



# WEB APPLICATION ATTACK REPORT #5

Edition #5 - October 2014

## PHP APPLICATIONS

3X as many Cross Site Scripting Attacks as .NET APPLICATIONS

## WORDPRESS WEBSITES

Were attacked 24.1% more than ALL OTHER CMS PLATFORMS COMBINED

## RETAIL WEBSITES

Were targeted by 48.1% of all attack campaigns

## 2014

Attacks are 44% longer in comparison to 2013

## AWS SERVERS

Originated 20% of all known vulnerabilities (CVEs) exploitation attempts

## 2014

Increase of 24% in Remote File Inclusion (RFI) attacks 2013

# Table of Contents

<b>1 Introduction.....</b>	<b>3</b>
<b>2 Key Finding Explanation.....</b>	<b>4</b>
2.1 Threat Growth.....	4
2.2 Targets.....	5
2.2.1 - Retail applications suffer the most from web application attacks; financial institutes second.....	5
2.2.2 - WordPress is the most attacked Content Management System (CMS) .....	5
2.2.3 - PHP applications suffer three times as many Cross Site Scripting attacks as .NET applications.....	6
2.2.4 - Websites containing some form of consumer information suffer up to 59% of the attacks.....	7
2.3 Sources.....	7
2.3.1 - The United States generates the majority of the web application attack traffic world-wide.....	7
2.3.2 - Infrastructure as a Service (IaaS) providers are on the rise as attacker infrastructure due to market share growth.....	8
<b>3 Analysis Methodology.....</b>	<b>9</b>
3.1 Data Corpus.....	9
3.2 Data Analysis and Presentation Methodology .....	9
3.3 Analysis Glossary .....	9
<b>4 Analysis Results.....</b>	<b>10</b>
4.1 Comparative Attack Overview.....	10
4.1.1 - Web Application Participation .....	10
4.1.2 - Number of Attack Incidents .....	11
4.1.3 - Attack Incident Magnitude .....	12
4.1.4 - Attack Incidents Duration.....	13
4.1.5 - Battle Days .....	14
4.1.6 - Requests per Battle Day .....	15
4.1.7 - Reputation Based Traffic .....	15
4.2 Last Year Report Comparison .....	16
4.2.1 - Web Application Participation per Attack Type.....	16
4.2.2 - Number of Incidents .....	17
4.2.3 - Number of Battle Days.....	19
4.3 Data Slicing.....	20
4.3.1 - Comparing Vertical Industries .....	20
4.3.2 - Comparing Frameworks .....	21
4.3.3 - Comparing Content Management Systems .....	23
4.4 Geographic Dispersion.....	25
4.4.1 - Traffic Volume .....	25
4.4.2 - Attacking Hosts .....	27
<b>5 Conclusions and Recommendations.....</b>	<b>28</b>
<b>6 Attack Glossary.....</b>	<b>29</b>
6.1 SQL Injection.....	29
6.2 Remote File Inclusion .....	29
6.3 Local File Inclusion .....	29
6.4 Directory Traversal.....	29
6.5 Cross-Site Scripting.....	29
6.6 Comment Spamming .....	29
<b>7 References.....</b>	<b>30</b>

# 1 Introduction

Welcome to Imperva's annual Web Application Attack Report #5 (WAAR). This report contains a thorough analysis of data accumulated over time. Ideally, there would be no need for our annual Web Application Attack Report. If all of the other layers of security products at work within most enterprises were living up to their promise, there would be no need for a Web Application Firewall (WAF), the last frontier before the actual application being protected. But even with all those layers protecting the endpoint, the network, and everything in between, threats still sneak through to the application, and Imperva's SecureSphere WAF is kept way too busy as the last line of defense before the application itself. So, until all security products are perfect and/or applications are able to heal themselves instantly if attacked, there will be a WAAR. It is our hope that the findings in our report – especially when combined with other vendors' security reports – enable security practitioners to better anticipate, see, and take action to mitigate cyber threats.

## Our Approach

In this report, our Application Defense Center (ADC) group has analyzed attacks on a subset of 99 applications of 300 applications protected by Imperva's Web Application Firewalls, over a time period of nine months, from August 1, 2013 to April 30, 2014. This subset was comprised of the 99 applications that experienced at least one attack incident during the nine-month period. This is in order to provide an accurate picture of today's application threat landscape, as seen in the wild, by looking at how attackers behave.

Imperva conducts this annual analysis report with a goal of defining the critical risk elements and techniques used by hackers during that period of time. We aim to educate business owners and consumers on the critical issues that may hurt their public infrastructure.

### Key Findings:

#### 1. Threat Growth

- 1.1 Malicious traffic exposure has increased significantly year-over-year. The following substantial increments were observed:
  - 1.1.1 An increase of 10% in SQL Injection (SQLi) attacks
  - 1.1.2 An increase of 24% in Remote File Inclusion (RFI) attacks
- 1.2 Attacks are 44% longer in comparison to the prior period reviewed (i.e., from June 1, 2012 to November 30, 2012).

#### 2. Targets

- 2.1 Retail applications suffer the most from web application attacks; financial institutions second.
  - 2.1.1 48.1% of all attack campaigns targeted retail web sites, while 10% targeted financial institutions.
  - 2.1.2 40% of all SQL injection attack campaigns targeted retail web sites.
  - 2.1.3 64% of all malicious HTTP traffic (protocol violations and malformed requests) was targeted at retail websites.
- 2.2 Wordpress is the most attacked Content Management System (CMS)
  - 2.2.1 Websites running Wordpress were attacked 24.1% more than websites running on all other CMS platforms combined.
  - 2.2.2 Wordpress suffers 60% more Cross Site Scripting (XSS) incidents than all other CMS-running websites combined.
- 2.3 PHP applications suffer three times as many Cross Site Scripting Attacks as .NET applications.
- 2.4 Websites containing some form of consumer information suffer up to 59% of the attacks

#### 3. Sources

- 3.1 The United States generates the majority of the web application attack traffic world-wide.
  - 3.1.1 The US generates the majority of web attacks
  - 3.1.2 The majority of attacking hosts reside in the US
- 3.2 Infrastructure as a Service (IaaS) providers are on the rise as attacker infrastructure
  - 3.2.1 We look into Amazon AWS as a proxy of this space for our case study
  - 3.2.2 20% of all known vulnerabilities (CVEs) exploitation attempts have originated from AWS servers
  - 3.2.3 10% of all SQL Injection attacks observed originated from AWS servers

## 2

## Key Finding Explanation

### 2.1 - Threat Growth

#### Malicious traffic exposure has increased significantly year-over-year

In this part of the research, we compared data from our last report (i.e., June 1, 2012 to November 30, 2012) with data from this report (i.e., August 1, 2013 to April 30, 2014) to assess changes in attack magnitude and velocity. If an application has experienced at least one attack incident during the studied timeframe, we start tracking it against what we call “battle days.” (You’ll see this term used frequently in our report.)

When we compared this year’s data to last year’s WAAR report ([WAAR #4](#), July 2013), we noticed significant changes in attacker behavior, over time.

We found that attacks have increased in duration since our last report. For example, a sample of applications which had to cope with 16 Battle Days of SQL Injection attacks last year suffered 23 Battle Days in this period, representing a 44% increase.

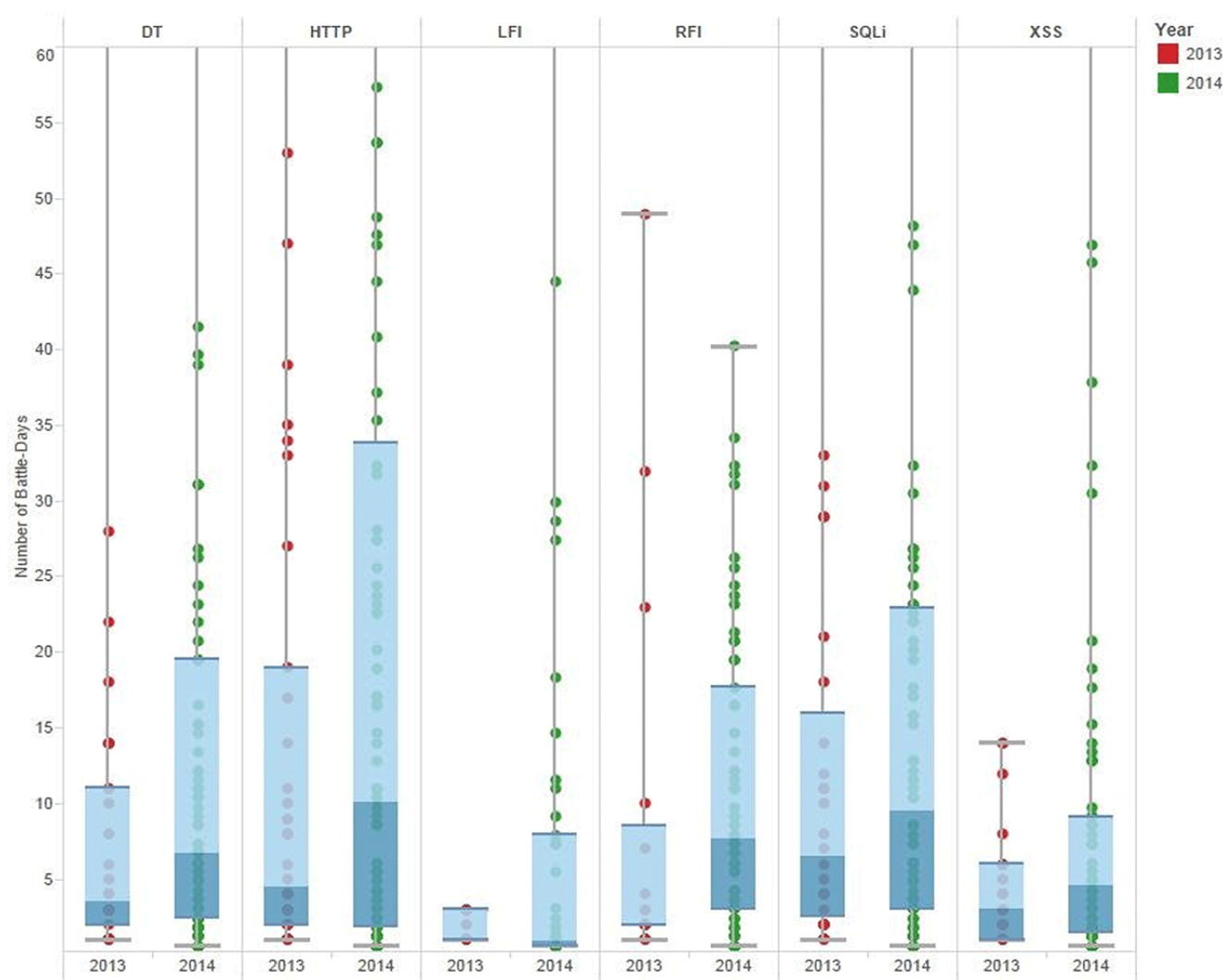


Figure 1 – Malicious Traffic Exposure

There has also been a significant increase in attack vector incidents, most notably Remote File Inclusions (RFI), which have increased from 38% to 62% year-over-year.



## 2.2 - Targets

In this section, we discuss the findings that show how hackers are focused on certain platforms and infrastructure pieces more than others. Since this data comes from gathering real attacks in real time, it shows how the hacker community behaves in respect to online web applications.

### 2.2.1 - Retail applications suffer the most from web application attacks; financial institutions second

One of the key elements involved in determining trends in hacking is to map attack types to volume and duration, and then to align the findings to significant verticals in order to point out who is a likely target of a web attack. In our effort to do so we mapped our attack data to different verticals and then collapsed verticals that either had small results, or were not distinct enough to come to a conclusion. At the end of this exercise, what really stood out were the Retail and Finance verticals; not too surprising considering the high-profile attacks on Target, Neiman Marcus, Michael's, and many other retail chains in the past year, as well as on JP Morgan Chase very recently, and countless other banks and credit card services.

In the following chart, we visualize some of the key attack categories for which attack campaigns were mapped.

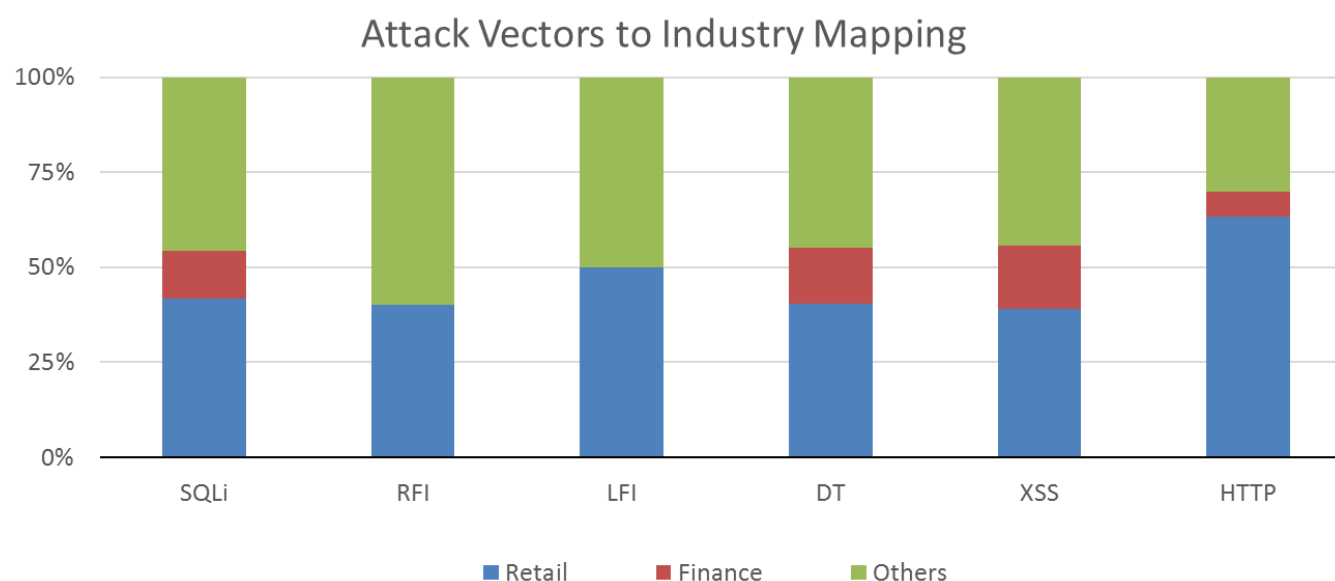


Table 1 – Key Attack Categories for Attack Campaigns

### 2.2.2 - WordPress is the most attacked Content Management System (CMS)

WordPress has been in the headlines, in the past couple of years, both because of its popularity, and because of the amount of vulnerabilities found in its application and exposed by hackers. We believe that popularity and a hacker's focus go hand-in-hand. When an application or a platform becomes popular, hackers realize that the ROI from hacking into these platforms or applications will be fruitful, so they spend more time researching and exploiting these applications, either to steal data from them, or to use the hacked systems as zombies in a botnet.

This trend was first observed in our 2013 "CMS Hacking" [research](#) and was also a key finding of the [Verizon DBIR 2014](#) report, which highlights: "Many of the attacks in our 2013 dataset targeted off-the-shelf content management systems (e.g., Joomla!, Wordpress, or Drupal) to gain control of servers for use in DDoS campaigns."

This year, we've added a chapter that shares the results of our comparative analysis of CMS platforms, and specifically compares WordPress to the other popular CMS systems.

As we can see in the chart below, websites that use WordPress suffer 24.1% more attacks than all other CMS systems combined.

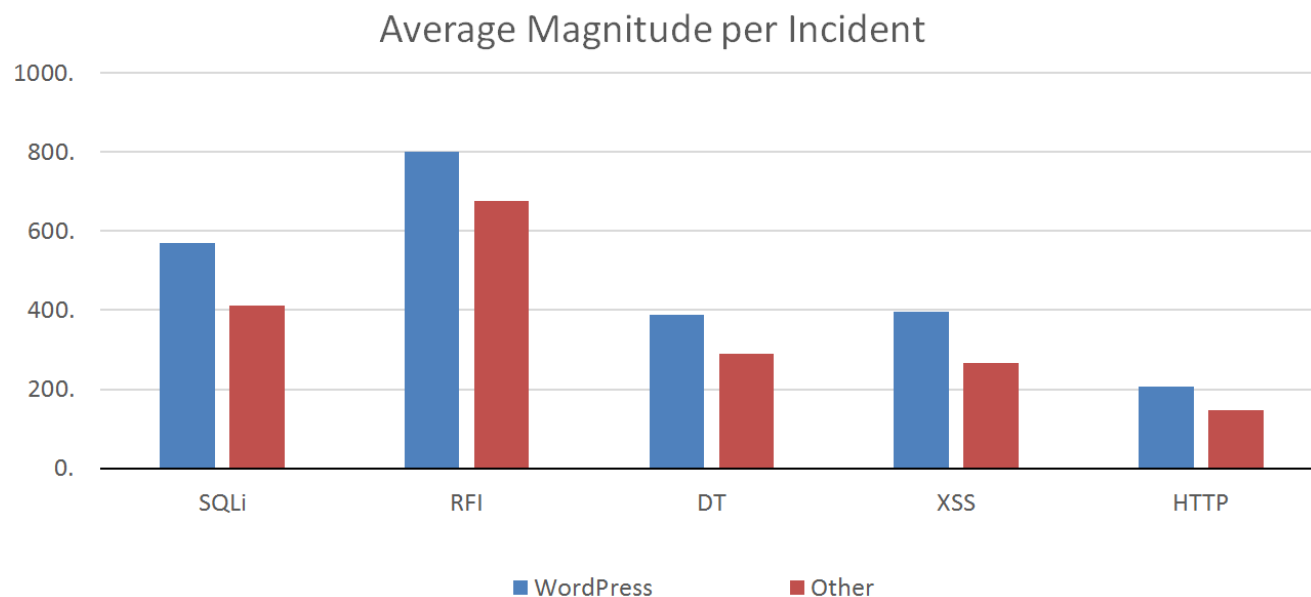


Table 2 – Attacks on WordPress Sites Versus Other CMS Systems

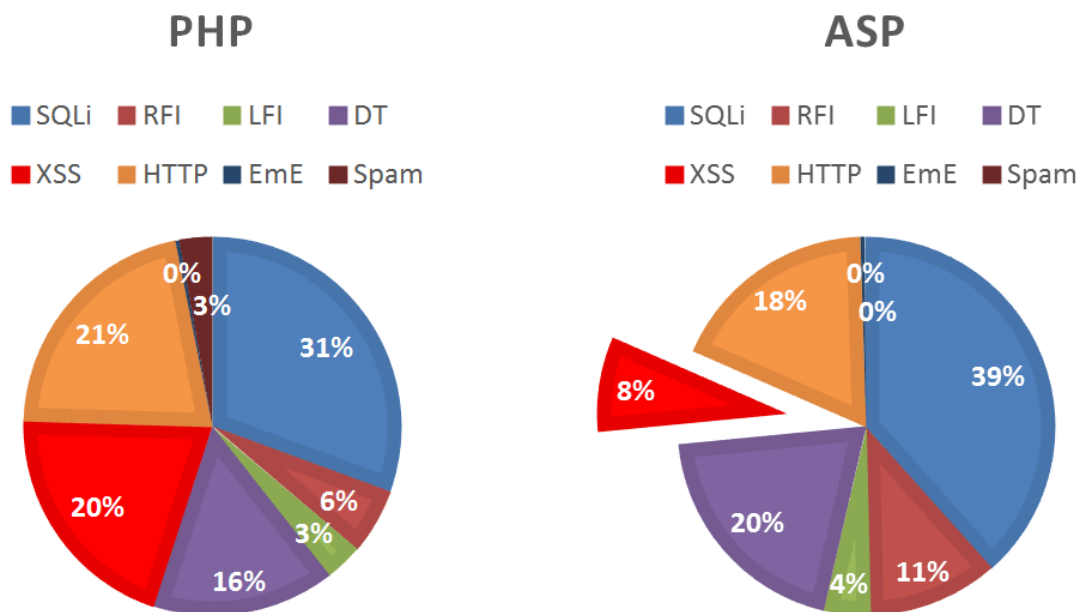
### 2.2.3 - PHP applications suffer three times as many Cross Site Scripting attacks as .NET applications

A new addition to this year's data analysis is infrastructure mapping. We have decided to add this element due to the perceived difference in attacker behavior and defender challenges for applications running on different infrastructures, such as PHP, ASP, etc.

Since 99.5% of the web applications, reviewed for purposes of this report, use PHP or ASP as web application frameworks, we chose to focus on that specific subset.

From our data analysis we learned that:

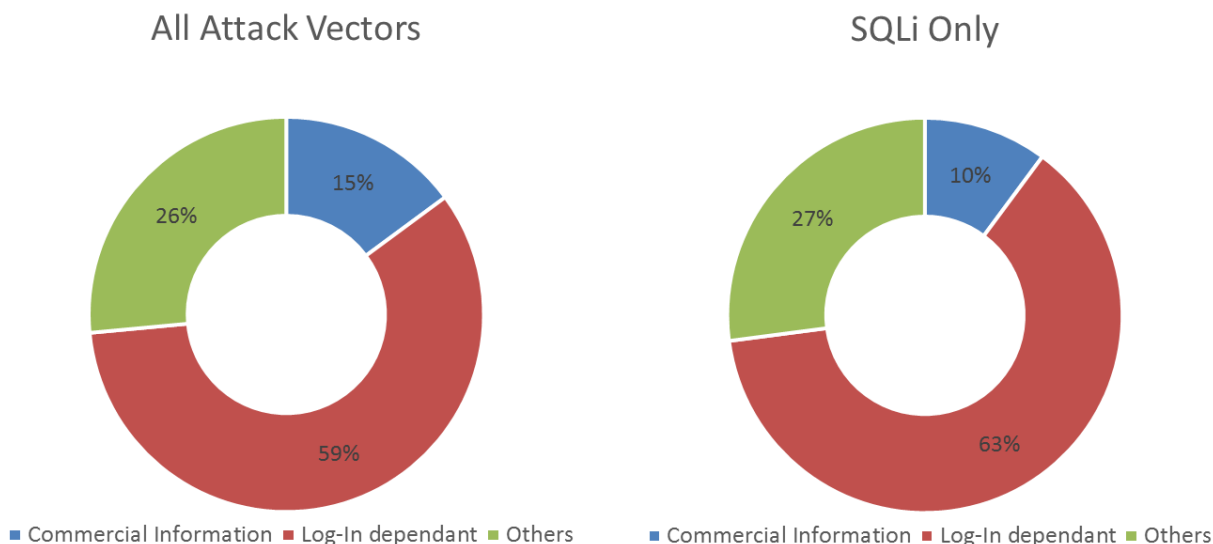
- PHP applications suffer almost 3X more Cross Site Scripting (XSS) attacks than ASP applications.
- PHP applications suffer almost 2X more Directory Traversal (DT) attacks than ASP applications.
- ASP applications suffer almost 2X more SQL Injection attacks than PHP applications.



### 2.2.4 - Websites containing some form of consumer information suffer up to 59% of the attacks.

One of the interesting mappings we analyzed was to compare applications that require some level of interaction with personal user information. We categorized all sites that required some form of *log in* to the website (meaning that consumers are required to have credentials and user information submitted at one stage or another with the website, in order to interact). From there, we divided sites into those which offered *Commercial Information*, such as product information, company portfolio, etc., and those that we put in an *Others* category, which are sites that may have online interactive offerings, but do not require any form of user authentication in order to interact.

In the charts below, you'll see that websites which have log-in functionality, and hence contain consumer specific information, suffer 59% of all attacks, and 63% of all SQL Injection attacks.



If we look into data breaches from 2013 and 2014 containing personal information that analysis indicates were performed via a web application attack, the hacker motivation is clear. Consumer information such as personal details and credit cards are a valuable and tradable black market piece of information, and the observed hacker activity shows just that.

## 2.3 - Sources

In this section, we discuss where attacks are coming from. Imperva has a benefit of observing attacks in real time and therefore is able to see real attackers, their actions, where they come from and how their behavior is trending.

### 2.3.1 - The United States generates the majority of the web application attack traffic world-wide.

It's always interesting to map all of the attack information against our Geo IP database and map attacks to countries of origin.

This year, we decided to run this exercise in two different ways:

1. Determine attack vectors against geography mapping to analyze which countries generate the most attacks of a certain type
2. Determine aggressiveness of attacking Hosts by correlating what attacks originate from which IP address, and then geographically mapping those IPs

The results show that the United States has the most attacks per vector and the most attackers (hence aggressiveness) per vector, with one exclusion – Cross Site Scripting, which is being dominated by the United Kingdom.

In our educated opinion, based on years of analyzing attack data and origins, we propose that attackers from other countries are using U.S. hosts to attack, based on those hosts being geographically closer to targets.

While this may be overwhelming, we believe that there is more to this picture. Attacks originating in the U.S. may indicate other things such as TOR exit nodes, Botnet infected machines, etc., and so this information needs to be looked at in proportion. What it potentially teaches us is the quality of targets. It makes sense for an attacker to execute the attack as close to the target as possible, to remain undetected or to maximize the available bandwidth of the attack. So while not a proven conclusion, it makes sense that attackers are using U.S. hosts to attack, based on relative distance from the target.

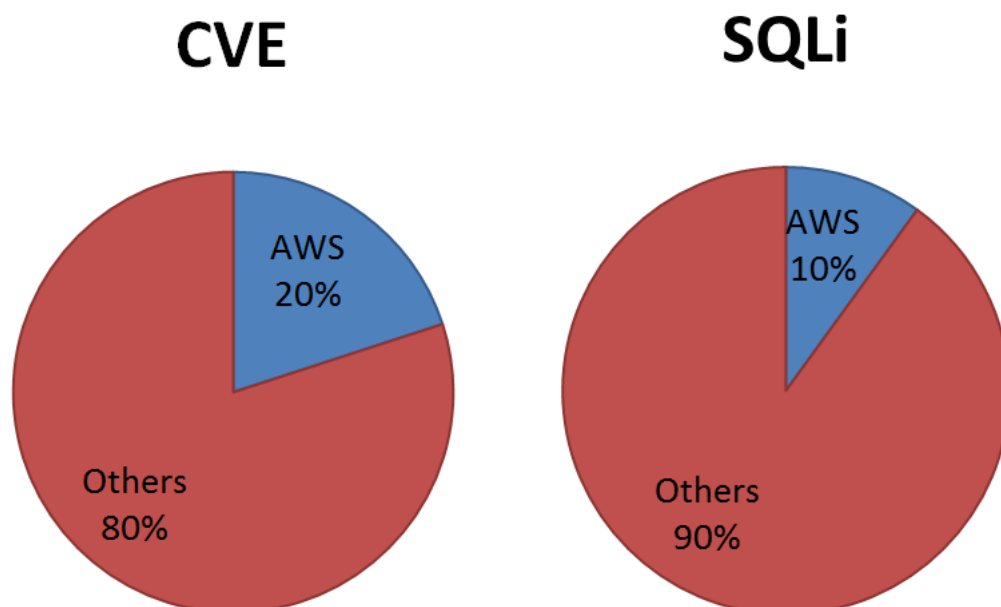
### 2.3.2 - Infrastructure as a Service (IaaS) providers are on the rise as attacker infrastructure due to market share growth

As part of our meta-analysis on top of our data, one thing we wanted to explore this year was whether we would find any trends around attacks originating in cloud-hosted and infrastructure as a service (IaaS) hosted applications and servers. Many companies have endorsed “cloudification” of IT, moving assets into infrastructures such as Amazon Web Services, Microsoft’s Azure, and others. We chose Amazon AWS as a proxy for this case study, as it is clear to us that Amazon AWS is the clear leader in size and market share, having more than 5 times the size of all of their 14 competitors combined. This reflects a huge and growing portion of all online entities and therefore the impact we see on our own numbers.

In order to analyze the data, we mapped the source attacker host/IP to AWS’s IP ranges list, which is publicly available through their service. We calculated the ratio between all of the attacks in the world, per their attack type and separating out the ones that originated from AWS.

The following ratios were discovered:

- 20% of all known vulnerability (CVE) exploitation attempts originated in AWS source IPs
- 10% of all SQL Injection attempts originated in AWS source IPs



To better understand what these numbers mean we need to explain our thesis about the AWS-originated incidents. We believe that these incidents could be classified into three categories in order to better understand who the attackers are:

- Attackers who have breached the web application directly and have injected malicious code that allows remote control of the server
- Attackers who have taken over the company’s AWS account and have tampered with the server through the administrative interface
- Attackers who simply use AWS as a platform for their malicious server instances, in this case the attacker is the account owner

This is an interesting beginning to what could become a trend in security, where attackers use cloud-provided public infrastructure, either by employing it themselves, or by hijacking some other organization’s account, and elevating attacks from these platforms. Since cloud infrastructure is usually designed to be highly available and distributed globally, it makes it very effective for hackers to elevate large scale attacks such as DDoS attacks from these platforms.



## 3

## Analysis Methodology

### 3.1 - Data Corpus

This report is based on monitoring 300 web applications and analyzing 99 of them, over a period of nine months, from August 1, 2013 to April 30, 2014. Automated tools recorded the web applications traffic, and malicious events were documented in log files. Application Defense Center (ADC) security researchers analyzed this data using a special-purpose software and knowledge base.

### 3.2 - Data Analysis and Presentation Methodology

The analysis and presentation methodology in this report follows the previous 2013 [WAAR #4](#) report. We defined an attack incident to consist of at least 30 requests in five minutes (on average, one attack request every 10 seconds). Thus, a single attack may span several consecutive five-minute periods. We also defined an even broader concept of "battle days", which are days in which at least a single attack incident targeted an application.

Attacks against a web application can be measured according to several criteria, all having practical security implications:

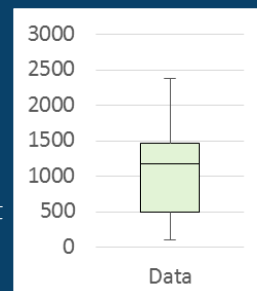
- How many attack campaigns occur in a given period of time?
- How long does each attack last?
- How intense is an attack campaign; that is, how many HTTP requests are issued as part of the attack?
- If an application suffered an attack incident yesterday, how likely will they suffer an attack today?

We used statistical analysis to answer these questions. However, statistics can be misleading. Benefiting from the experience of preparing previous WAARs, we concluded that we need more insightful statistical analysis of our data. Web attacks have asymmetric distributions, with rare but significant outliers. A security manager who prepares for an "average" number of attacks or a typical attack duration will be unprepared for an intense attack. Therefore, we use relevant descriptive statistics, such as those from the median and quartiles, rather than averages and standard deviation. Graphically, we present the resulting numbers using box-and-whisker plots.

After analyzing nine months of application attacks, we uncovered some results that reinforced existing trends and statistics and other results that deviated dramatically from previous reports. Based on our examination of web attack methods, attack sources, and incident intensity and duration, organizations' security teams can prioritize their efforts and develop plans to improve their security posture.

#### Box-and-Whisker Plots

The Box-and-Whisker plot is a convenient way to present variations of statistical data. The bottom and the top of the box represent the first and third quartiles; the middle line represents the median. We chose the end of the whisker to represent the 95% value of the data, and the lower whisker to represent the lowest value.



### 3.3 - Analysis Glossary

- **Attack request** – A single, malicious HTTP request
- **Attack incident** – A burst that exceeded the rate of 30 attack requests per a five minute period
- **Attack incident magnitude** – The number of attack requests per attack incident
- **Attack incident duration** – The length, in minutes, of an attack incident
- **Battle day** – A day in which an application experienced at least one attack incident
- **Effective Period** – from August 1, 2013 to April 30 2014 - the time period of the current report
- **Previous Period** – from June 1, 2012 to November 30 2012 - the time period of WAAR #4.

#### Report Time Period

This WAAR #5 analyzes the attacks during a period of nine months, longer than the period of WAAR #4 which spanned only six months. For the sake of comparison, whenever we compare absolute numbers WAAR #4 between the reports, e.g., quantities of attack incidents or battle days, we normalize the numbers.

## 4 Analysis Results

### 4.1 - Comparative Attack Overview

In this section, we summarize the characteristics of each attack type across the monitored applications. We focus on the following attack types: SQL injection (SQLi), Remote File Include (RFI), Local File Include (LFI), Directory Traversal (DT), Cross-Site-Scripting (XSS), Comment Spam (Spam) and HTTP. The HTTP attack type includes protocol violations which often indicate automated threats, evasion techniques and denial of service attacks.

We begin this a comparative attack overview in section 4.1 and provide global attack trends for web applications. We continue in section 4.2 with a comparison to WAAR #4. In section 4.3 we present attack patterns across different vertical industries, frameworks and content management systems. Finally, we present in section 4.4 the geographic dispersion of the attacks.

#### 4.1.1 - Web Application Participation

Table 3 shows for each attack type the ratio between web applications that had been exposed to malicious traffic, to ones that were not exposed.

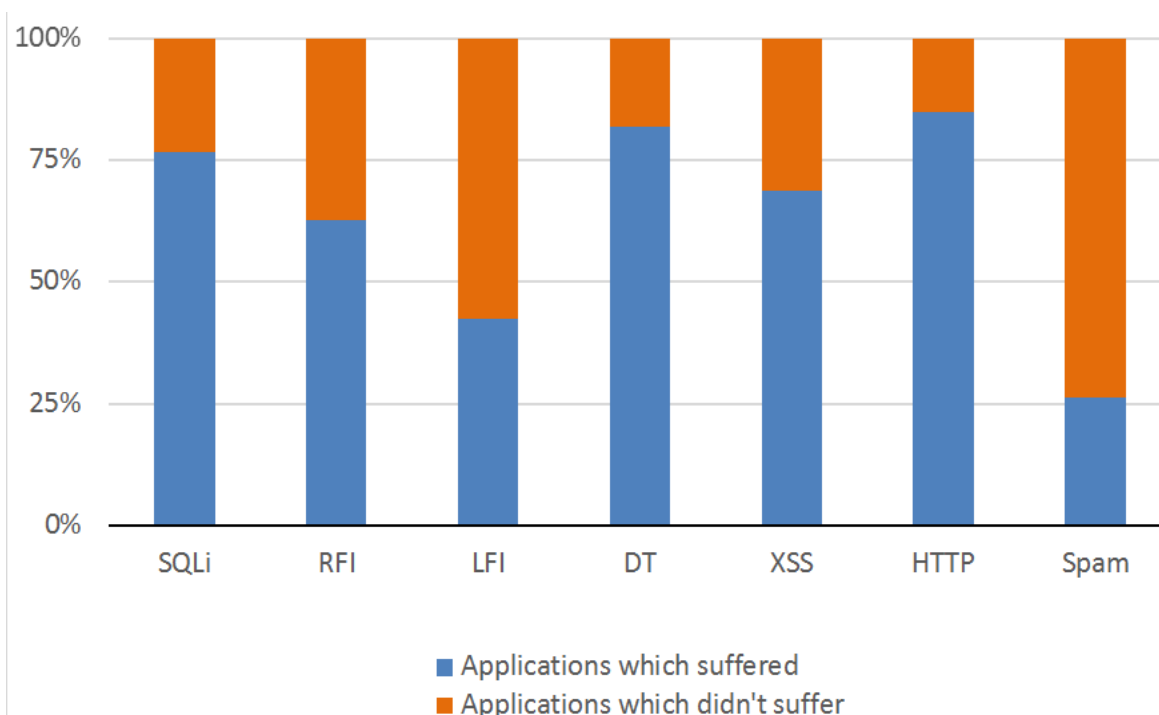


Table 3 – Web application Malicious Traffic Exposure Ratio

The exposure of web application to Comment Spam and LFI attacks are low (26% and 42%, respectively), meaning that most of the web application had not suffered from Comment Spam and LFI attacks during the effective period.

#### 4.1.2 - Number of Attack Incidents

Table 4 and Figure 2 show the distribution of the number of incidents per attack type during the effective period. This analysis shows global trends in attacking web applications.

	SQLi	RFI	LFI	DT	XSS	HTTP	Comment Spam
Median	7	2	0	6	2	7	0
Max	1561	135	290	1256	393	603	66
1st Quartile	1	0	0	1	0	1	0
3rd Quartile	31	13	1	24	10	39	1

Table 4 – Number of Attack Incidents

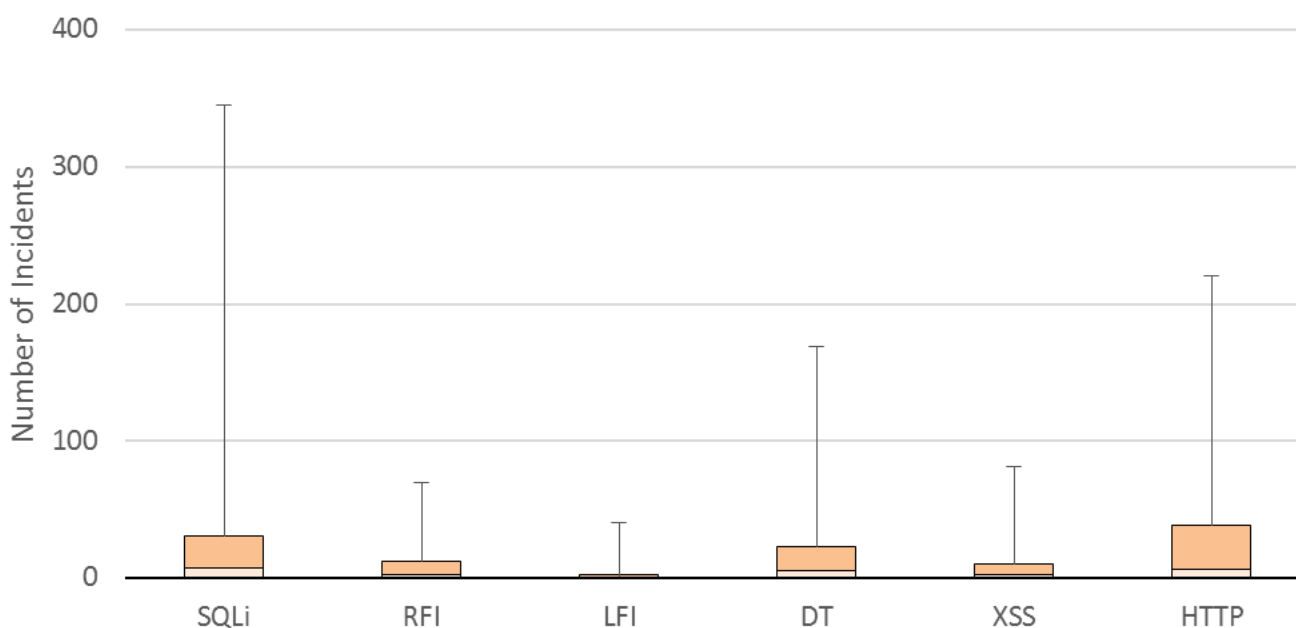


Figure 2 – Number of Attack Incidents

The main observation from these results is that similarly to WAAR #4, the most prevalent types of attacks are SQL Injection, Directory Traversal and HTTP protocol violations, which stand out in all measured percentiles. In particular, more than 75% of the applications had experienced SQL Injection attempts and Directory Traversal attempts.

Another interesting observation is that the application which was most frequently attacked had suffered from an average rate of almost 10 attacks per day, during the effective period of nine months.

### 4.1.3 - Attack Incident Magnitude

Attack incidents, which are bursts of malicious HTTP requests, vary significantly in their magnitude - number of requests, and their duration (duration analysis is presented in section 4.1.4 below). Table 5 and Figure 3 show the distribution of the magnitude of attack incidents for each attack type.

	SQLi	RFI	LFI	DT	XSS	HTTP	Comment Spam
Median	81	961	46	91	95	67	41
Max	57235	19945	874	47555	12012	38340	1253
1st Quartile	42	169	36	49	38	41	32
3rd Quartile	160	1092	65	274	412	148	59

Table 5 – Magnitude of Attack Incidents

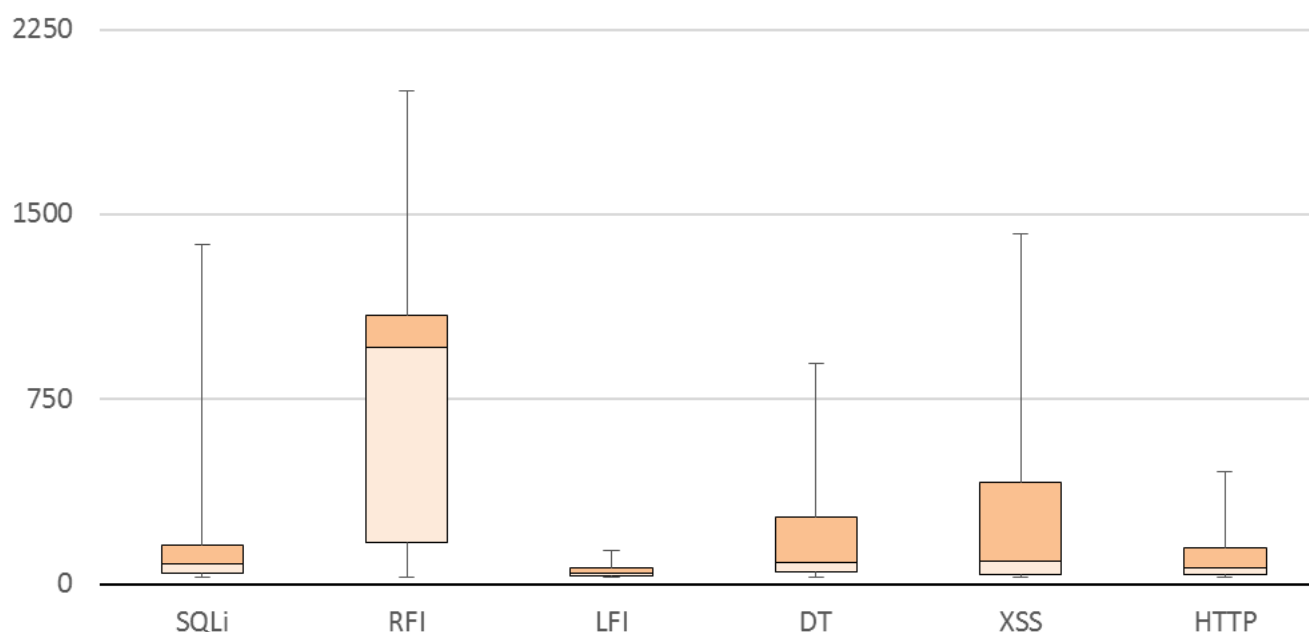


Figure 3 – Magnitude of Attack Incidents

The main observation from these results is that RFI attack incidents, which were shown to be relatively few when compared to other attack types, have significantly larger magnitude, with half of the incidents including more than 900 requests. At the other end, for LFI and Comment Spam, 75% of the attack incidents included no more than 70 requests.

Another interesting observation is that for SQLi, RFI, DT, XSS and HTTP, there were attack incidents where the target web application faced several dozen thousands of attack requests.

#### 4.1.4 - Attack Incidents Duration

Attacks usually occur in bursts. This trend was demonstrated in WAAR #4 and is demonstrated in this report once again. Table 6 and Figure 4 show the statistical descriptive of the duration of incidents for each attack type.

	SQLi	RFI	LFI	DT	XSS	HTTP	Comment Spam
Median	5	10	5	5	5	5	5
Max	1160	160	15	775	675	1105	45
1st Quartile	5	5	5	5	5	5	5
3rd Quartile	10	20	5	10	10	5	5

Table 6 – Duration of Attack Incidents

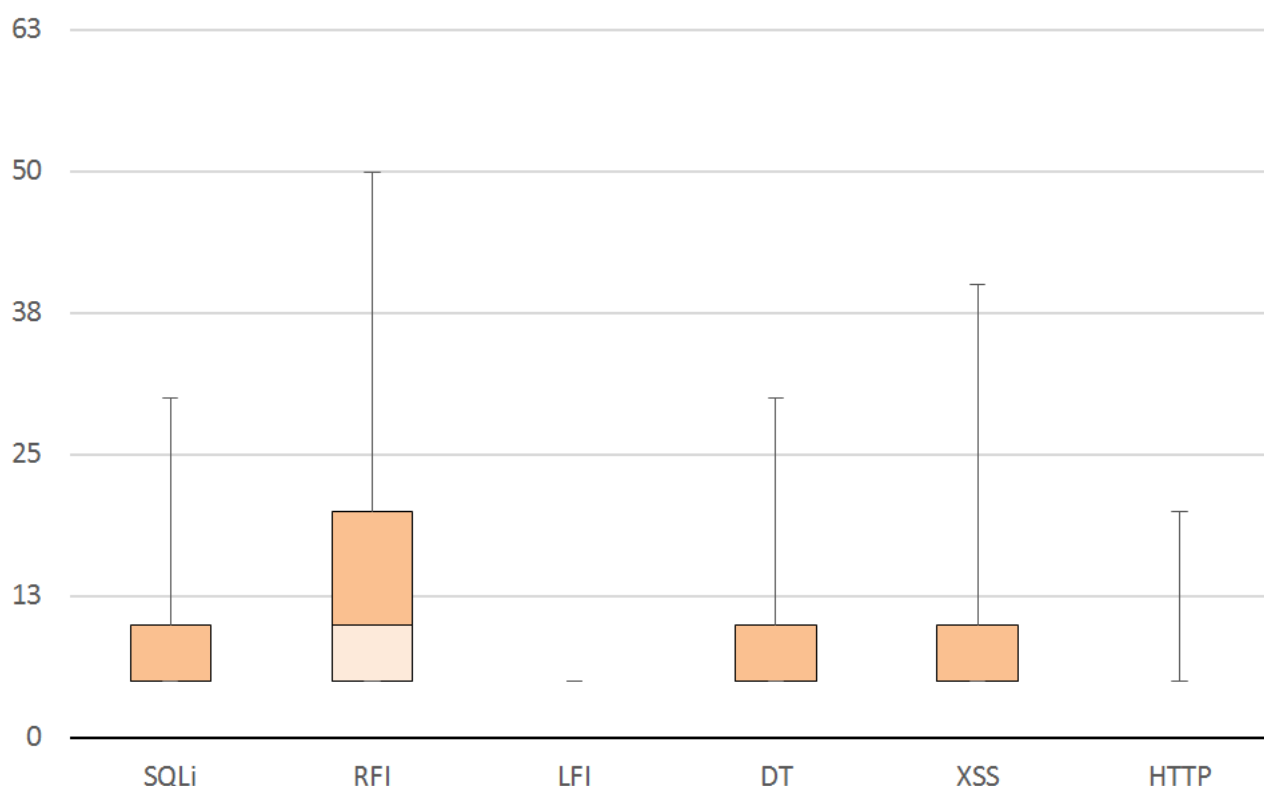


Figure 4 – Duration of Attack Incidents

The most significant observation is that most attack incidents last less than 10 minutes.

Unsurprisingly, there is strong correlation between incident magnitude and incident duration. RFI attack incidents, which have the highest magnitude also have duration that is significantly higher than incidents of other attack types. Most of the RFI attack incidents last longer than those for SQLi, DT, XSS and HTTP. Still, the longest incident of an RFI attack is significantly shorter than the one for other attack types.

#### Short Time Durations

The durations of LFI, HTTP and Comment Spam attacks are so short that 75% of their incidents lasted less than five minutes - our minimal incident duration. Thus LFI and HTTP don't have a proper "box" in Figure 4.



#### 4.1.5 - Battle Days

The above statistical analysis of attack incidents may be significantly affected by a few campaigns that include consecutive sequence of attacks. In order to reduce the impact of such campaigns, we use the notion of battle days.

Table 7 and Figure 5 show the number of days in which a web application suffered attack incidents during the effective period. As expected, we find strong correlation with the attack incident analysis from section 4.1.2 above, with SQL Injection, Directory Traversal and HTTP attacks being the most dominant attacks.

	SQLi	RFI	LFI	DT	XSS	HTTP	Comment Spam
Median	5	2	0	5	2	5	0
Max	153	40	71	137	95	123	40
1st Quartile	1	0	0	1	0	1	0
3rd Quartile	19	9	1	14	6	27	1

Table 7 – Battle Days

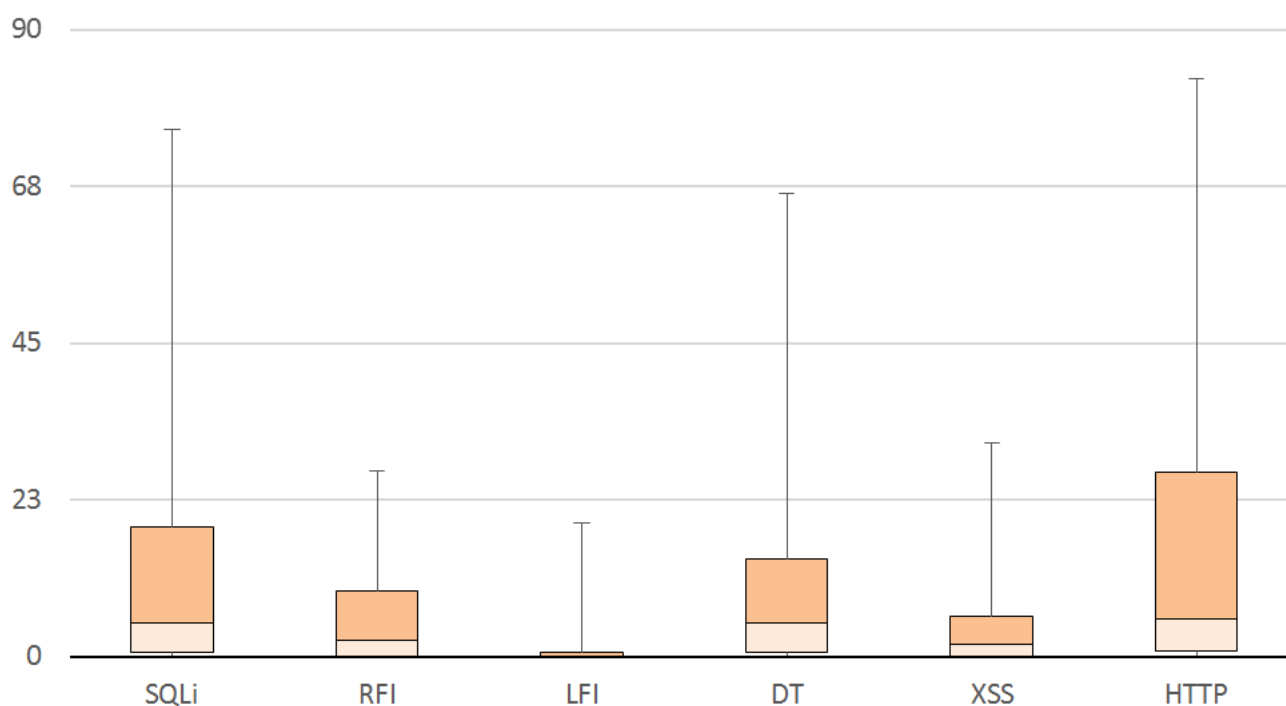


Figure 5 – Battle Days

#### 4.1.6 - Requests per Battle Day

High number of battle days (Section 4.1.5) do not necessarily hold the highest traffic volume intensity. Table 7 and Figure 6 show the HTTP requests per battle day for each attack type. This measure captures the volume of malicious traffic an application suffers from during battle days.

	SQLi	RFI	LFI	DT	XSS	HTTP	Comment Spam
Median	366	1086	80	261	233	138	50
Max	62321	22309	1625	69596	135177	49579	1253
1st Quartile	110	745	43	83	77	60	34
3rd Quartile	1131	1991	140	651	892	345	81

Table 8 – Requests per Battle Day

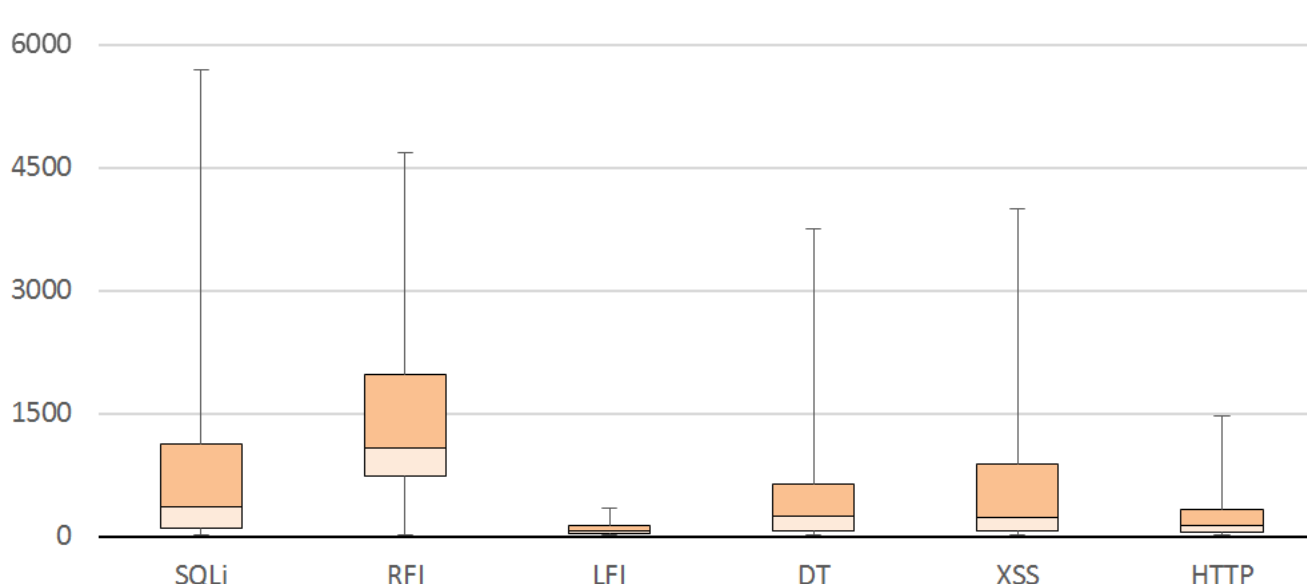


Figure 6 – Requests per Battle Day

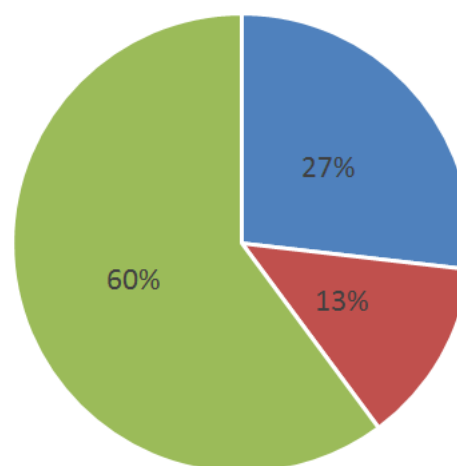
While SQL injection, Directory Traversal and HTTP attacks have relatively more battle days (Section 4.1.5), the attack types that have higher traffic volume are RFI, SQL Injection and XSS (Figure 6).

#### 4.1.7 - Reputation Based Traffic

Reputation based mitigation is based on classifying events according to pre-learned features, e.g., the source IP public reputation. We present two reputation based mitigations: Anonymous Proxies and TOR IPs, both rely on lists of IPs. We make the assumption that users who wish to stay anonymous probably have something to hide; are more likely to produce malicious traffic; or at the very least, have lower value to the web application.

Figure 7 shows the distribution of alerts between Anonymous Proxies, TOR IPs, and the attack types we analyzed so far in the report: SQLi, RFI, LFI, Directory Traversal, XSS, HTTP and Comment Spam.

Reputation based alerts comprise 40% of the alerts we examined (Figure 7) and proves the significant value this mitigation provides.



■ Anonymous Proxies ■ TOR IPs ■ All Attack Types

Figure 7 – Alerts Distribution, Attacks versus Reputation

## 4.2 - Last Year Report Comparison

This section compares the data from WAAR #4 and the data for the effective period, with focus on the number of incidents and battle days. In order to provide more accurate analysis in this report, we changed the analysis of incidents and battle days presented in sections 4.1.2 and 4.1.5, to consider all web applications protected by Imperva's WAFs, including those that weren't exposed to attacks (zero value instances). For the sake of comparison, the analysis presented in sections 4.2.2 and 4.2.3 below, follows WAAR #4 and counts only non-zero instances.

### 4.2.1 - Web Application Participation per Attack Type

Table 9 shows the comparison of the ratio of web applications that suffered from each attack type, between the periods of WAAR #4 and the current report.

For most of the attack types, significantly more web applications were exposed to malicious traffic in the effective period.

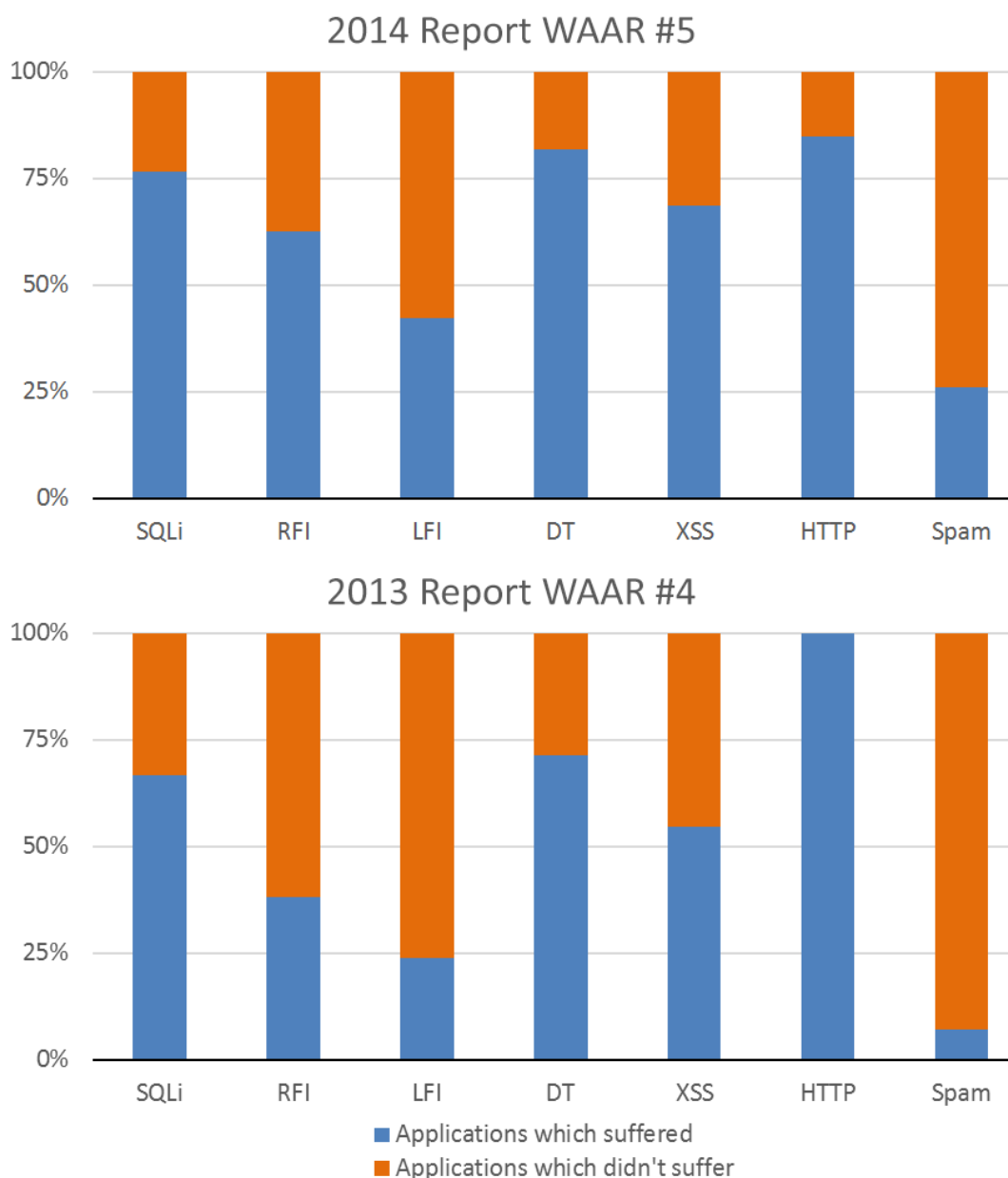


Table 9 – Web Application Malicious Traffic Exposure Ratio 2013-2014

Table 8 shows that in 2014 all attack types, except for HTTP, there was an increase in the percent of web application exposure to malicious traffic, compared to WAAR #4. The only attack type that shows a decrease is HTTP, which can be explained with more accurate classification of HTTP attacks.

#### 4.2.2 - Number of Incidents

Figure 8 shows the comparison of the number of incidents between the periods of WAAR #4 and this report, for each attack type.

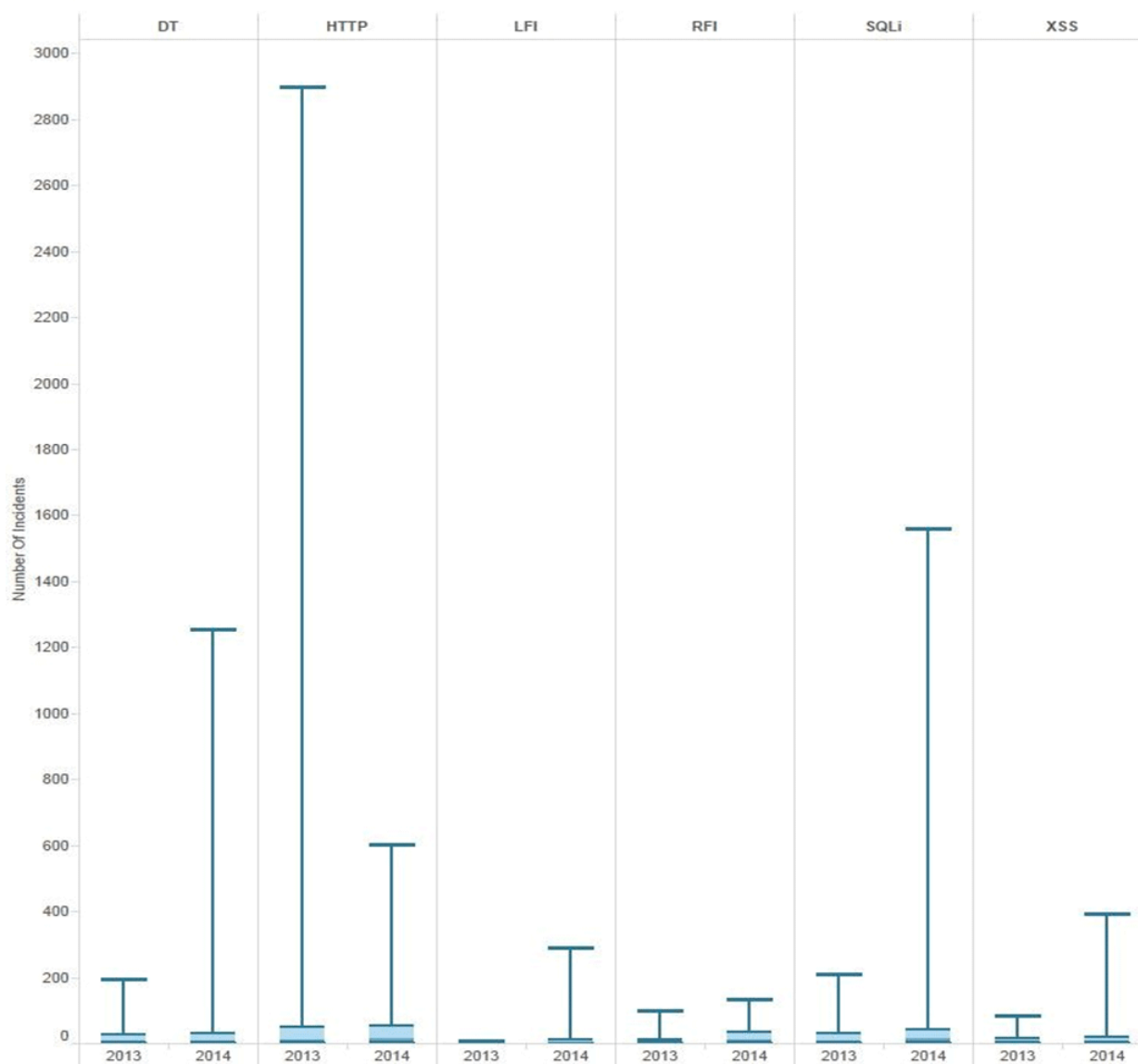
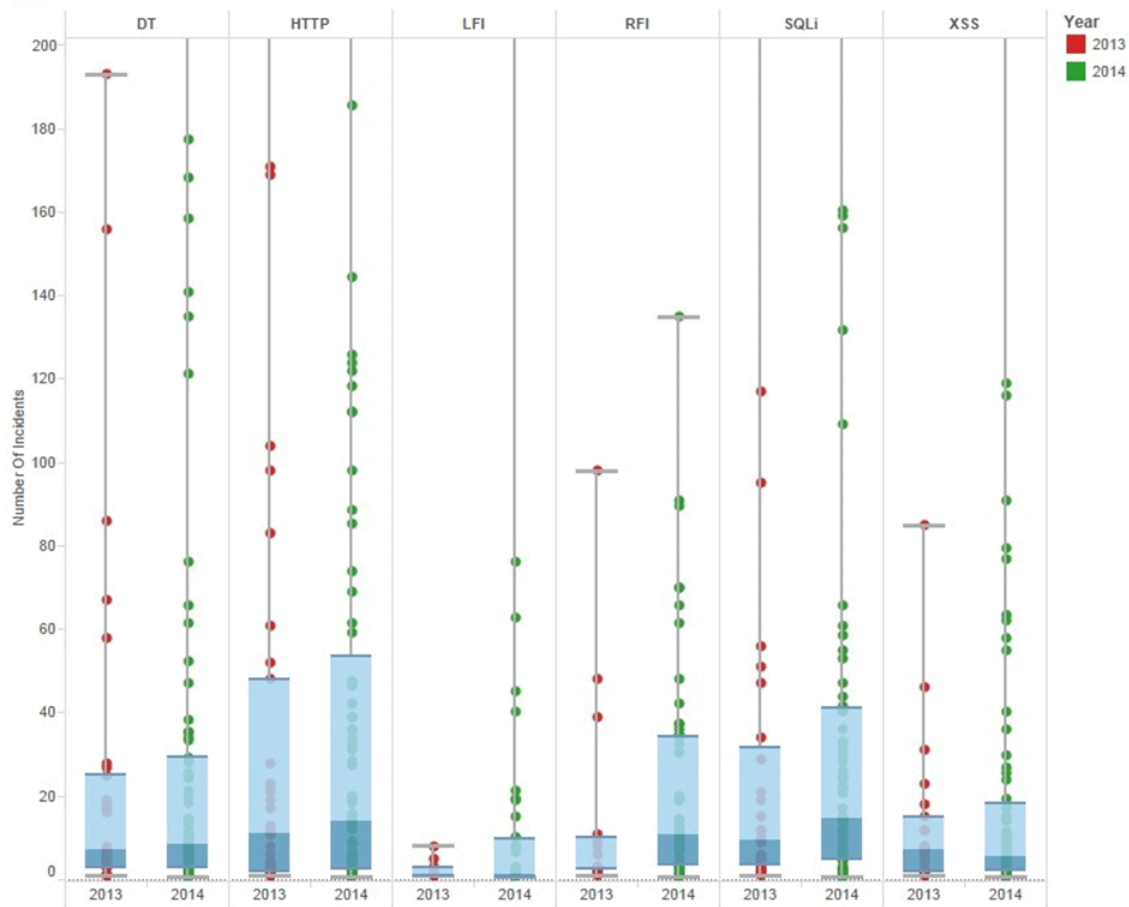


Figure 8 – Comparison of Number of Incidents between WAAR #4 and WAAR #5

The trend that was shown in section 4.2.1 is demonstrated also in Figure 8, with an increase in the maximum values for all attack types, except for HTTP attacks.

In order to observe the quartiles of these trends, in Figure 9 we zoom into this diagram. The green and red dots represent particular web applications that suffered attack incidents in the previous and effective periods and give more accurate visualization of the distribution of the number of attack incidents.



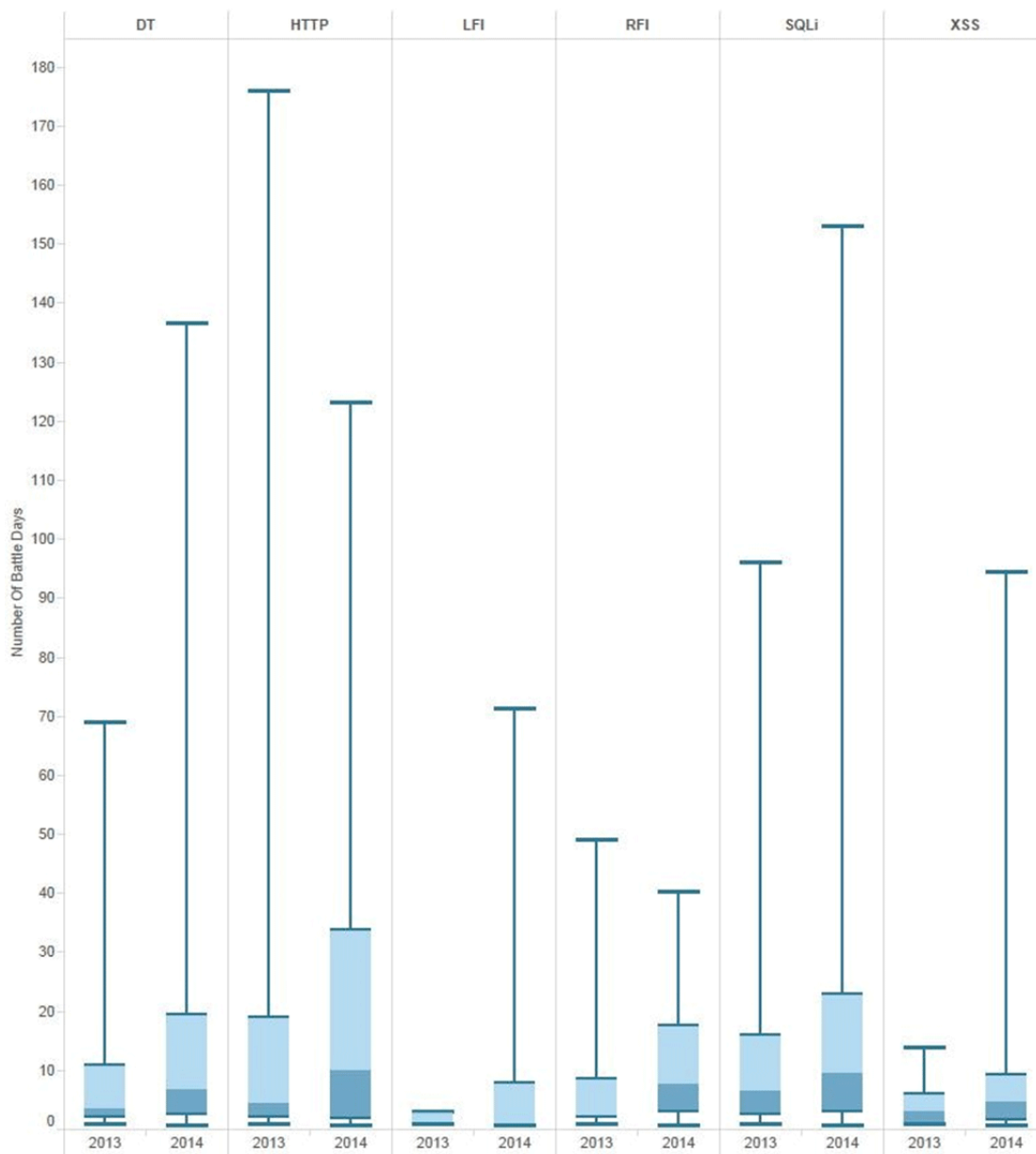
**Figure 9 – Comparison Zoom of Number of Incidents between WAAR #4 and WAAR #5**

The main conclusion that stems from Figure 9 is that the ones that suffer, suffer more. Web applications that suffered attack incidents in the effective period, had to cope with more attack incidents than those in previous period.. Figure 9 shows a significant increase in incidents for both the median and the 3rd quartile for all attack types between WAAR #4 and WAAR #5. RFI shows the most significant growth, which becomes even more significant when considering the increase in the number of web applications that suffer from RFI (increasing from 38% to 62%).



### 4.2.3 - Number of Battle Days

Figure 10 shows the battle day comparison between the periods of WAAR #4 and this report.



*Figure 10 – Battle Days 2013-2014 Comparison*

In order to observe the quartiles of these trends, in Figure 11 we zoom into this diagram, with the green and red dots representing particular web applications that suffered attack incidents in the previous period and the effective period.

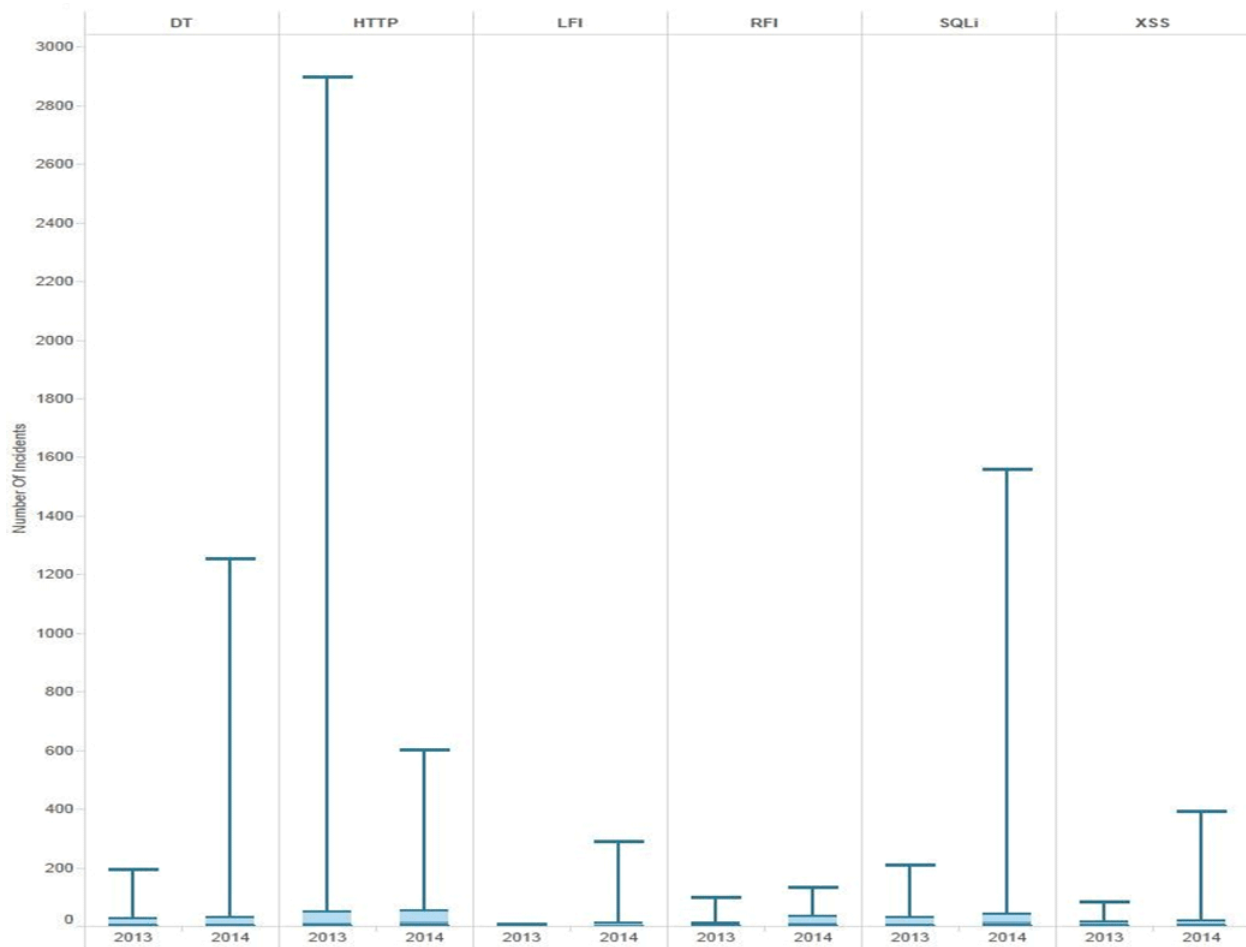


Figure 11 – Battle Day 2013-2014 Comparison Zoom

Once again, the ones that suffer, suffer more. All attack types show an increase in the median and 3rd quartile values (the same conclusion as was presented in section 4.2.2). It means that web applications that suffered attacks in the effective period, had more battle days than web applications that suffered attacks in the previous period.

For example, one quarter of the web applications that suffered SQL Injection attacks in the current period had to cope with 23 battle days – 44% more than the 16 battle days in the previous period. Similar trends are demonstrated for other attack types, in particular RFI and HTTP attacks, which show the most significant increase.

The increase in the number of HTTP battle days stems from the fact that in the effective period, the HTTP attack analysis focuses on fewer web applications, due to more accurate detection.

## 4.3 - Data Slicing

### 4.3.1 - Comparing Vertical Industries

We analyzed attack patterns within different vertical industries. We classified the web applications into the following categories:

- Retail – online stores, such as mobile equipment, clothing, jewelries, and vehicles
- Commercial information – websites presenting different organizations' commercial content such as business-to-business and business-to-consumer marketing sites.
- Finance – financial institutions and companies, such as banks and investment funds
- Online services – different SaaS, information sharing, mail services, and payments
- Other – web applications which don't match any other category

Figure 12 illustrates the median of the distribution of incidents number per category, for each attack type.

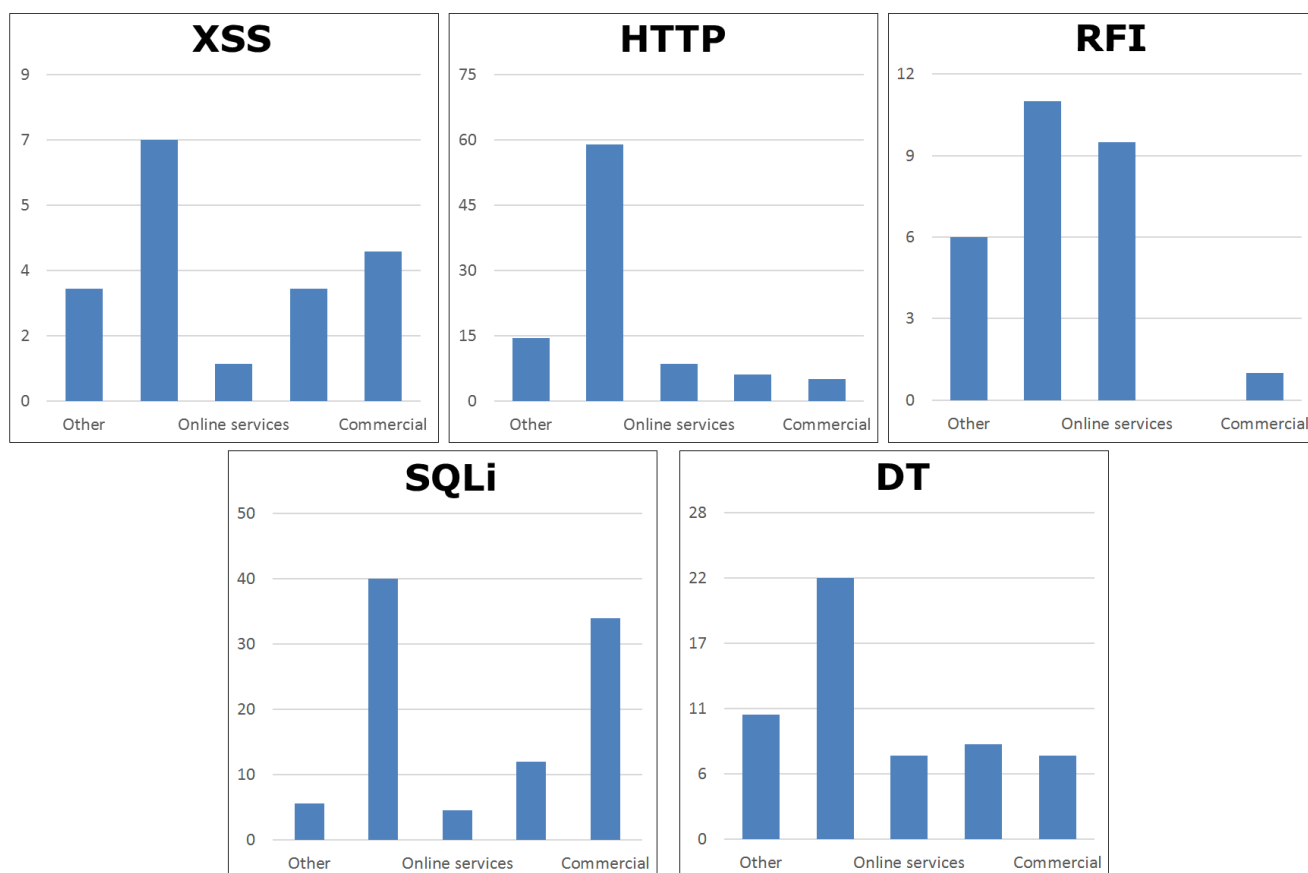


Figure 12 – Median Number of Incidents by Category for Different Attack Types

Retail applications suffer significantly more attack incidents for SQL Injection, XSS, Directory Traversal, HTTP, and RFI attacks. It is deduced by the higher median value for retail applications for each attack type presented in Figure 12.

Commercial Information applications suffer more SQL injection and XSS attack incidents compared to finance, online services, and other web applications.

### 4.3.2 - Comparing Frameworks

99.5% of the web applications use PHP or ASP Web Application Frameworks, as shown in Figure 13 (3).

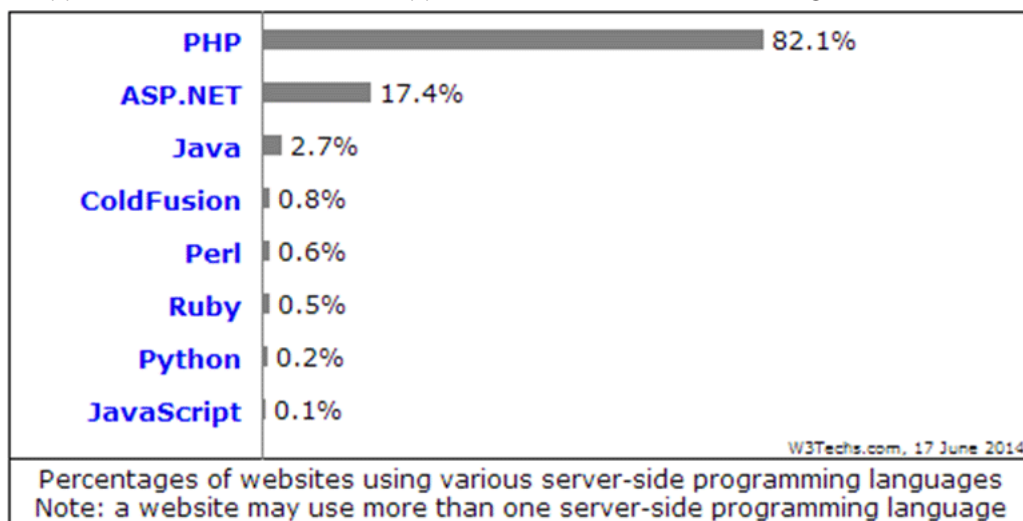


Figure 13 – Usage of Web application Frameworks

We classified 64 web applications out of the 99 web applications we examined: 33 (52%) are ASP and 26 (41%) are PHP.

Figure 14 shows the relative portion of each attack type for ASP versus PHP.

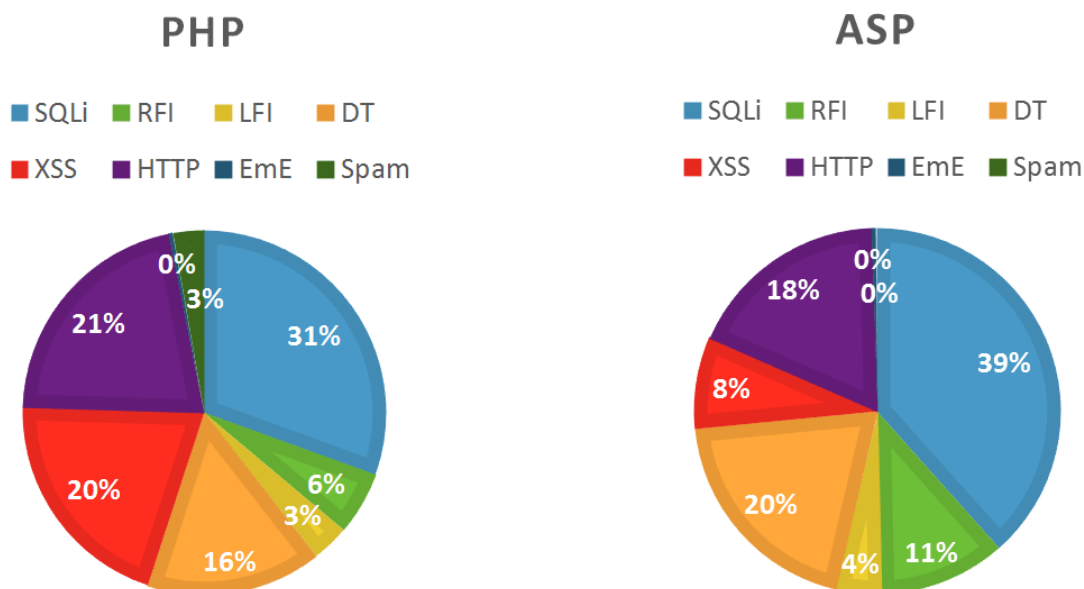


Figure 14 – Relative Portion of Each Attack Type, ASP vs. PHP

The portion of XSS attacks is significantly larger for PHP applications, almost three times the portion in ASP applications. On the other hand, the portion of RFI attacks in ASP applications is almost two times the portion for PHP applications.

In Figure 15 we compare PHP and ASP applications in their median for the number of incidents; the average magnitude per incident; and the average duration per incident for PHP versus ASP.

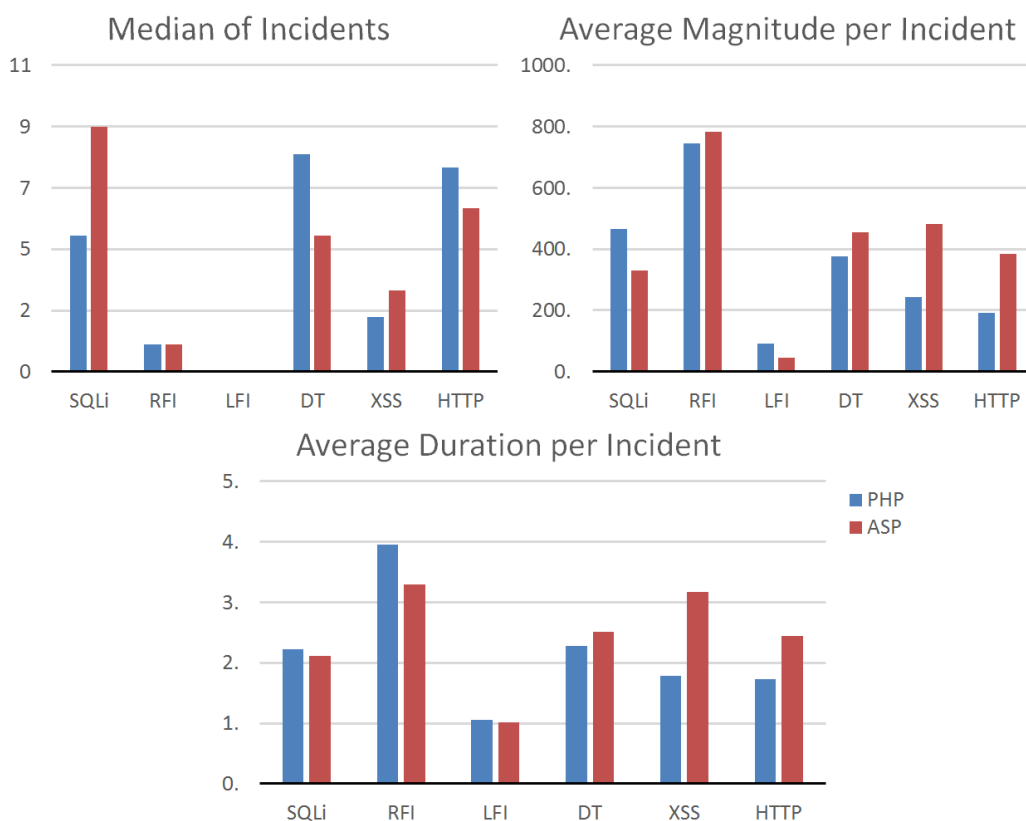


Figure 15 – Framework Attack Type Incidents Breakdown

Figure 15 shows the following:

- ASP applications suffered almost twice SQL injection attack incidents compared to PHP applications. However, these attacks are characterized with lower magnitude.
- PHP applications suffered almost twice Directory Traversal attack incidents compared to ASP applications.
- XSS attacks and HTTP attacks on ASP applications are significantly longer and have significantly higher volume, than those on PHP applications.

### 4.3.3 - Comparing Content Management Systems

WordPress is the most popular Content Management System (CMS) used today (4). From the population we examined, we were able to classify the CMS of 33 web applications, out of which 11 use WordPress, and 22 use other CMS applications. Figure 16 shows the distribution of attacks relative portions of the different attack type incidents for WordPress applications versus non-WordPress applications. We found out that when comparing the magnitude of XSS attacks on WordPress to those of other Content Management Systems, WordPress has 60% more XSS incidents.

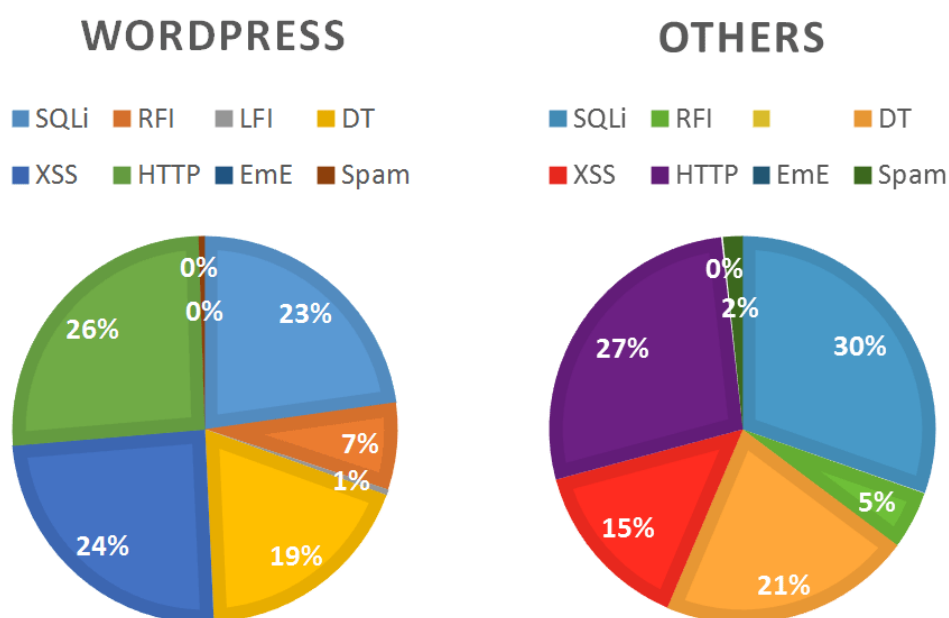
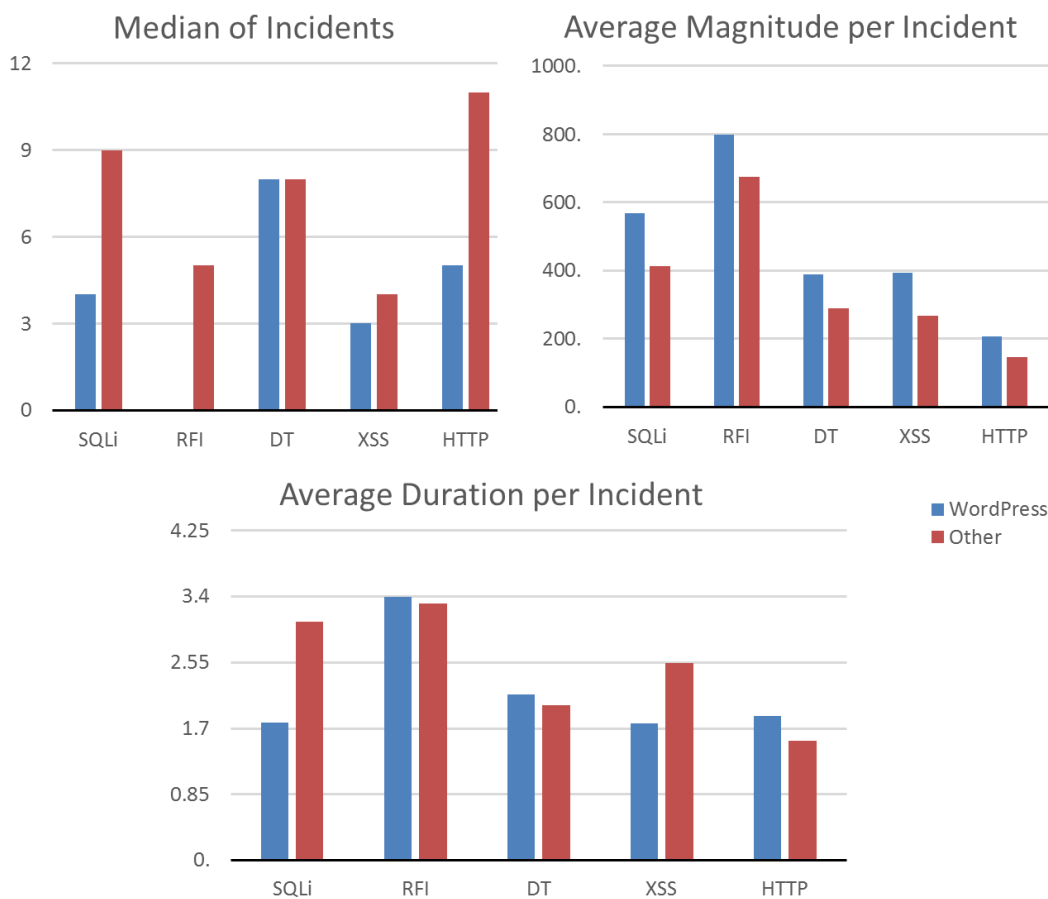


Figure 16 – Relative Portions of Each Attack Type in WordPress vs. Other CMS



Figure 17 shows the incidents median for each attack type; the magnitude per incident; and the duration per incident.



**Figure 17 – CMS Attack Type Breakdown**

The following facts arise from Figure 17:

- In average, WordPress attack incidents have higher attack incident magnitude for all attack types, compared to other CMS applications (Average Magnitude per Incident graph).
- WordPress is more likely to suffer less (or equal) number of incidents for each attack type (Median of Incidents graph).
- WordPress suffers a higher traffic volume for each attack type (Magnitude per Incident graph)

Notice that although there is smaller number of attack incidents for WordPress, the volume of each attack incident is higher. This situation recurs for all attack types.

## 4.4 - Geographic Dispersion

We analyzed the geographic distribution of the attack sources for different attack types. We examined both the number of requests (section 4.4.1) and the number of attacking hosts (section 4.4.2). In both cases, we examined the top quartile and normalized the data using an estimation of the per-country number of internet users, as published by the WorldBank (5).

### 4.4.1 - Traffic Volume

Table 10 shows the top 10 countries with largest amounts of malicious traffic, per attack type. We defined the ‘malicious traffic’ of a country as the number of malicious HTTP requests per internet users (5) – the higher the ratio, the more malicious is the country’s traffic. For each country we present the absolute number of HTTP requests. The most dominant countries across different attack types are marked in different colors.

DT		XSS		LFI	
Country Name	#Requests	Country Name	#Requests	Country Name	#Requests
COSTA RICA	39,965	CYPRUS	15,860	CYPRUS	1,905
CYPRUS	10,563	UNITED STATES	4,751,155	CZECH REPUBLIC	11,734
IRELAND	71,414	ICELAND	3,435	ISRAEL	5,237
UNITED STATES	2,886,810	UNITED KINGDOM	579,419	FRANCE	37,838
CZECH REPUBLIC	61,344	COSTA RICA	6,081	GERMANY	42,701
LATVIA	11,828	CANADA	63,644	NETHERLANDS	8,461
FRANCE	345,346	RUSSIA	129,962	UNITED STATES	67,377
KUWAIT	12,265	KUWAIT	5,042	IRAN	5,271
LITHUANIA	15,586	SWEDEN	14,072	SINGAPORE	828
UNITED KINGDOM	254,007	MOLDOVA	2,440	LUXEMBOURG	11,579
HTTP		RFI		Comment Spam	
Country Name	#Requests	Country Name	#Requests	Country Name	#Requests
SWITZERLAND	336,666	BULGARIA	41,366	UNITED STATES	324,806
FINLAND	176,294	BELARUS	43,371	NORWAY	1,751
UNITED STATES	6,809,542	NETHERLANDS	143,175	NETHERLANDS	4,792
GERMANY	2,206,500	UNITED STATES	1,962,459	HONG KONG	1,427
CYPRUS	13,343	ROMANIA	167,905	CANADA	6,874
INDIA	2,218,717	CANADA	195,786	PANAMA	693
NETHERLANDS	282,266	SINGAPORE	22,796	BELARUS	839
ESTONIA	13,198	SLOVAKIA	30,342	TURKEY	14,436
LATVIA	24,099	UNITED KINGDOM	300,765	UKRAINE	7,583
CROATIA	25,861	HONG KONG	216,15	RUSSIA	8,679
SQLi					
Country Name	#Requests				
CYPRUS	34,593				
KUWAIT	95,435				
COSTA RICA	48,065				
UNITED STATES	4,575,883				
NETHERLANDS	229,831				
CANADA	365,866				
MOLDOVA	19,292				
LATVIA	21,244				
SWEDEN	66,774				
BELARUS	30,428				

Table 10 – Top 10 Countries with Malicious Traffic per Attack Type

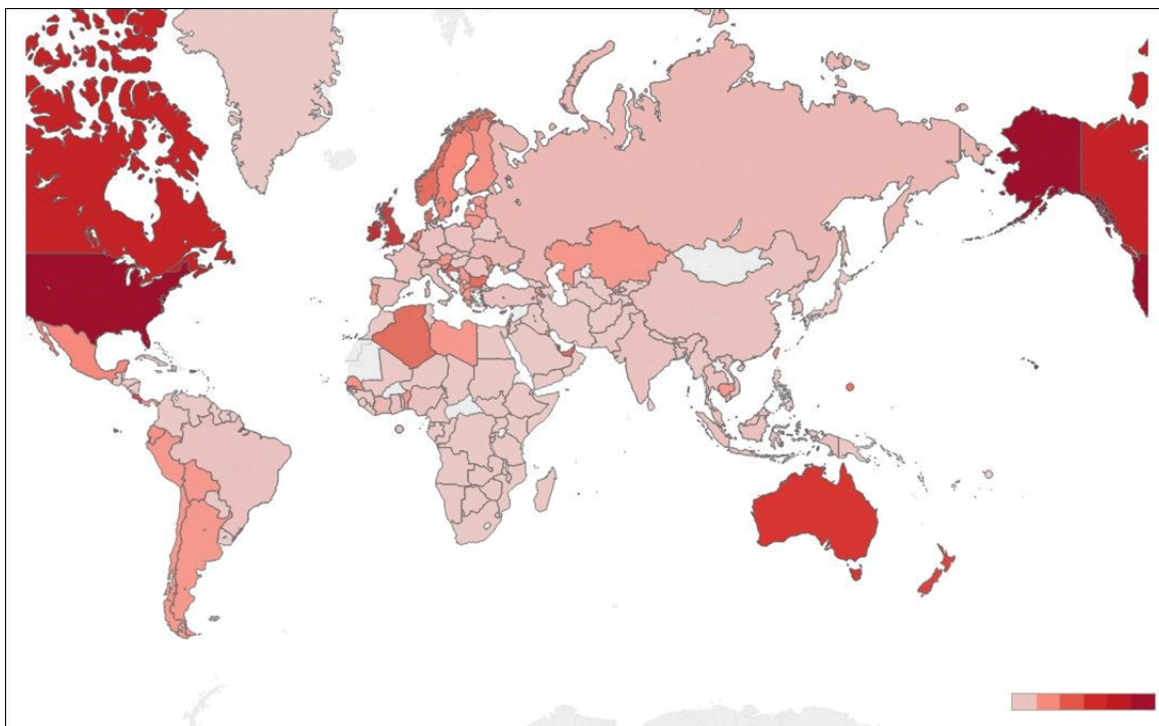
The top four dominant countries across all attack types are: United States, Netherlands, Cyprus, and Canada (Table 10).

Figure 18 shows the world map of malicious countries for all attack types.

Figure 18 shows the world map of attacking IPs per country for the different attack types: Directory Traversal, HTTP, LFI, RFI, Comment Spam, SQL injection, and XSS attacks.

#### Report Time Period

The country's position in the ranking of Traffic Volume and Attacking Hosts (Section 3.2) is not correlated with the absolute number of malicious HTTP requests arriving from this country. This is due to the normalization to the number of internet users that cause small countries sometimes appear above larger countries with larger numbers.



**Figure 18 – World Map Distribution of Malicious Countries for All Attack Types**

Figure 18 shows that even when normalizing their size and amount of web users, the more industrial countries: United States, Canada, United Kingdom and Australia, are responsible for the major part of the malicious traffic.

#### 4.4.2 - Attacking Hosts

Table 11 shows the top 10 countries with malicious attacking hosts. We ranked the countries according to the number of attacking hosts per internet users (5). The higher the ratio, the higher the ranking, which means that the country has more malicious hosts. For each country we present the absolute number of attackers. The most dominant countries across different attack types are marked in different colors.

Comment Spam		RFI		LFI	
Country Name	#Attackers	Country Name	#Attackers	Country Name	#Attackers
SEYCHELLES	55	SWEDEN	385	SINGAPORE	16
UNITED STATES	19,635	BELARUS	136	UNITED STATES	837
CANADA	839	UNITED STATES	6,799	GERMANY	204
SWEDEN	266	CANADA	566	NETHERLANDS	40
CHINA	12,017	FRANCE	1,111	SWITZERLAND	17
UNITED ARAB EMIRATES	158	SOUTH KOREA	517	FRANCE	101
PANAMA	62	NETHERLANDS	183	CANADA	36
SINGAPORE	63	CZECH REPUBLIC	72	UNITED KINGDOM	62
UNITED KINGDOM	952	ROMANIA	206	RUSSIA	42
HONG KONG	66	AUSTRIA	61	SOUTH KOREA	28
HTTP		SQLi		DT	
Country Name	#Attackers	Country Name	#Attackers	Country Name	#Attackers
UNITED STATES	124,793	BELARUS	1,259	SEYCHELLES	29
UNITED KINGDOM	25,957	UNITED KINGDOM	10,830	BELARUS	2,749
COSTA RICA	561	UNITED STATES	32,160	QATAR	167
TAIWAN	5,433	ALGERIA	636	UKRAINE	6,925
CANADA	6,008	PUERTO RICO	372	KAZAKHSTAN	1,123
UNITED ARAB EMIRATES	1,228	CANADA	2,350	ICELAND	31
CROATIA	445	KAZAKHSTAN	625	UNITED STATES	20,482
MEXICO	6,047	SOUTH KOREA	2,662	UNITED ARAB EMIRATES	617
PANAMA	453	NETHERLANDS	1,072	CANADA	1,661
NETHERLANDS	2,163	SWEDEN	573	MALTA	15
XSS					
Country Name	#Attackers				
UNITED KINGDOM	301,397				
ICELAND	1,028				
UNITED STATES	199,620				
IRELAND	3,279				
CANADA	19,283				
AUSTRALIA	9,351				
NEW ZEALAND	1,546				
SINGAPORE	1,110				
BULGARIA	862				
SWEDEN	2,022				

Table 11 – Top 10 Countries with the Malicious Attacking Hosts

The list of top four dominant countries across all attack types is very similar to the one for attacking IPs, with United-States, Canada, Netherlands and United-Kingdom (Table 11) – the latter replacing Cyprus.

## 5 Conclusions and Recommendations

As noted in various sections of this report, attack volumes and density are growing period over period. As attackers become more sophisticated, it is important that companies educate themselves on the threats that they are facing, and on the risk factor that is aligned with that threat.

To mitigate the attacks and threats mentioned in this report, organizations should consider several actions:

- Deploy security solutions that mitigate against automated attacks. To do so, security solutions should recognize known automated sources; differentiate between bots and human clients; as well as detect unusual activity, such as an extremely high rate of web requests from a single user. Automated attack detection must be identified as early as possible during an attack incident.
- Learn from peers. Applications in similar business verticals may share similar attack characteristics. In this report, we have shown that online retail applications suffer more than other verticals. Applications with private consumer information seem to suffer more, and there are certain attack vectors such as RFI that are growing fast.
- Detect and block attacks that target known vulnerabilities. The knowledge base of exploitable weaknesses in an application must be frequently updated. The difference in attacks on infrastructure frameworks such as PHP and ASP point to the frequent need to patch, in order to avoid a breach induces by a third party software component.
- Acquire intelligence on malicious sources and apply this intelligence in real time. Black lists of attack sources are still an efficient counter-measure. However, lists must be up-to-date, in order to be effective.
- Participate in a security community and share threat intelligence. The increased automation and scale of attacks leave a large footprint which can only be seen by analyzing data gathered from a large set of potential victims.
- Attack distribution is burst-oriented and far from consistently distributed. Estimations for security measures should be based on the worst case scenario, not on the average case.
- Security procedures and solutions should be as automated as possible, since attack volume is too overwhelming for humans to monitor, and typically, there will be no advance warning of an attack.

## 6

## Attack Glossary

### 6.1 - SQL Injection

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

### 6.2 - Remote File Inclusion

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 client side, such as JavaScript execution, which can lead to other attacks. This vulnerability occurs due to the use of user-supplied input without proper validation.

### 6.3 - Local File Inclusion

Local File Inclusion (LFI) is an attack that includes files on a server into the web server. This attack can lead to malicious code execution on the web server. The vulnerability occurs when an included page is not properly sanitized and allows, for example, directory traversal characters to be injected. LFI attacks often append a null character to the included file path in order to bypass value sanitization.

### 6.4 - Directory Traversal

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. Characters representing “traverse to parent directory” are passed through to the file APIs.

### 6.5 - Cross-Site Scripting

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.

### 6.6 - Comment Spamming

Comment spamming is a way to manipulate the ranking of the spammer's web site within search results returned by popular search engines. A high ranking increases the number of potential visitors and paying customers of the site. The attack targets web applications that allow visitors to submit content that contains hyperlinks. The attacker automatically posts random comments or promotions of commercial services to publicly accessible online forums that contain links to the promoted site.

In 2014, Imperva released a research paper defining [Comment Spam](#).

# 7

## References

- (1) Retrieved from [http://en.wikipedia.org/wiki/Box\\_plot](http://en.wikipedia.org/wiki/Box_plot)
- (2) Retrieved from [http://www.imperva.com/docs/HII\\_Web\\_Application\\_Attack\\_Report\\_Ed4.pdf](http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed4.pdf)
- (3) Retrieved from [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)
- (4) Retrieved from [http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)
- (5) Retrieved from <http://data.worldbank.org/indicator/IT.NET.USER.P2>