurity_challenge_2821

For breached organizations, the cost of a breach is difficult to determine, though we haven't found anyone who would label breaches "cheap" or "fun."[19] We can, anecdotally, categorize the cost of a data breach stemming from:

- **Loss in shareholder value:** For example, Heartland Payment Systems experienced a large data breach in 2009. For nearly two years, financial analysts watched as large legal payments for damages were settled before the market could feel comfortable about Heartland's ability to stabilize revenues.
- **Legal costs:** Companies can be sued for data breaches. In May 2011, the FTC concluded legal settlements with Ceridian Corporation and Lookout Services, Inc., which were "part of the FTC's ongoing efforts to ensure that companies secure the sensitive consumer information they maintain."[20]
- **Loss in brand:** After the Sony breach, USA Today reported that, "Sony had a mediocre buzz score of 24.3 on YouGov's daily consumer perception tracking BrandIndex until the breach dropped its score to a lackluster 7.6 among adults 18 and older."[21]
- **Notification costs:** In many parts of the world, organizations are required to notify everyone if their data records have been compromised. In the US, for instance, data breach notification requirements could well become federal law.[22]
- **Service outages:** In many cases, hacking an application means the online service must be removed until the issue is fixed. Costs vary greatly on vertical, ranging from thousands of dollars per hour to millions.[23] And this doesn't include the fact that after a breach, normal development activity freezes. Instead, there is a mad rush dedicated to strengthening defenses which delays new product development.

④ Analysis Methodology

This security summary report is based on the analysis of Internet traffic to 30 different web applications during the past 6 months (December 2010-May 2011) as well as TOR traffic. From the raw traffic we extracted some 10 million events of interest which we analyzed for attack traffic. The observed applications belong to organizations and companies from various industry segments: e-commerce, online services, web search, banking, communication, entertainment, marketing and technology.

Monitoring of the web applications deployed at these sites over a period of several months produced data that was processed using automatic tools. In addition, Imperva's security experts performed detailed analysis of important events and patterns.

The processing of the observed data was based on attributes such as vulnerability signatures that match HTTP properties in the access traffic and filtering the traffic based on black lists of known malicious sources. We used Imperva's knowledge-base to sift through the monitored traffic and find attack-related events. We matched these events against known attacks and known security vulnerabilities in web applications. The events were categorized based on parameters like the type of attack they were part of, the identity of the attacker, the identity of the victim and the exploited security vulnerability.

In the next step of the analysis we used statistical tools and data mining methods to look for patterns in the identified web attacks, like long term trends in specific attacks, combinations of attacks by the same attacker, and coordination between different attack sources used by a single attacker.

Finally, we extracted insights about global security trends from the data analysis and came up with recommendations for thwarting new and growing threats and improving web security.

[19] A great read on the topic comes from Microsoft: Sex, Lies and Cybercrime Surveys by Dinei Florencio and Cormac Herley.
[20] http://www.ftc.gov/opa/2011/05/ceridianlookout.shtm
[21] http://www.usatoday.com/tech/gaming/2011-05-09-playstation-reputation-takes-pounding_n.htm?loc=interstitialskip
[22] http://www.infosecurity-us.com/view/18750/senators-introduce-national-data-breach-notification-legislation/
[23] http://www.zdnet.com/news/the-real-cost-of-application-outages/244421

⑤ Analysis Results

There are four major attack types that were observed with high frequency in the data: SQL Injection, Remote File Inclusion, Directory Traversal and Cross Site Scripting. We begin our analysis by describing phenomena and trends that are common to all attack types. After that we dedicate a section to describe phenomena associated with each specific attack type.

5.1 General Observations

5.1.1 - Part I: Attack Automation

Our findings show that automation is prevailing and hackers have become very adept at automating attacks. We can confirm what the 2011 Verizon Data Breach report noted when they observed that hackers "have created economies of scale by refining standardized, automated, and highly repeatable attacks directed at smaller, vulnerable, and largely homogenous targets." However, 27 attacks per hour is only an average. When sites come under automated attack, the target can experience up to 25,000 attacks per hour or 7 per second. Detecting and stopping automated attacks will be an essential security skill. We analyzed attacks against Alexa rankings and found that size doesn't matter. In a world of automated attacks, everyone is – or will be – a target.

Our analysis includes smaller companies whose revenues are less than $500 million per year and experience much lower traffic than highly-rated Alexa sites. In one forum, a hacker boasted finding 5,012 websites that were vulnerable to SQL injection (redaction ours):

5012 SQL Injectable Websites

Collected by [redacted] - [redacted]

http://www.[redacted]/trainers.php?id='4
http://www.[redacted].com/trainers.php?id='30
http://[redacted].com/trainers.php?id='30
http://www.[redacted].com/article.php?ID='338
http://www.[redacted].com/publications/article.php?ID='51
http://www.[redacted].com/article.php?id='13798
http://www.[redacted].com/news/article.php?Id='0222
http://www.[redacted].com/article.php?id='59
http://www.[redacted].com/press/article.php?id='000073
http://www.[redacted].com/article.php?id='5
http://www.[redacted]et/article.php?id='104
http://[redacted].net/article.php?id='1089
http://www.[redacted].net/news/article.php?id='416
http://[redacted].net/article.php?id='2524
http://www.[redacted].net/article.php?id='11012&lang='th
http://www.[redacted].net/news/article.php?id='48
http://www.[redacted].net/nl/article.php?id='1512&type='col

It is well known that attackers are relying more and more on automated tools to design and implement concentrated and coordinated attacks on web applications. Figure 11 highlights, for example, that a steady stream of attacks are punctuated by an intense, focused burst.

The following attributes identify an attack in the monitored network traffic as automated:

- **Attack volume:** the attack is composed of a large number (thousands) of attack vectors.
- **Attack rate:** attack vectors are sent at a high rate (several requests per second) which is impossible to achieve manually.
- **Attack consistency:** attack vectors are similar to each other. For example, RFI attack vectors contain the same remote script URL, or SQLi attack vectors contain similar SQL phrases.
- **Attack coordination:** Similar attack vectors targeting the same application are sent within a short time period from multiple sources, indicating that the activity of these hosts is coordinated.
- **Violation of HTTP protocol during the attack:** Either by design or due to bugs in the automation scripts, some of the attack vectors are invalid HTTP messages (e.g., the URL or a parameter containing an illegal character). This will never occur when the client is a regular browser. The monthly rate of HTTP violation in the traffic is shown in Figure 1. A good correlation between HTTP violations and Directory Traversal attacks during May can be observed in Figure 2.

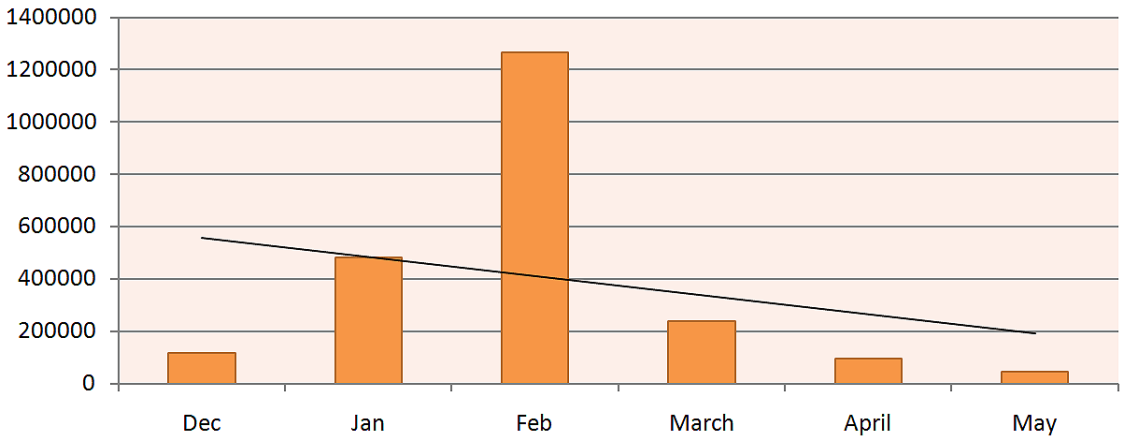


Figure 1: HTTP violations per month

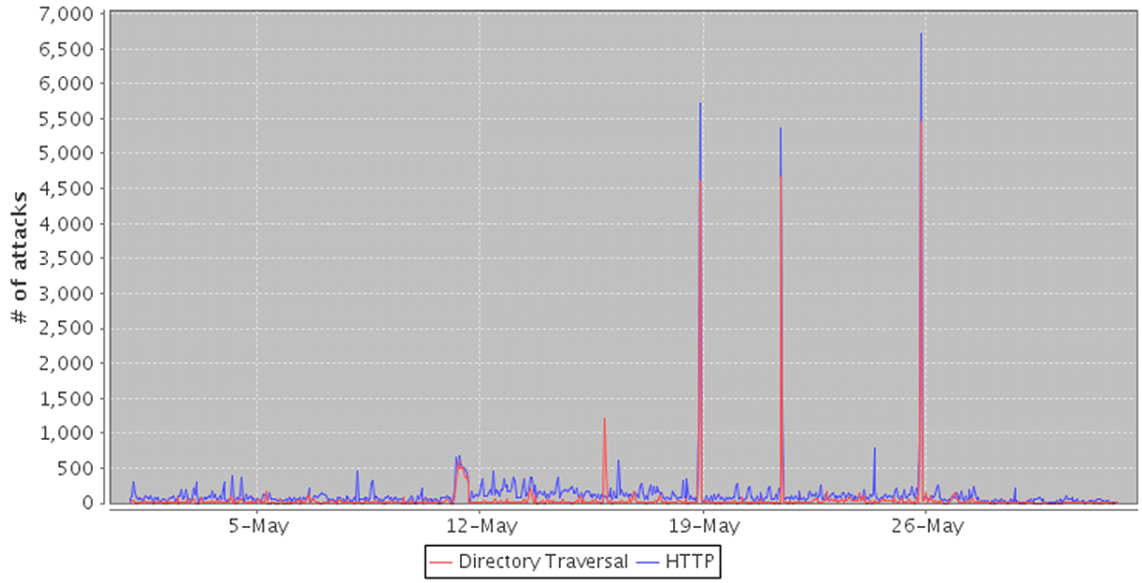


Figure 2: HTTP violations and Directory Traversal occurrences - May 2011

Based on these criteria, our observations clearly show that RFI, SQLi and DT attacks are indeed mostly implemented using automatic attack tools.

Automated attacks usually consist of two phases: Scan and Exploit. The scan phase finds potential vulnerable candidates to attack, and the following exploit phase takes advantage of the detected vulnerabilities.

Scanning may be done in several ways:

- **Vertical scanning:** After selecting a specific target application, the attack tool is traversing it to find all the URLs and parameter it uses. It tries to attack every parameter with a malicious vector. This is often done with a cracked version of legitimate commercial penetration testing tools.
- **Horizontal scanning:** The attack tool has a database of searchable indicators of known applications with vulnerabilities (known as *Google dorks*). This database is built from various sources, including public forums in which security experts publish weaknesses they found in popular applications. By querying regular search engines with these indicators and analyzing the returned results the attack tool scans the internet looking for deployed vulnerable applications.
- **Opportunistic scanning:** Some tools skip the scanning phase entirely and attack the application with a set of known vulnerabilities of popular web applications, regardless of their existence in the application itself.

An attack tool usually contains a pool of means to exploit found vulnerabilities for the attack's purpose: retrieval or modification of data, denying service from the application, etc. For example, in an RFI attack the tool uses a pool of URLs of scripts, while in an SQLi attack the tool can use a set of predefined SQL snippets intended to retrieve the contents of sensitive database tables (like tables of account names and passwords).

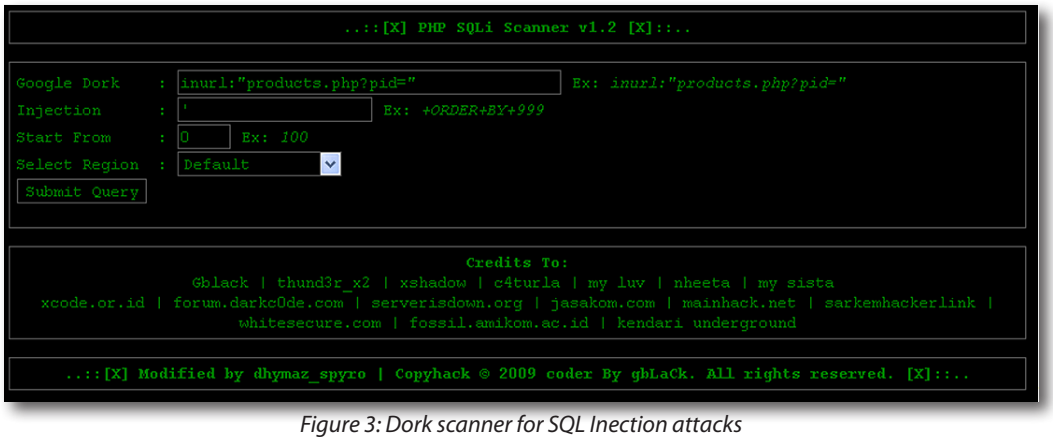


Figure 3: Dork scanner for SQL Inection attacks

5.1.2 - Part II: The Unfab Four

The relative volume of traffic associated with the investigated attack types is shown in Figure 4. *Cross Site Scripting (XSS) and Directory Traversal (DT)* are the most prevalent attack types, while there were fewer attack events related to *Remote File Inclusion (RFI) and SQL Injection (SQLi)*.

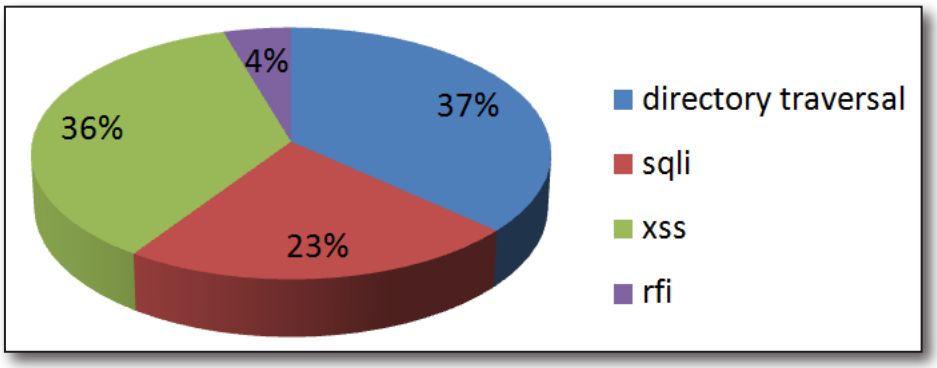


Figure 4: Relative traffic-volume per attack type

The following table summarizes the statistical attributes of the attacks. We defined an activity peak as a period in which the activity is larger than the average activity by at least one standard deviation. The attack traffic has short periods of activity peaks punctuated with long periods of low activity, so we measured the average activity with the peaks and used the median of the number of attacks as an estimate of the activity without the peaks (since it is less sensitive to outliers):

	Maximum	Median	Average	Standard Deviation
SQLi	2705	10	48	158
RFI	7072	4	17	165
DT	8233	25	63	241
XSS	6148	28	63	209

Table 1: Hourly attack activity statistics

	Maximum	Median	Average	Standard Deviation
SQLi	21734	553	982	2021
RFI	7092	79	190	583
DT	28773	1024	1493	2503
XSS	18259	862	1440	2011

Table 2: Daily attack activity statistics

5.1.3 - Part III: Geographic Dispersion

Even though the identity of the host that initiated an attack is not necessarily indicative of the identity of the attacker that controls it, we have investigated the geographic distribution of the attack-initiating hosts as determined by their IP addresses.[24] For all attack types the attackers were spread around the world but most of the attacks (both in absolute numbers and counting the distinct hosts initiating the attacks) were from the United States. A significant portion of SQL Injection attacks we observed coming from a relatively small number of Chinese hosts (see Figure 3). The leading source countries were rather consistent across all attack types. The average ratio of attacks to attacking hosts is about 10:1 for RFI, 25:1 for SQL injection and 40:1 for DT. (Note: Cross-site scripting is not included in this chart as the geographic location of these attacks cannot be determined).

RFI		SQLi		DT	
Country	**Attacks**	**Country**	**Attacks**	**Country**	**Attacks**
USA	20918	USA	91606	USA	189474
United Kingdom	1897	China	47800	Sweden	13535
Netherlands	1879	Sweden	8789	France	9417
France	1253	Indonesia	3604	Netherlands	8320
Republic of Korea	1070	United Kingdom	3419	Germany	7656
Germany	1030	Netherlands	2793	United Kingdom	6692
Sweden	1012	Ukraine	2489	European Union	4159
Brazil	506	Republic of Korea	2374	Canada	3492
Russian Federation	490	Romania	2136	Republic of Korea	2838
European Union	460	Germany	1263	China	2507

Table 3: Countries from which most attacks were initiated

[24] To properly understand the statistical significance of geographic traffic, these figures should be normalized by the overall traffic from these countries or the total number of host IPs in each country. We did not do this normalization.

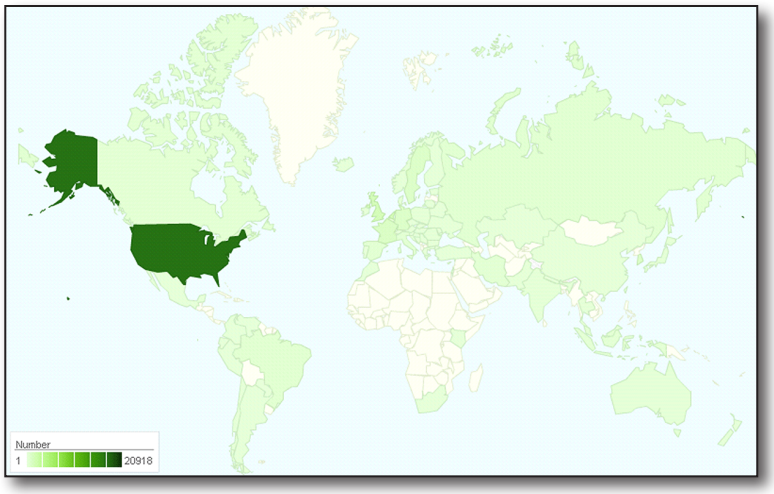


Figure 5: Remote File Inclusion - number of attacks from each country

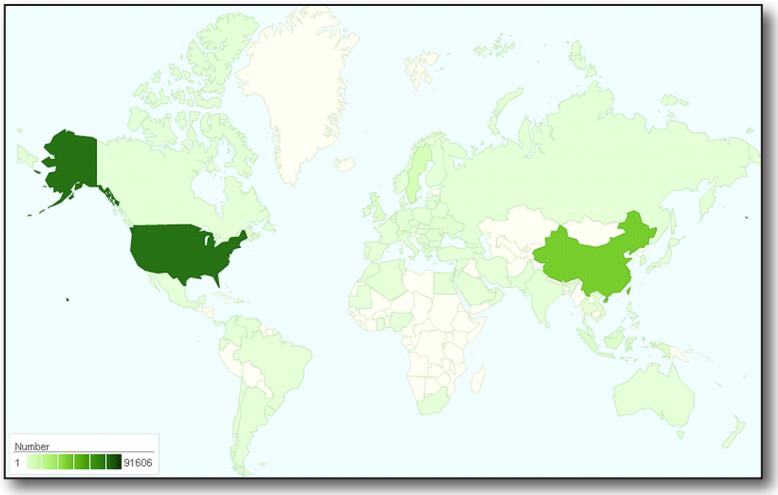


Figure 6: SQL injection – number of attacks from each country

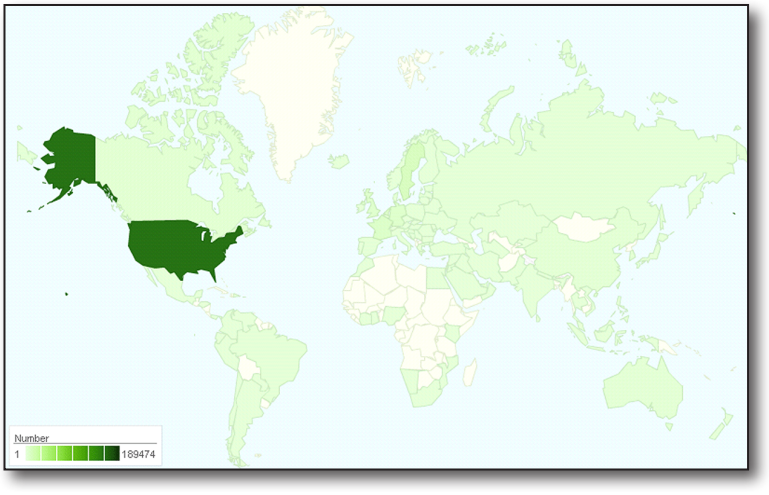


Figure 7: Directory Traversal - number of attacks from each country

RFI		SQL injection		DT	
Country	**Attackers**	**Country**	**Attackers**	**Country**	**Attackers**
USA	714	USA	20424	USA	5169
Germany	96	China	575	United Kingdom	733
Republic of Korea	93	United Kingdom	322	Germany	398
France	73	Canada	193	France	311
European Union	62	Russian Federation	105	European Union	264
Netherlands	61	Indonesia	105	Netherlands	214
United Kingdom	61	European Union	90	Australia	178
Italy	54	India	58	Canada	170
Russian Federation	42	Republic of Korea	56	China	149
Canada	36	Germany	53	Russian Federation	131

Table 4: Countries with the most distinct attackers

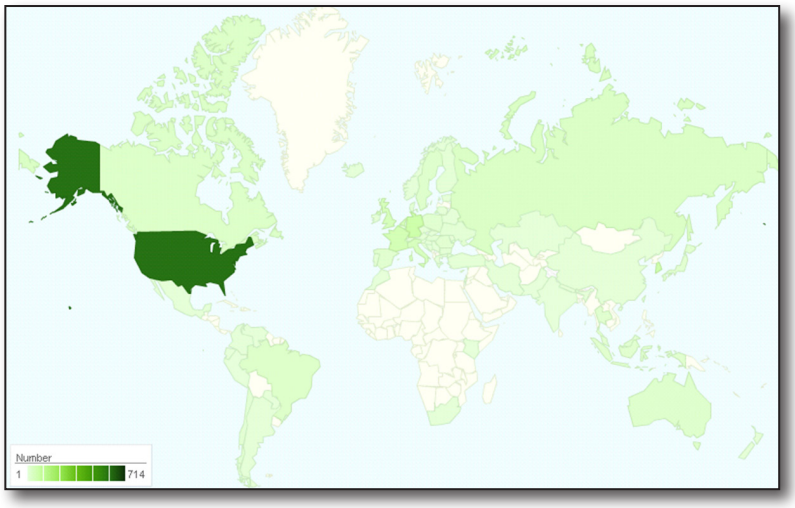


Figure 8: Remote File Inclusion - number of distinct attackers per country

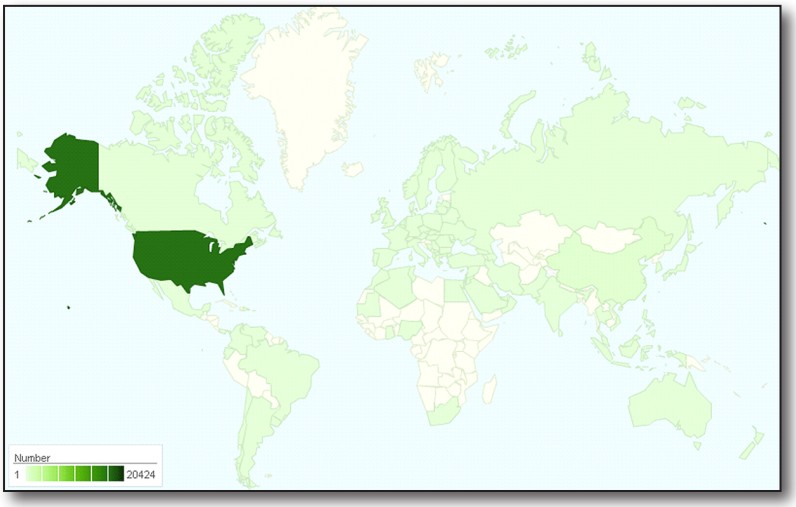


Figure 9: SQL Injection - number of distinct attackers per country

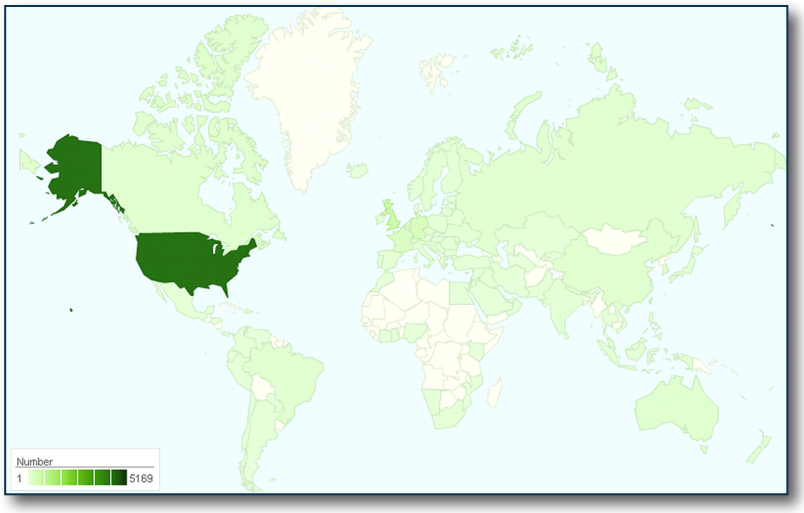


Figure 10: Directory Traversal - number of distinct attackers per country

5.2 More of the Unfab Four

***SQL Injection (SQLi)** is an attack that exploits a security vulnerability occurring in the database layer of an application (like queries). Using SQL injection the attacker can extract or manipulate the web application's data. The attack is viable when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.*

The monthly number of SQLi attacks is shown in Figure 11. There were 15000-53000 SQLi attacks each month on the observed sites (27,000 attacks per month on average). Our data shows a general long-term decrease in the number of monthly attacks, but we are still unable to explain this observation. We intend to check this trend in the following months.

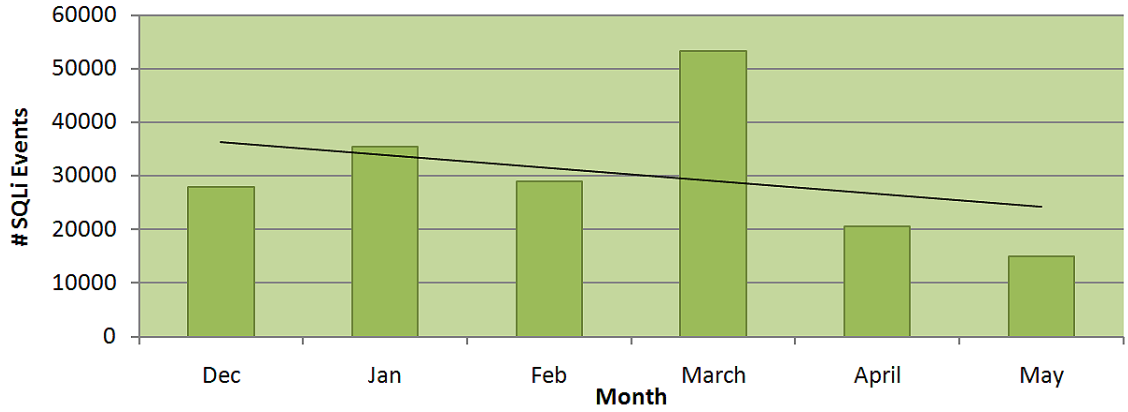


Figure 11: SQLi attacks per month

We have also seen that a large portion of SQLi attacks (24%) show up as bursts of high-frequency attack attempts during short time periods – typically up to a thousand attempts in less than an hour. Between these attack peaks, the average number of SQLi attacks is low (10 per minute). Figure 12 shows this by zooming into the occurrences of SQLi attacks during March 2011. From the high attack frequency (2705 per hour or 45 per second), we can conclude that the attack bursts are generated by automatic attack tools.

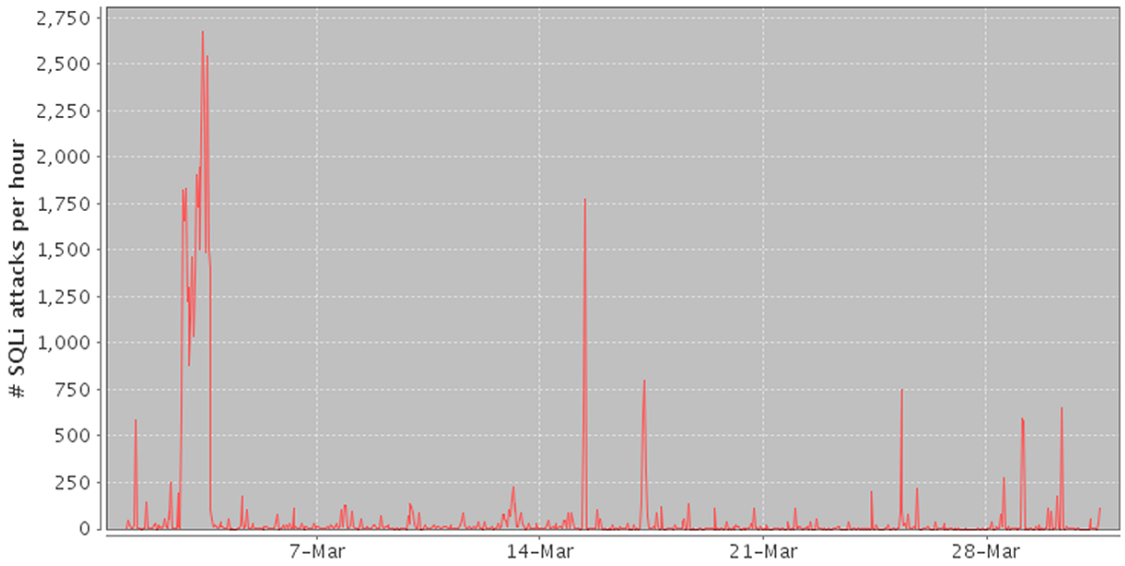


Figure 12: SQLi attack peaks

In a particular attack during a one hour period on March 5th, a few source hosts targeted the same web application with the same type of payload. This attack was on a set of related application URLs. The attack's payload was a variation on SQL ***waitfor*** delay. Because of the repetitions of the same payload, we concluded that in this case the attacker intention was Denial of Service (DoS), by tying-up the application's database and shutting down the site.[25]

> ***Remote File Inclusion (RFI)*** *is an attack that allows an attacker to include a remote file, usually through a script, on the web server. This attack can lead to data theft or manipulation, malicious code execution on the web server, or malicious code execution on the application's client side (such as Javascript which can lead to other attacks).This vulnerability occurs due to the use of user-supplied input without proper validation.*

The monthly occurrence of RFI attacks is summarized in Figure 13. There were 2300-13500 RFI attacks each month on the observed sites (5200 attacks per month on average). We don't have a firm conclusion about the long-term trend in RFI occurrences. We intend to keep monitoring this in the following months.

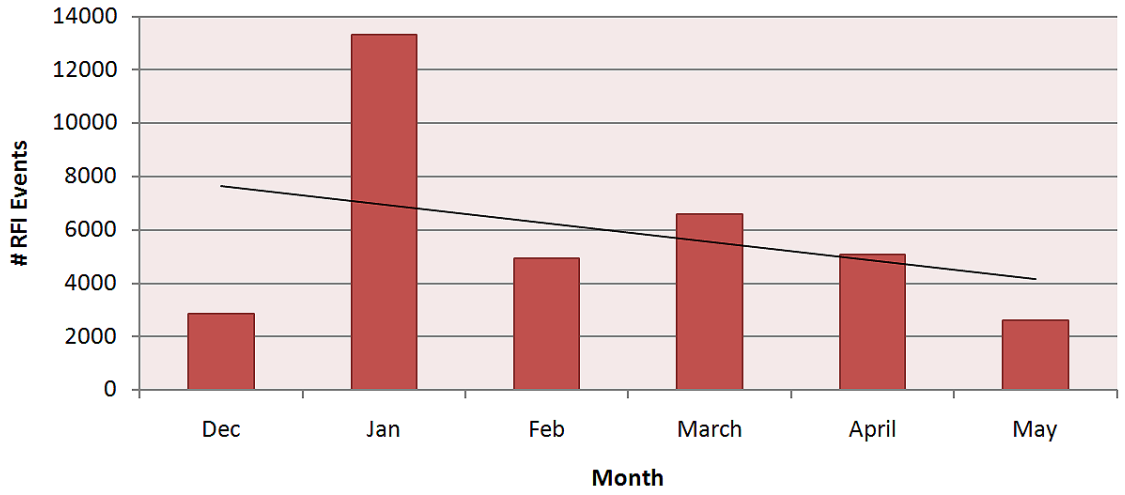


Figure 13: RFI attacks per month

[25] See: http://stackoverflow.com/questions/2023854/defending-against-a-waitfor-delay-sql-injection-attack

We have seen that a large portion of RFI attacks (41%) show up as bursts of high-frequency attack attempts during short time periods – typically up to a 35 attempts per minutes or about 1 attack every two seconds. Between these attack peaks, the average number of RFI attacks is low (4 per minute). This behavior can be observed, for example, by zooming into the occurrences of RFI during January and March of 2011 (Figure 14, showing a single massive attack during January, and Figure 15 showing several attack peaks during March). Judging by the rapid attack rate, the attack bursts are generated by automatic attack tools (see section Error! Reference source not found.). We concluded that RFI attacks in particular are almost exclusively done by automatic tools: hackers build automatic scripts that collect potential victims, prepare a pool of malicious scripts for exploitation, and try to inject the script into the victim's web application using a database of known RFI vulnerabilities. This process can be done without human intervention.

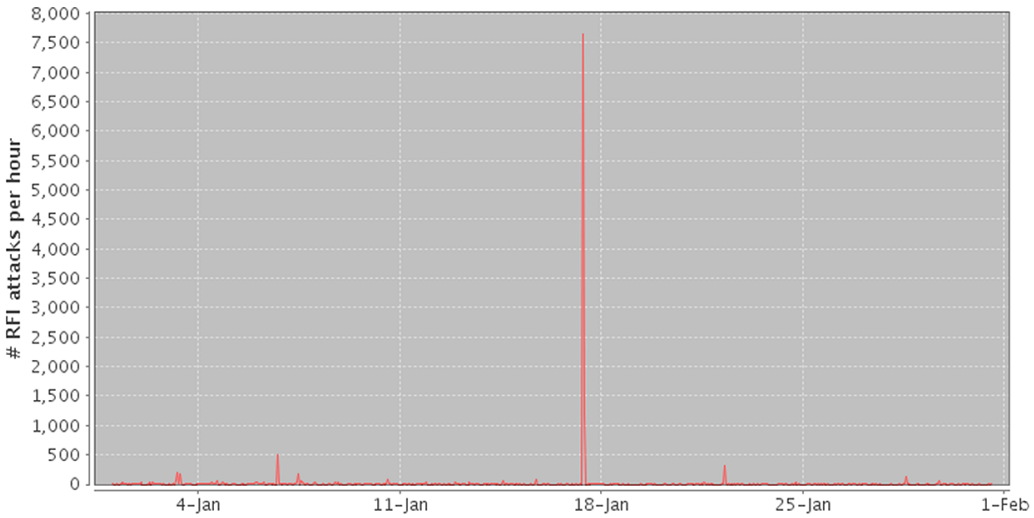


Figure 14: RFI attack peaks – January 2011

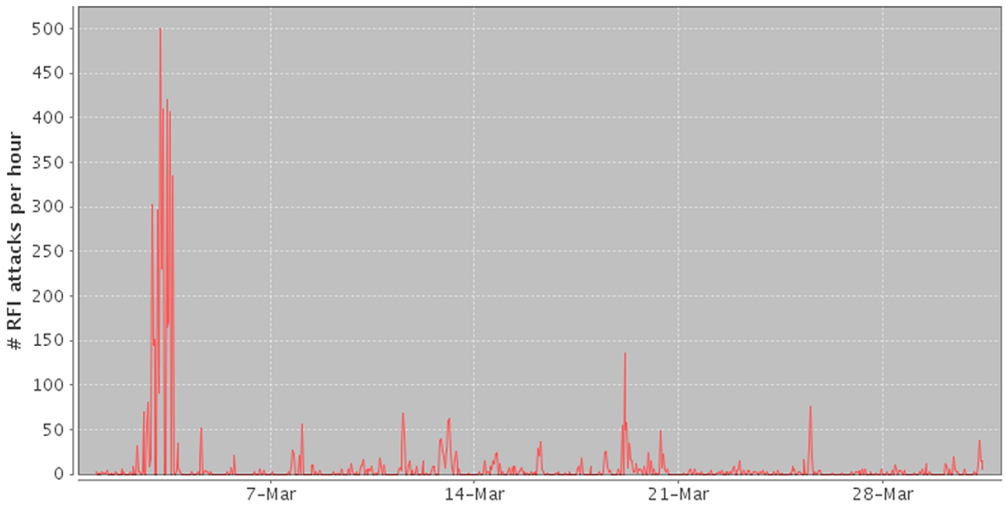


Figure 15: RFI attack peaks - March 2011

Remote File Inclusion is used by an attacker to execute malicious code on vulnerable web servers, usually as a first step towards taking over that server. Once the code is injected and executed on the server, the attacker can control the web application and its interaction with the users. Furthermore, the attacker may escalate the penetration into the web server (for example, using misconfigured files permissions) in order to use it for purposes like attacking other hosts.

In an RFI attack the attacker specifies a URL of a malicious script as a parameter in legitimate calls to the application's code, counting on the application to load and execute it without validation. We collected more than 800 different URLs that were used as parameters in RFI attempts. We investigated more than 150 unique injected scripts that these URLs reference. These scripts were variations on 10-15 basic scripts that were slightly modified by various hacker groups. They were usually written in the PHP scripting language, since RFI vulnerabilities are typical to applications using PHP. A few of the scripts, however, were written in the Perl language. There are various functionalities that the scripts provide:

- 85% of the scripts are just vulnerability probes. They test the attacker's ability to execute code by including a distinctive message in the application's output. These scripts are short (less the 4Kbytes) and there are multiple copies of each one that the attackers use interchangeably to avoid detection or overload in their hosting computers.
- 10% of the scripts are more sophisticated and open a control channel back to the attacker. This IRC-based channel enables the attacker to remotely control actions performed by the scripts, like extracting information from the host, scanning the injected host for other security vulnerabilities and exploiting the discovered vulnerabilities. Additionally, they enable the attacker to use the host as a platform for attacking other sites, as part of a botnet. Scripts of this type are usually 4-90Kbytes long.
- The remaining 5% of the scripts are similar in attack potential to the previous category, but they also inject HTML pages into the legitimate application. This lets the attacker control the injected script using a hidden Web UI that the application unknowingly exposes instead of through IRC commands. The piggybacked attack-UI remains online while the vulnerable web application is online. Scripts of this type are naturally longer, up to 200Kbytes.

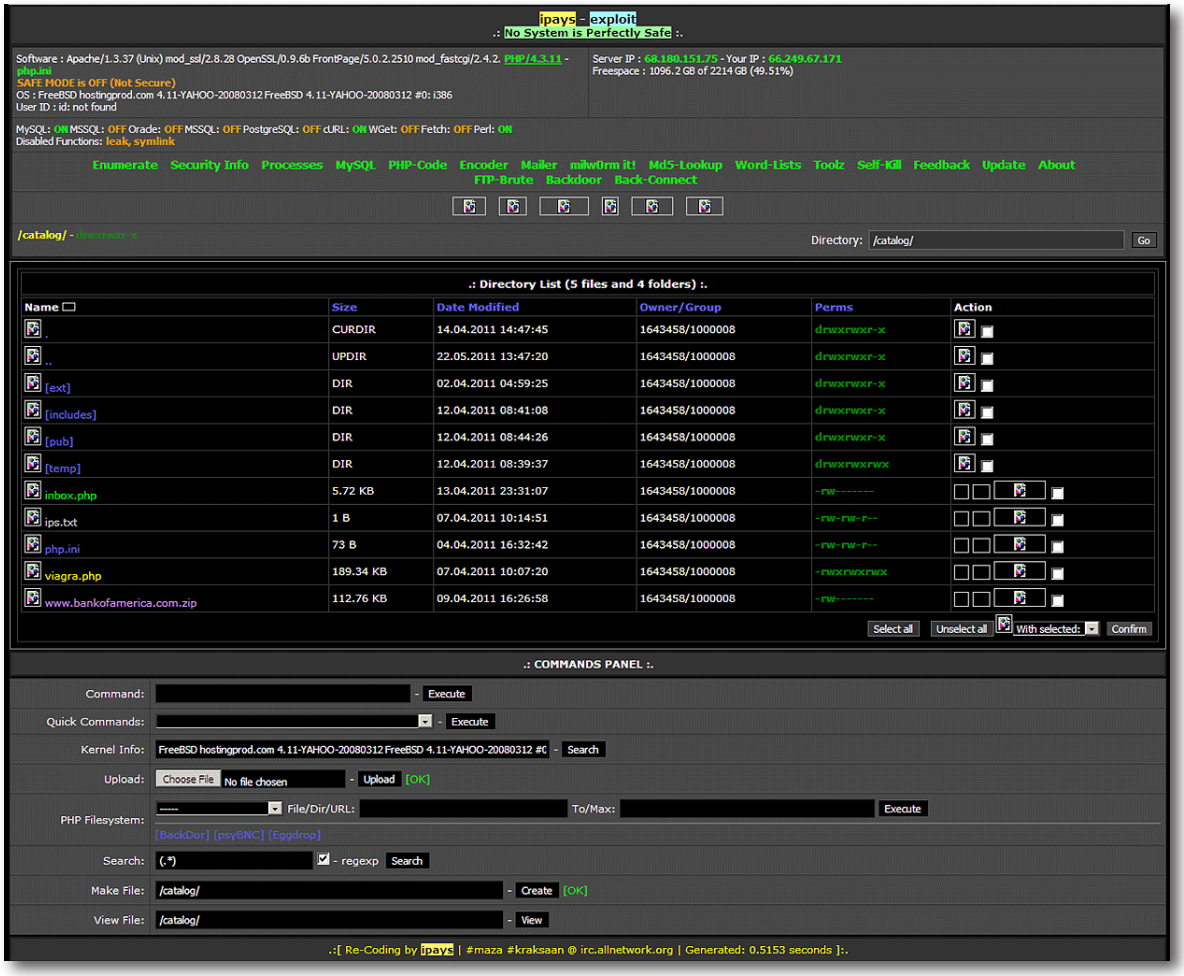


Figure 16: Web UI of a malicious script injected via RFI vulnerability

The usual method for identifying RFI attacks is by matching HTTP requests against ***signatures of known vulnerabilities of common web applications*** that may be vulnerable to RFI. A signature-based attack detection mechanism works well for protecting popular applications with enough mileage that their security weaknesses were investigated and published. However, attackers may succeed in bypassing signature-based defenses by exploiting newly discovered vulnerabilities in web applications (especially in new versions of them) or exploiting vulnerabilities in custom code. We have compared this detection method against other mechanisms for detecting RFI attacks:

- Matching HTTP requests with signatures for ***URLs of known remotely-included malicious files*** (which are passed in the HTTP request of an RFI attack)
- Using ***black lists of RFI-initiating hosts***.

These detection mechanisms are less application-specific, so they can be updated whenever an attack is detected in any set of sites and then be rapidly published for protecting all the other sites. We note that this protection approach fits the modern model of RFI attacks in which the same set of hosts acting as a centrally-controlled botnet attacks a large set of potential victims, and uses a shared pool of malicious scripts in the RFI attacks they initiate. Initial results of this community-based detection and protection approach show that it caught a large number of RFI attacks that were not identified using only vulnerability-related signatures.

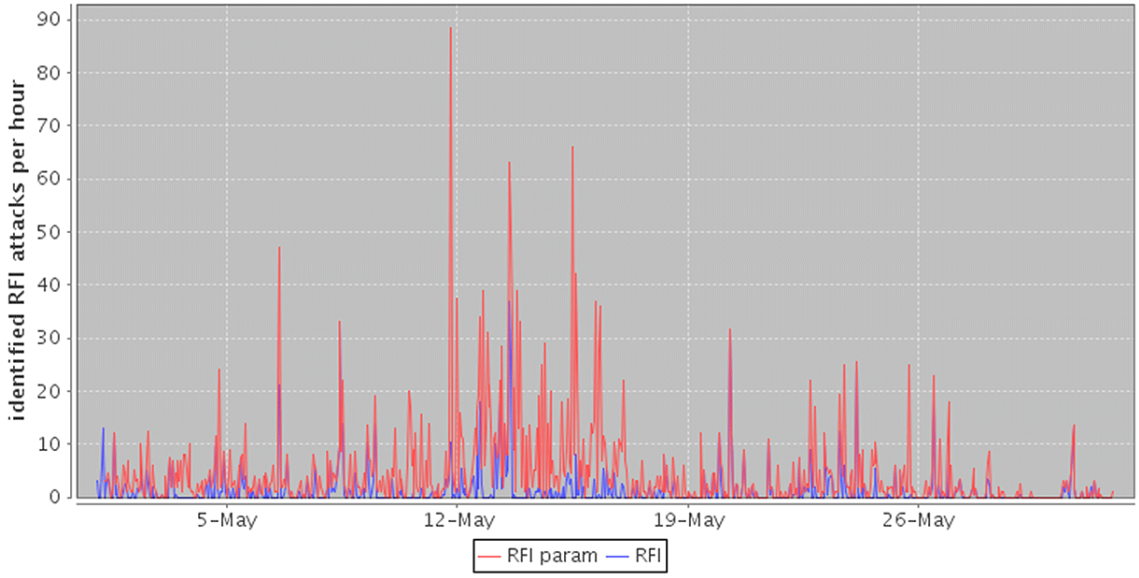


Figure 17: Detected RFI attacks over time - using vulnerability signatures (RFI) and scripts-URLs signatures (RFI param)

Directory Traversal (DT) *is an attack that orders an application to access a file that is not intended to be accessible and expose its content to the attacker. The attack exploits insufficient security validation or insufficient sanitization of user-supplied input file names, so that characters representing "traverse to parent directory" are passed through to the file APIs.*

The monthly occurrence of Directory Traversal attacks is summarized in Figure 18. There were 29000-73000 DT attacks each month on the observed sites (42000 attacks per month on average). The number of monthly occurrences of DT attacks is generally increasing. The average number of hourly DT attacks (63 per minute) is high relative to the other attacks.

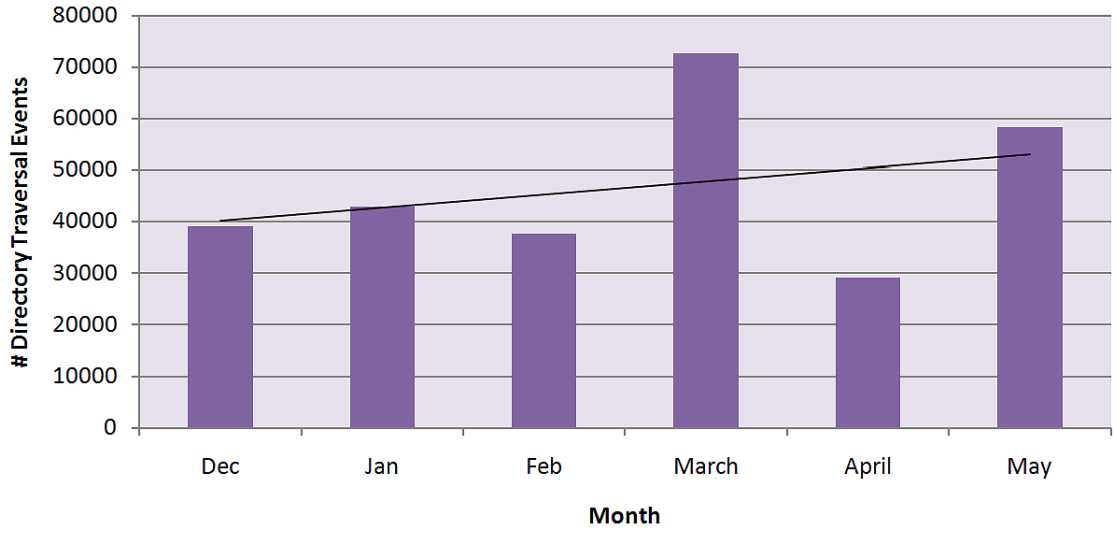


Figure 18: Directory Traversal attacks per month

Directory Traversal is a prevalent attack that is traditionally used to expose the contents of sensitive files on the attacked web server to the attacker. For example, a common DT attack attempts to extract the user's passwords from the system file in which they are stored. Our observation is that in addition to this traditional notion of DT, this attack is also used together with other attack types. Directory Traversal is used by attackers during a reconnaissance phase of the combined attack to identify the directory structure of an attacked server before sending an additional effective exploit vector.

Significant changes in the volume of Directory Traversal attacks are less common than in other attack types. However, when DT is used in conjunction with other attack types like RFI, the attack burst includes both DT reconnaissance attempts and the RFI exploit attempts, as can be seen from the DT occurrences during March.

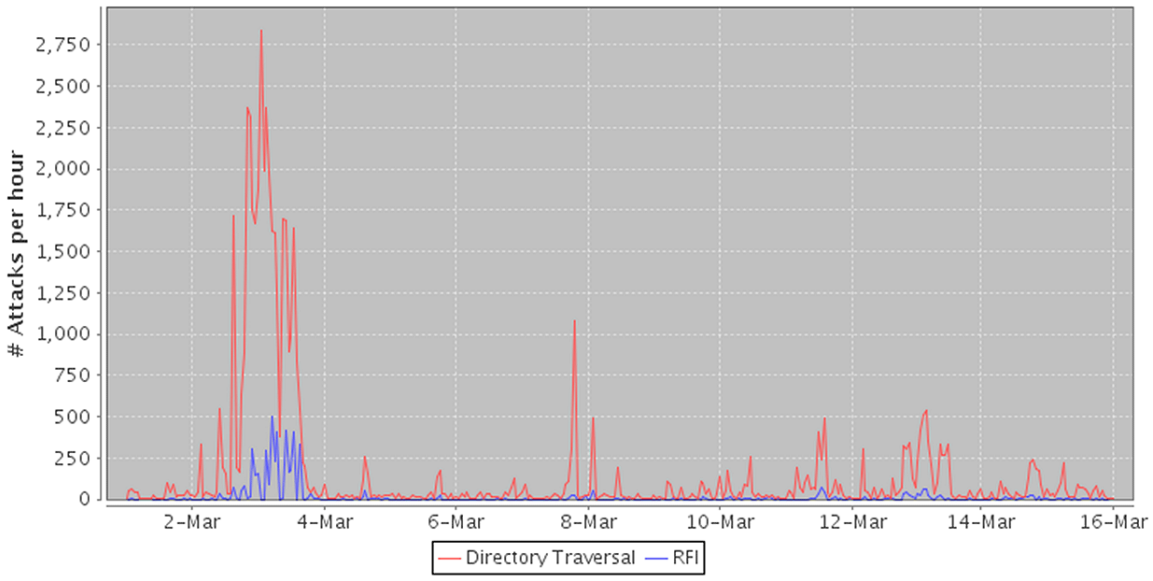


Figure 19: Directory Traversal and RFI attacks during March

Cross Site Scripting (XSS) *is an attack that lets the attacker execute scripts in a victim's browser to hijack user sessions and steal his credentials, deface web sites, insert hostile content, redirect users, hijack the user's browser using malware, etc. XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content.*

The monthly volume of traffic related to Cross Site Scripting is summarized in Figure 20. Cross Site Scripting is targeted at web users rather than servers. It is used to attack the application's client executing malicious browser code in the context of the trusted application, potentially abusing the trust between the user and the application and vice versa. The traffic we have been able to monitor represents the attacker's set up (see below) and not the traffic of victimized clients.

According to the data, the amount of XSS traffic is growing.

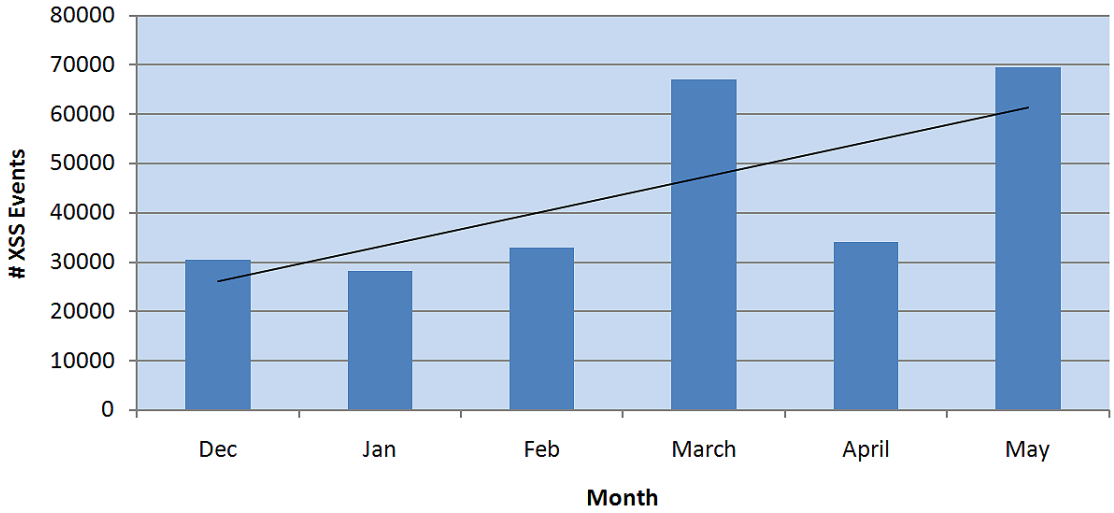


Figure 20: Cross Site Scripting traffic per month

There are several flavors of XSS attacks. An interesting variant of XSS injects specially crafted links to XSS-vulnerable sites into the results returned by search engines for popular queries. A user that issues such a query and selects one of the injected results is unknowingly redirected to a site controlled by the attacker (see our report that we cited earlier). This is a sophisticated attack that requires detailed preparations by the attacker. Apparently, the benefits are well worth the effort, since this attack has been going on continuously for at least 18 months.

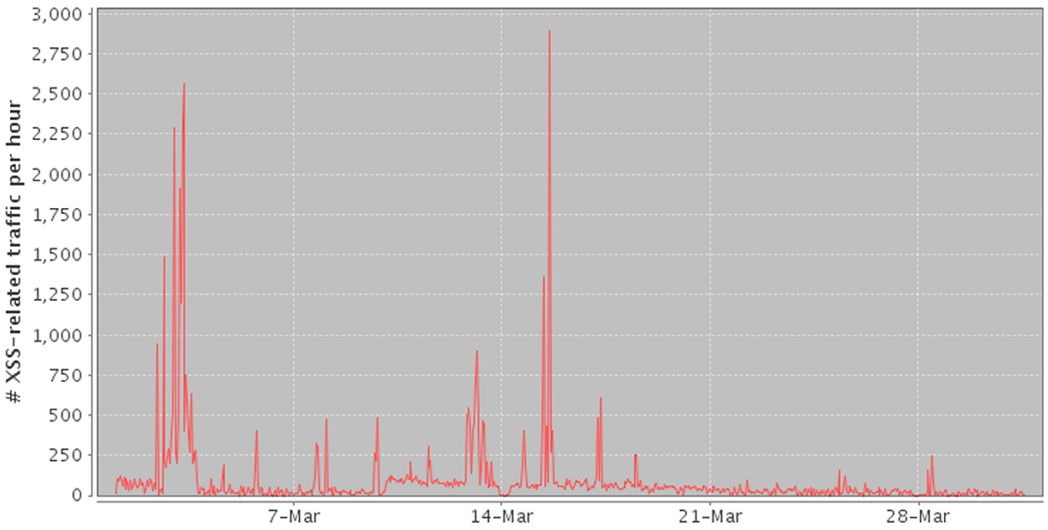


Figure 21: Cross Site Scripting traffic - March

We have observed waves of Search-Engine Poisoning (SEP) XSS traffic during the past 6 months, as seen in Figure 22.[26] This traffic is generated by automatic crawlers of search engines that pick up the maliciously-crafted links on the Internet. This occurrence pattern shows that periodically (approximately once a month) maliciously-crafted links are followed by the search engine's crawlers into XSS-vulnerable sites. This periodicity is due to a combination of the scan cycle of the search engine and the spreading of new malicious links on the internet by the attacker.

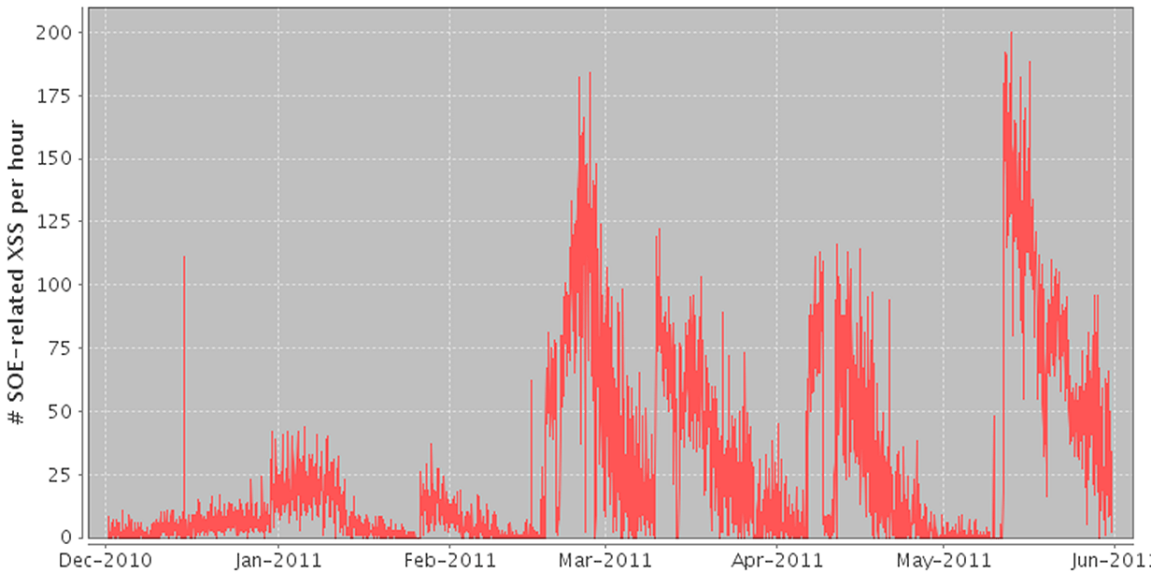


Figure 22: Search Engine Optimization XSS traffic

Figure 23 shows the distribution of search engine crawlers that we have observed following the maliciously-crafted XSS links. According to this comparison, Yahoo is the most susceptible to SEP-related XSS, while Google apparently filters most of these links.

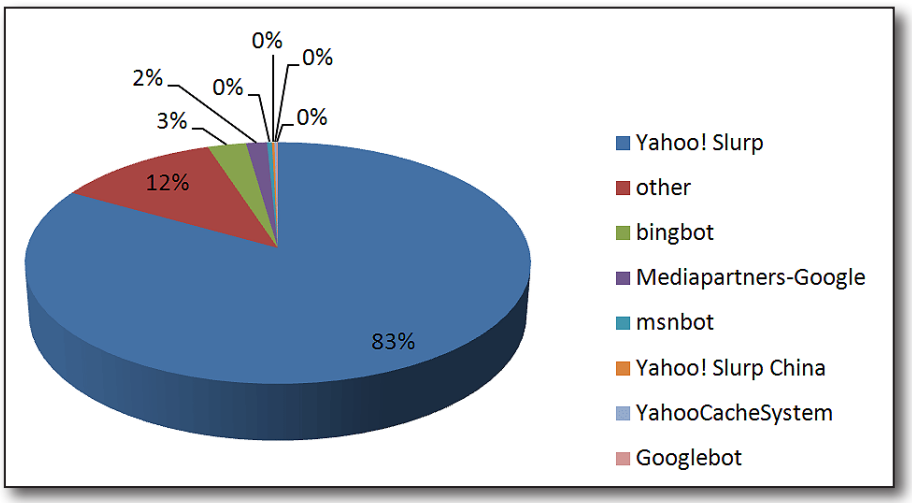


Figure 23: Search engine crawlers following XSS links

[26] http://www.imperva.com/docs/HI_Search_Engine_Poisoning_SEP.pdf

6 Recommendations

6.1 Technical Recommendations

1. **Deploy security solutions that deter automated attacks.**

   Automation is a key arsenal in a hacker's toolset. Trying to mitigate their attacks manually is like bringing a knife to a gun fight. It's impossible to manually handle attacks incoming at a rate of few events per second.

   Automation mitigation should be applied with care, since some web automation tools (such as search engine crawlers) are considered to be good and even crucial for the web application's functionality. This also means detecting protocol anomalies even if they are not considered malicious is essential. It can be the first sign of automated attack. Some of the attacks' scanning is vertical across the entire web application. Therefore, it's important to detect in real time reoccurring violation of the security policy from the same sources.

2. **Detect known vulnerabilities attacks**

   The security organization needs to be aware of known vulnerabilities and have an up-to-date list to know what can and will be exploited by attackers. We hope that this report will help with the prioritization process.

   Even if your application isn't vulnerable to the attack an early detection of malicious scans can assist in mitigating unknown "0 days" vectors included in the scan as demonstrated by the RFI signatures.

3. **Acquire intelligence on malicious sources and apply it in real time.**

   Having advanced knowledge on malicious sources that is integrated into your security solution – can help you stop them at the gate of your web application. For example, from our sampling, even knowing the top ten malicious sources for RFI can help you mitigate up to 40% percent of the malicious traffic seeking to exploit this vulnerability.

4. **Participate in a security community and share data on attacks.**

   Some of the attacks' scanning is horizontal across similar applications on the internet. Having a community that shares data would be beneficial for the early detection of automation and blocking of attacks.

5. **Detect automated attacks early.**

   It is very important and beneficial to detect automation attacks as a whole and not only treat it in a per request context. Quickly identifying thousands of individual attacks as one attack allows you to prioritize your resources (such as security analysts, complicated attack detection mechanisms, etc.) more efficiently. Furthermore, detecting the sources of an automated attack as malicious can help in the detection of previously unknown attack vectors (e.g., "0 days") included in the attack.

   To summarize, automated attack detection requires collecting data, combining it and then analyzing it automatically in order to extract relevant information and apply security countermeasures. Gathering the required data requires monitoring protocol anomalies even if they are not malicious or if the web application is not vulnerable. Combining this data with intelligence gathered on known malicious sources will help enlarge the knowledge base for identifying attacks and selecting appropriate attack mitigation tools.

6.2 Nontechnical Recommendations: CEO Checklist

The McKinsey Quarterly recently investigated cyber security, saying "Eliminating threats is impossible, so protecting against them without disrupting business innovation and growth is a top management issue."[27] Our technical recommendations won't amount to anything if CEOs (or the equivalent) does not take responsibility to protect data.

1. **Assume you're a target and have already been compromised.** Consider yourself a target if you have an application. Consider yourself an even more attractive target if you hold sensitive information with value for hackers, governments, employees or competitors. Producing a full inventory of what is attractive and how it can be abused by external or internal threats will be sobering.
2. **Make data security a strategic priority.** For example, and admittedly at the top end of the spectrum, a major finance company that is a frequent target of cyber attacks, codified an approach to security saying, "We're not a bank, we're a security company that does banking." This may seem extreme. However, would any of the recently breached organizations wished they'd have adopted such an attitude before the hack?
3. **Give security a seat at the table.** Organize the company for security. To enhance security's authority and stature, some firms have security reporting to the CEO or the board of directors. Another firm decided to put cyber security in into every technology decision and reversed conventional wisdom by having IT report into security, instead of vice versa.
4. **Work with law enforcement.** Companies have worked with the FBI and state departments to help pinpoint hackers, even overseas, to ensure that the weeds don't grow back. The NSA has even put together a program called "Perfect Citizen" designed to aggregate information about cyber attacks and help private organizations improve their security posture. What may seem like a minor cyber attack could be part of a larger criminal effort that only law enforcement can recognize. In 2010, law enforcement took down the servers disseminating spam – dropping spam levels by 30% overnight.
5. **Embrace data security regulations.** The credit card industry, for instance, regulated itself and created the payment card industry data security standard (PCI-DSS). It's working. PCI forced companies transacting credit cards to implement the basic elements of data security. And something has proved better than nothing: a Ponemon survey on the topic from 2011 showed that companies complying with PCI were twice as likely to avoid breaches as noncompliant firms.
6. **Put the right technology in place.** By design, this is last on the list. Without having pursued the above steps technology can't help. In today's world, this does not mean "we have updated our anti-virus and put in place a network firewall." Rather, it means "we have identified all sensitive data and have put in place technology with the audit and protection capabilities required to safeguard that data."

[27] https://www.mckinseyquarterly.com/Meeting_the_cybersecurity_challenge_2821

7 Authors and Acknowledgements

This report was assembled by Imperva's Application Defense Center. Specifically, we'd like to call out Tal Be'ery and Nitzan Niv. Amichai Shulman, Robert Rachwald and Noa Bar Yosef also contributed to this effort.

We'd like to extend a huge thanks to Stephanie Rogers for analyzing mountains of data with very compressed timelines. Everyone should be so lucky to have someone so competent and hard working who puts in global hours. (Go Bears!).

Imperva
Headquarters
3400 Bridge Parkway, Suite 200
Redwood Shores, CA 94065
Tel: +1-650-345-9000
Fax: +1-650-345-9004

Toll Free (U.S. only): +1-866-926-4678
www.imperva.com