

# **Imperva's Web Application Attack Report**

**Edition #3 - July 2012**

## Table of Contents

<b>1 Abstract</b>	<b>3</b>
<b>2 Executive Summary</b>	<b>4</b>
<b>3 Analysis Methodology</b>	<b>5</b>
3.1 Data corpus	5
3.2 Updates in data analysis and presentation methodology	5
3.3 Analysis glossary	6
3.4 Attack glossary	6
3.4.1 SQL Injection	6
3.4.2 Remote File Inclusion	6
3.4.3 Local File Inclusion	6
3.4.4 Directory Traversal	6
3.4.5 Cross Site Scripting	6
3.4.6 Email Extraction	7
3.4.7 Comment Spamming	7
3.4.8 HTTP Protocol Violations	7
<b>4 Analysis Results</b>	<b>8</b>
4.1 Overview	8
4.2 Trends over time	8
4.3 Comparative attack overview	12
4.3.1 Amount of attack incidents	12
4.3.2 Attack incident magnitude	13
4.3.3 Attack incident duration	14
4.3.4 Battle days	15
4.4 Single application case study- full year	15
4.5 Worst-case scenarios	16
4.6 Battle days predictions	16
4.7 Geographic dispersion	17
<b>5 Conclusions and Recommendations</b>	<b>19</b>

This document contains proprietary and confidential material of Imperva. Any unauthorized reproduction, use or disclosure of this material, or any part thereof, is strictly prohibited. This document is solely for the use by Imperva employees and authorized Imperva customers.

## ① Abstract

In our previous Web Application Attack Reports (WAAR), we described the intensity of application attacks where websites are probed about once every two minutes, or 27 times per hour. This analysis gave a snapshot of an average application under attack. In this report, we identify how many attacks a typical application can expect annually as well as the duration. Specifically, we take a deeper look to expose the underlying distribution and gain a more comprehensive understanding of the cyber battlefield. We found that the typical application:

- › Can expect attack incidents 120 days per year or 33% of the time with some targets experiencing attacks 292 days per year or nearly 80% of the time.
- › Will be attacked 274 times per year with a maximum number of attacks hitting 2,766.
- › When under attack, the typical individual duration of the campaign is 7 minutes and 42 seconds with the longest attack reaching 79 minutes.
- › SQL injection is the most frequently used attack.

## ② Executive Summary

The 2011 Verizon Data Breach Report tells us that, in the case of a compromise, the web application provided an attack vector 80 percent of the time. What is behind this high percentage? From a hacker's perspective, the application is a gateway to the valuable data the application transacts. Technically, the main driver is the automation of attack tools. In other reports, we described some of these tools and quantified their efficacy. What does the effect of these tools mean, holistically?

We had found out that from an application defense standpoint, the cyber battlefield looks more like a border-keeping mission than total war. Most of the time very little happens, but every once in a while there's an outbreak of attacks.

We found that the average application:

- › Can expect attack incidents 120 days per year or 33% of the time, with some targets experiencing attacks 292 days per year or nearly 80% of the time.
- › Will be attacked 274 times per year, with a maximum number of attacks hitting 2,766.
- › When under attack, the typical individual duration of the campaign is 7 minutes and 42 seconds with the longest attack reaching 79 minutes.

However, regardless of attack frequency periods, compared to the peaceful periods, the success of the whole mission depends on the defense performance when under attack. Therefore, the defense solutions and procedures should be designed to accommodate attack bursts.

While, typically, an application will see only some serious attack action on 59 days in 6 months (roughly on every third day on average), and the attack period may last only a few minutes. The intensity of the attack will be overwhelming if the defense side was prepared for the average case (27 or 18 attacks per hour as discovered on our previous reports) as the attack will consist of hundreds or even thousands of individual attack requests.

Another key finding is that the correlation between different days seems to be low. It suggests that prediction of future attacks days, based on the attacks history, would be challenging, at best. Therefore, the defense side cannot count on having any advance notice.

Lastly, the geographic attack trends reported in the previous WAAR remain relevant, with the majority of requests and attackers originating in the USA, western European countries, China, and Brazil. For business logic attack types, email extraction is still widely dominated by western African countries, like Senegal, Nigeria, Ghana, and the Ivory Coast. The field of comment spamming shows unusual bias towards former eastern-bloc countries, such as Russia, Ukraine, Latvia, and Poland.

A consistent finding throughout our reports; more often than not, the largest amount of attack requests originate from IPs allocated to the USA. Therefore, it was very surprising to find out that France has earned the dubious title of being the leader in SQLi attacks – as attack volume of requests originated from France was almost four times greater than of the USA.

## ③ Analysis Methodology

### 3.1 Data corpus

This security summary report is based on observing and analyzing Internet traffic to 50 web applications during the past 6 months (December 2011 – May 2012). We extracted from the traffic security attacks on these applications, categorized them according to the attack method, and identified patterns and trends within these attacks. In order to protect the anonymity of the applications while maintaining their identity for analysis purposes, the applications were arbitrarily labeled with numerals.

Monitoring of the web applications deployed at these sites over a period of several months was accomplished using automatic tools. The logged security events were analyzed using Imperva's special-purpose software and knowledge base. This analysis used several measures, including matching to known attack signatures, comparison to black lists of malicious hosts, and calculation of statistical properties of malicious traffic. Imperva's security experts performed additional detailed analysis of the important events and patterns.

### 3.2 Updates in data analysis and presentation methodology

*Then there is the man who drowned crossing a stream with an average depth of six inches - W.I.E. Gates*

We would like to emphasize the changes made in our analysis and presentation methodology relative to our previous Web Attack Analysis Report.<sup>1</sup>

The focus of our previous reports was fine grained and concentrated on analyzing the measurements of malicious HTTP requests. In this report we take a higher level view and discuss the broader notion of attack incident. An attack incident may include hundreds or thousands of consecutive HTTP requests targeting the same application. The individual requests are all part of the same attempt to compromise the application. Furthermore, these related requests are usually sent at a high rate. A major reason for this nature of attacks is that in the wild, attacks are usually automated as shown in our previous HII publications.<sup>2</sup> In practice, the number of individual requests is usually less meaningful for the security manager, as the effort involved with mitigating the whole attack incident, or a persisting campaign, is mostly unrelated to the total number of requests it contains.

For the purpose of this report, we defined an attack incident to consist of at least 30 requests per 5 minutes (one attack request every 10 seconds on average). Of course, a single attack may span several consecutive 5-minute periods. We also defined an even broader concept of "battle days": These are days in which at least a single attack incident targeted the application.

Attacks against a web application can be measured according to several criteria, which have very practical security implications:

- › How many attack campaigns occur in a given period of time?
- › How long does each attack last?
- › How massive is each attack campaign (i.e., how many HTTP requests are issued as part of the attack)?
- › How many "battle days" should I expect over the long run?
- › If I had a "battle day" yesterday, what can I tell about tomorrow?

We base our answers to these important questions on numbers derived from the actual measurements (as described in section 3.1) using standard statistical tools. However, statistics can often be misleading. Benefiting from the experience we had gathered from preparing WAAR I and II, 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 plans the specifications and procedures for web application defense based on the average number of attacks (27 attacks per hour on last year's report) is likely to be overwhelmed when an attack stream of ten requests per second hits the application. Therefore, in this report, we had used relevant descriptive statistics like the median and quartiles rather than averages and standard deviation. Graphically, we present the resulting numbers using box-and-whisker plots.

<sup>1</sup> [http://www.imperva.com/docs/HII\\_Web\\_Application\\_Attack\\_Report\\_Ed2.pdf](http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed2.pdf)

<sup>2</sup> [http://www.imperva.com/docs/HII\\_Automation\\_of\\_Attacks.pdf](http://www.imperva.com/docs/HII_Automation_of_Attacks.pdf)

Based on the data we have collected over the last 6 months, we have reached answers for the questions above. These answers we can help by taking some real-world action items, such as the planning of the addition of ad-hoc help to beef up the organization's security.

### 3.3 Analysis glossary

- › **Attack request** – a single HTTP request identified as malicious.
- › **Attack incident** – as previously noted, attacks are burst-like in nature. Each burst that exceeded the rate of 30 attack requests per 5 minutes was defined an attack incident.
- › **Attack incident magnitude** – the number of attack requests consisting each 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.

### 3.4 Attack Glossary

#### 3.4.1 SQL Injection

*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.

#### 3.4.2 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'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.

#### 3.4.3 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 a page include 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 to bypass value sanitization.

#### 3.4.4 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, so that characters representing "traverse to parent directory" are passed through to the file APIs.

#### 3.4.5 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.

A box-and-whisker plot is a graphical tool used to depict the distribution of a dataset.

- › A box represents the range of values from first to third quartiles, that is, the interquartile range. The median divides the box into two ranges, showing the border between the second and third quartiles.
- › The lower whisker spans the range from the 5-percentile of the measured values to the first quartile, and the upper whisker spans the range from the third quartile up to the 95-percentile of the measured values.

This kind of plot displays differences between sets of measurements without making any assumptions of the underlying statistical distribution. The different parts of the box helps indicate the degree of dispersion (spread) and skewness in the data.

### 3.4.6 Email Extraction

*Email Extraction* (also called email scraping) is the practice of scanning web applications and extracting the Email addresses and other personal contact information that appears in it. These emails are then used for promotional campaigns and similar marketing purposes. Email extraction is one of several activities that harvest data from web applications against the intent of the data owners and the applications' administrators.

### 3.4.7 Comment Spamming

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

### 3.4.8 HTTP Protocol Violations

*HTTP Protocol Violation* is the case of sending requests that doesn't conform with the HTTP Protocol Specification. These violations consist of invalid HTTP methods, inclusion of invalid bytes inside parameters, etc. While not being an attack per-se, we have often observed violations of the HTTP protocol in the traffic that accompanies attack attempts. This kind of traffic is generated by custom scripts rather than standard web browsers. Consequently, protocol violations are an indicator of automatic attacks.

When summarizing measurement of an investigated phenomenon, like attacks against web application, we are often interested in succinctly describing:

- › What are the **typically** measured values? For web attacks, this is what the application usually faces.
- › How **diverse** are the measured values around this typical value? Since no single number can replace the full set of measurements, a measure of diversity indicates our confidence in using the typical value as a representative of the phenomenon and making decisions based on it.
- › What are the **extreme** values that were measured? Statisticians call these values "outliers." For web attacks, this is the volume of traffic observed during a highly concentrated automated attack. In practice, such attacks do occur, but are relatively rare.
- › How **asymmetric** (skewed) is the distribution of values. For example, if we measure the frequency of web attacks vs. their magnitude, we observe a skewed (asymmetric) distribution with a long tail: low-magnitude attacks are observed frequently, while extremely high-magnitude attacks do occur, but very rarely.

Summarizing a sample of values that is skewed and includes outliers using its average is often misleading, since the infrequent large numbers dominate the frequent low numbers, raising the average value above what is typically observed. In such cases, the typical value is better represented by the median, which is the numerical value separating the higher half of a sample of values from the lower half. In other words, half of the values are strictly less than the median, and half of the values are strictly greater than the median.

A generalization of the median are **quartiles**: a set of values with three points that divide the data set into four equal groups, each representing a fourth of the set of values that were measured.

- › **first quartile** (designated Q1) = lower quartile = splits lowest 25% of data = 25th percentile
- › **second quartile** (designated Q2) = median = cuts data set in half = 50th percentile
- › **third quartile** (designated Q3) = upper quartile = splits highest 25% of data, or lowest 75% = 75th percentile

The interquartile range (IQR), is a measure of statistical dispersion (how diverse is the data), being equal to the difference between the upper and lower quartiles  $IQR = Q_3 - Q_1$ .

## ④ Analysis Results

### 4.1 Overview

Our key stats on Web application attacks are summarized in the following table

Table 1 - Key stats summary

	Typical-case analysis (median)	Worst-case analysis (max)
Application battle days [battles days per 6 months]	59	141
Attack incidents against an application [attack campaigns per 6 months]	137	1383
Attack incident magnitude [attack requests]	195	8790
Attack incident duration [minutes]	7.70	79

The stats are summarized for all tested applications and for all attack types.

A typical application experienced 59 battle days, that is, days in which at least one attack incident occurred (roughly on every third day). In comparison, the worst case was 141 battle days in 6 months, meaning one of the tested applications suffered attack incidents in almost 80% of the days within this time period. Another interesting finding is that while the typical attack incident lasts around 8 minutes, the worst-case incident was about 10 times longer. These numbers refer to all attack types taken together; a breakdown of stats per specific attack type is described in detail on the sections below.

### 4.2 Trends over time

Here we describe the amount of attack incidents per week for a sample of applications for each attack type. For clarity, in the graph, we included only the top five applications that had suffered the most attack incidents during the observed period.

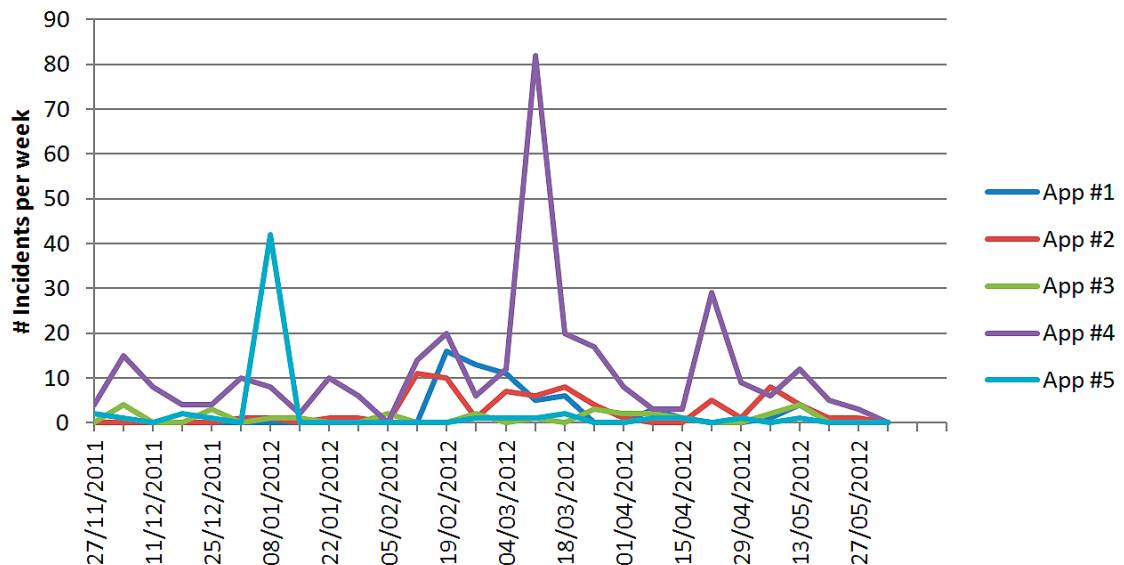


Figure 1 - SQLi attack incidents



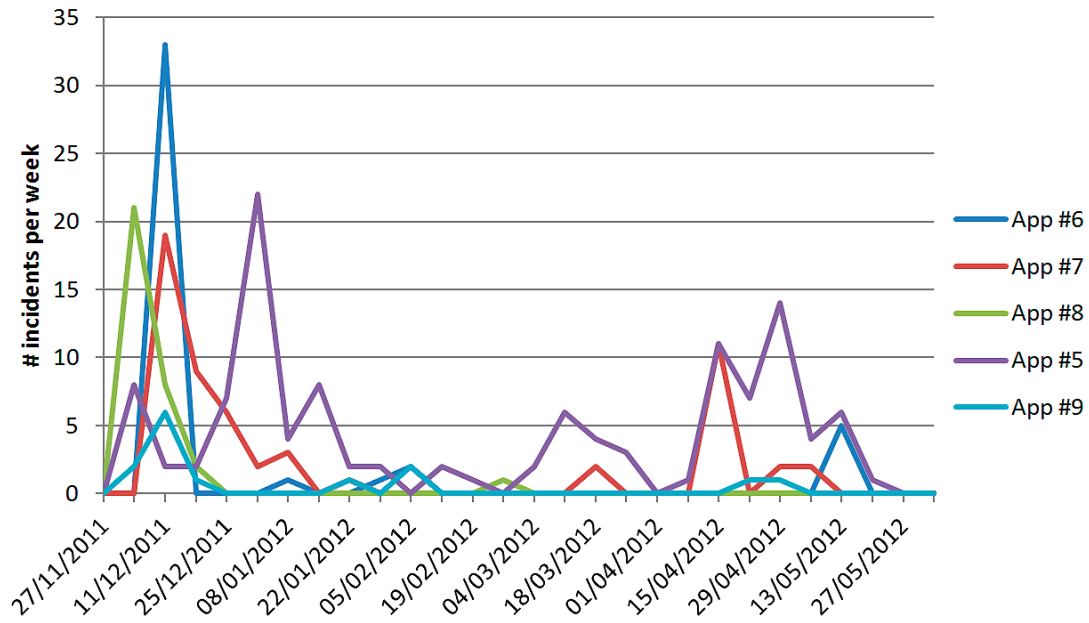


Figure 2 – RFI attack incidents

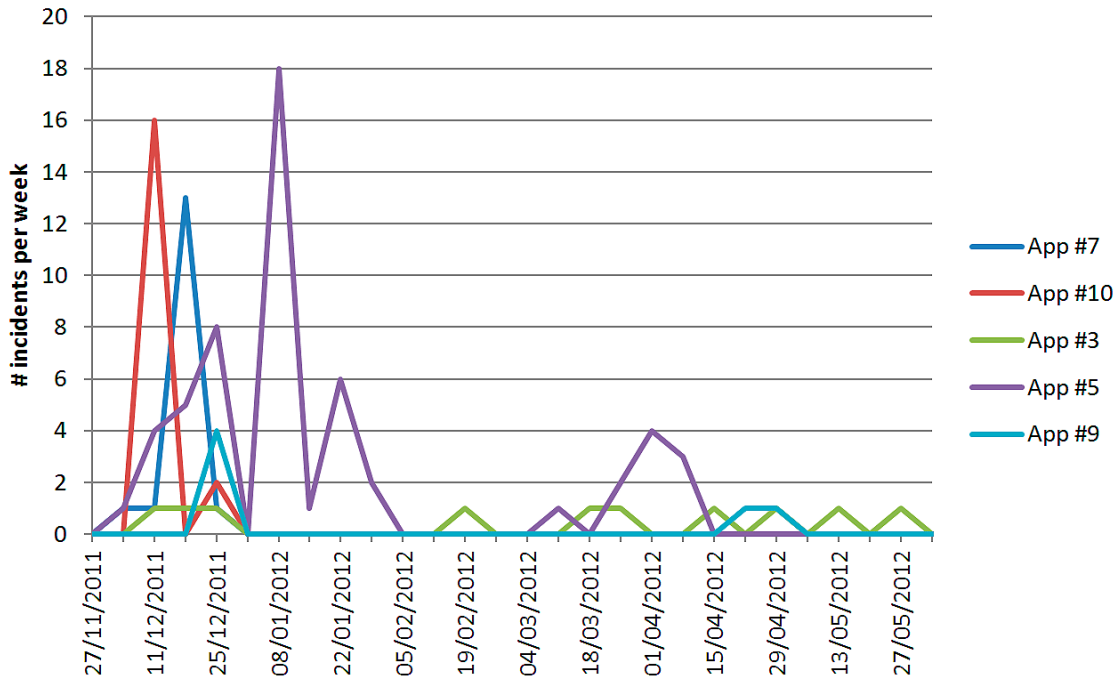


Figure 3 – LFI attack incidents

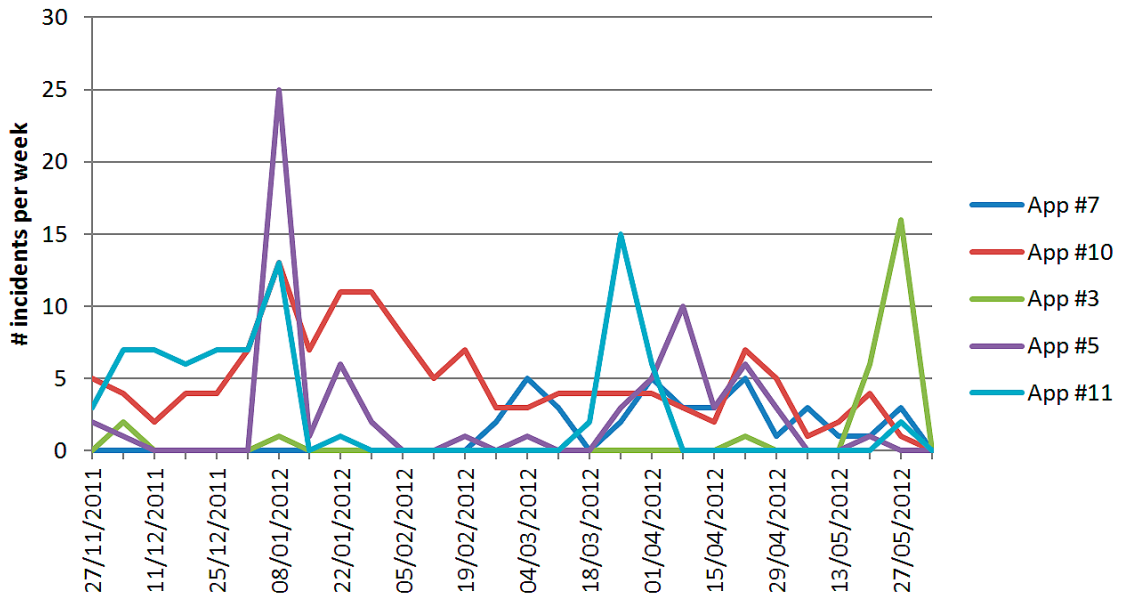


Figure 4 - Directory traversal attack incidents

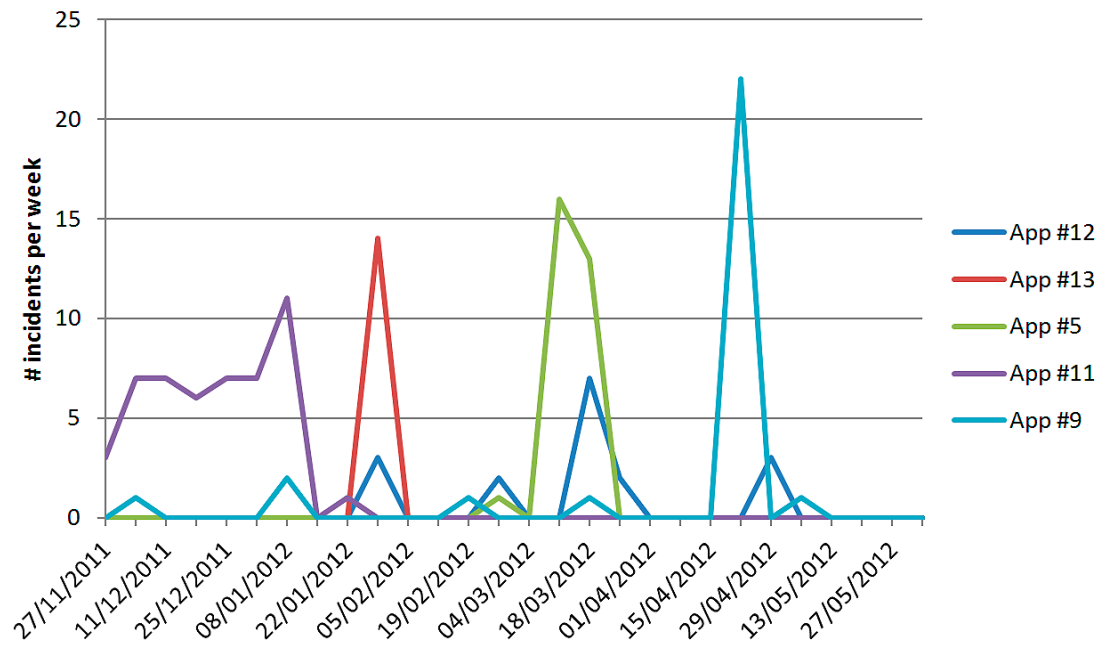


Figure 5 - XSS attack incidents

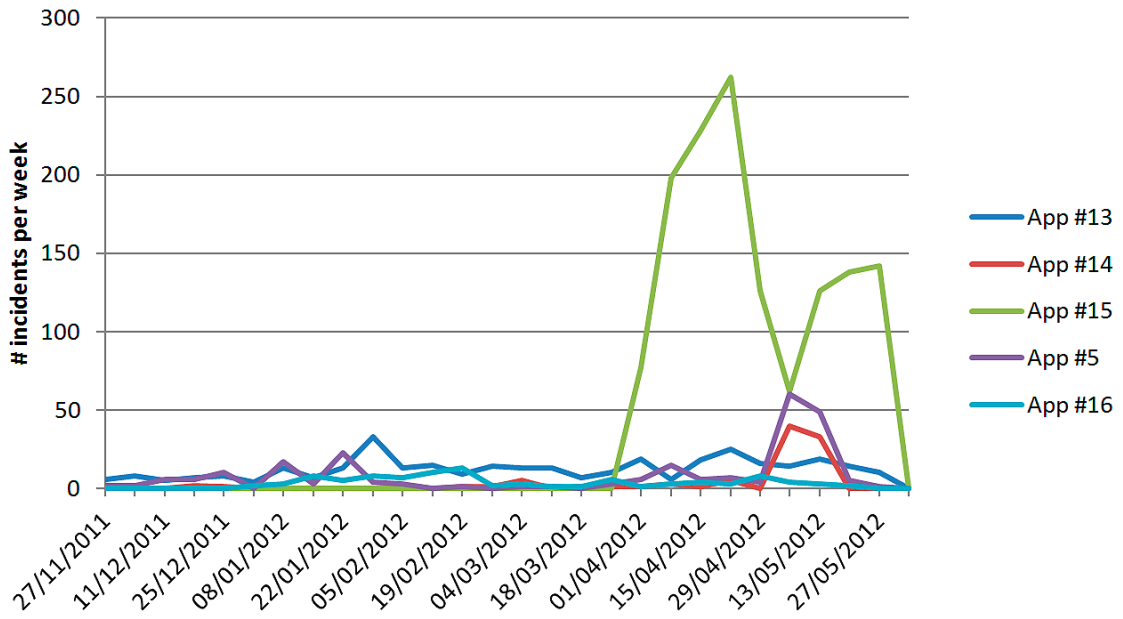


Figure 6 – Attack incidents of HTTP violations

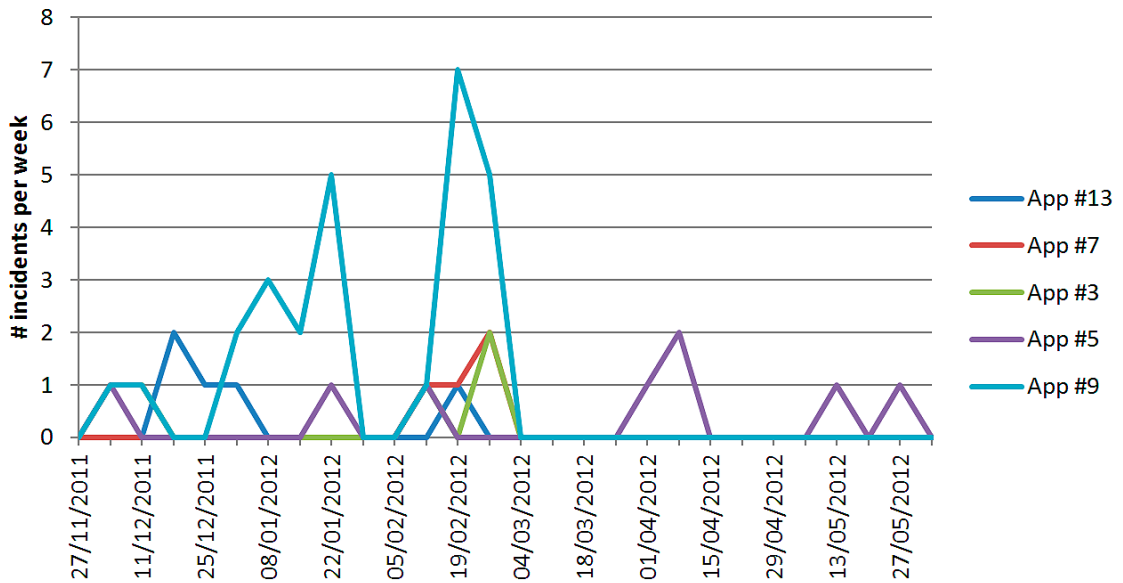


Figure 7 - Email Extraction attack incidents

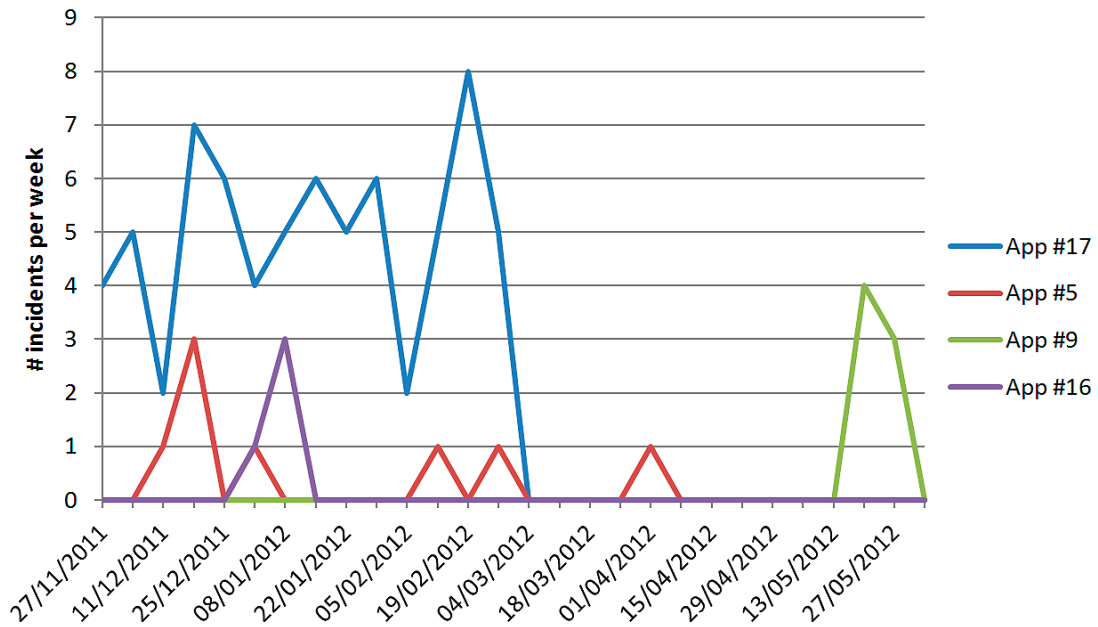


Figure 8 - Comment Spamming attack incidents

### 4.3 Comparative attack overview

In this section, we summarize the characteristics of each attack type, across the monitored applications. Only applications that suffered a substantial volume of attacks were included in the analyses (more than 1000 malicious HTTP requests in 6 months). After applying this threshold, the amounts of applications included in each category are as follows: 18 apps for SQLi, 18 apps for RFI, 15 for LFI, 12 for Directory Traversal, 17 for XSS, 18 for HTTP violations, and for business logic attacks, 10 apps for email extraction, and only 5 apps for comment spamming. The relatively low amount of application suffering from business logic attacks can be attributed to the nature of the attacks, which often don't require high burst rates. Therefore, it may be that such traffic didn't exceed our threshold for defining an attack incident. It is also important to keep in mind that business logic attacks strongly depend on the nature of the application itself, and are more likely to occur in applications with a lot of user-provided content like social networks, forums, and blogs.

#### 4.3.1 Amount of attack incidents

Table 2 – Amount of attack incidents

	Amount of attack incidents (incidents/6 months)						Amount of attack incidents (incidents/6 months)		
	SQLi	RFI	LFI	DT	XSS	HTTP	EmExt	ComSpm	
<b>Median</b>	17.50	8.00	5.50	13.00	6.00	27.00	<b>Median</b>	3.50	7.00
<b>Max</b>	320.00	119.00	55.00	135.00	49.00	1359.00	<b>Max</b>	27.00	70.00
<b>1st Quartile</b>	8.00	2.00	3.75	6.00	1.25	8.00	<b>1st Quartile</b>	2.25	4.00
<b>3rd Quartile</b>	53.25	23.00	11.50	26.00	16.25	68.75	<b>3rd Quartile</b>	5.00	8.00

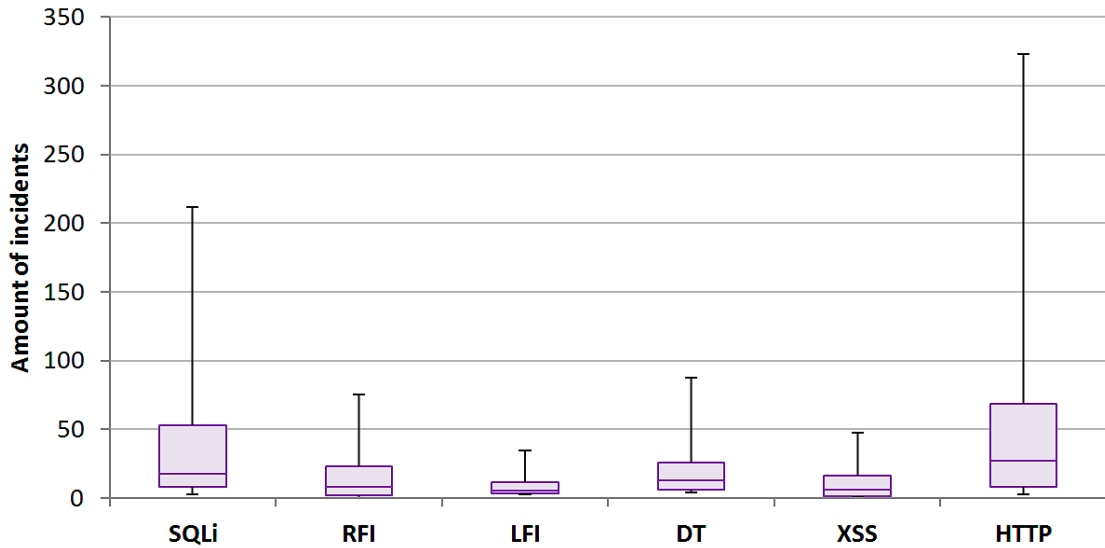


Figure 9 – Amount of attack incidents

### 4.3.2 Attack incident magnitude

Table 3 – Magnitude of attack incidents

	Mean attack magnitude (requests/incident)						Mean attack magnitude (requests/incident)		
	SQLi	RFI	LFI	DT	XSS	HTTP		EmExt	ComSpm
<b>Median</b>	189.04	307.50	413.93	233.47	144.18	190.58	<b>Median</b>	79.80	51.13
<b>Max</b>	4222.19	1178.00	771.83	1699.55	1651.45	8790.33	<b>Max</b>	362.60	305.29
<b>1st Quartile</b>	103.98	81.50	254.59	113.50	49.75	139.59	<b>1st Quartile</b>	46.50	43.68
<b>3rd Quartile</b>	435.51	664.84	507.59	558.25	320.39	281.79	<b>3rd Quartile</b>	191.50	119.82

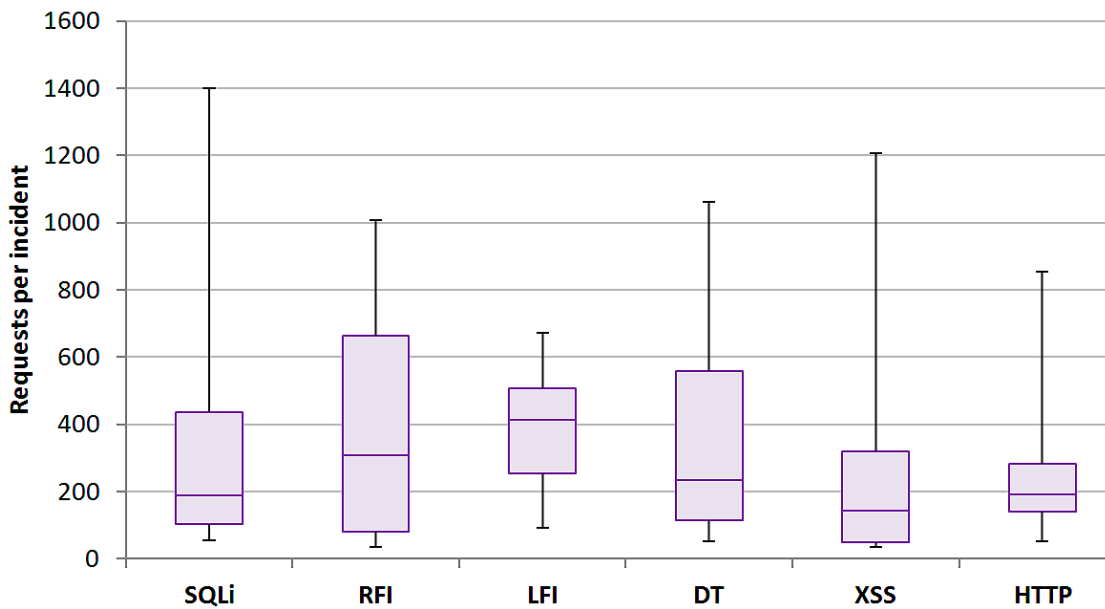


Figure 10 – Magnitude of attack incidents

As Table 3 shows, SQLi attacks have the highest magnitude. Although the median is not larger than that of the other attacks, the maximal magnitude of SQLi attack is much higher. This has to do with the nature of the attack, as SQLi attacks usually crawl as many resources as the victim application has, and involves many different vectors for each resource. The combinations for a SQLi attack are numerous, while not so for other attack types.

### 4.3.3 Attack incident duration

Table 4 – Duration of attack incidents

	Mean attack duration (min/attack)						Mean attack magnitude (requests/attacks)		
	SQLi	RFI	LFI	DT	XSS	HTTP		EmExt	ComSpm
<b>Median</b>	8.39	7.50	7.83	10.00	6.67	9.23	<b>Median</b>	5.63	5.07
<b>Max</b>	29.76	59.53	26.25	23.75	78.94	50.26	<b>Max</b>	9.00	12.14
<b>1st Quartile</b>	6.84	6.46	6.63	7.33	5.00	7.40	<b>1st Quartile</b>	5.00	5.00
<b>3rd Quartile</b>	12.50	10.83	12.26	14.17	9.29	11.13	<b>3rd Quartile</b>	7.50	6.89

From this table, it is evident that attacks are usually a burst event, with the medians of the attacks going from 8 to 10 minutes per attack. As the maximal values show, there certainly are longer attacks, lasting half an hour or more, but the majority are well below 15 minutes.

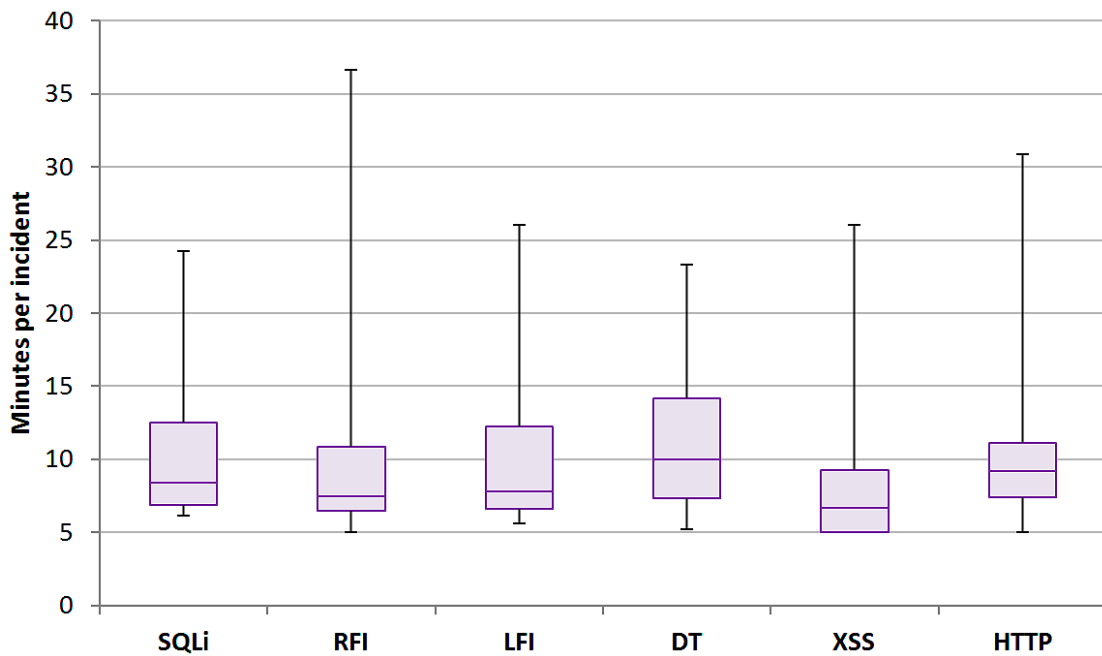


Figure 11 – Duration of attack incidents

### 4.3.4 Battle days

Table 5 – Battle days in 6 months

Battle days (Battle days/6 months)						Battle days (Battle days/6 months)			
	SQLi	RFI	LFI	DT	XSS	HTTP		EmExt	ComSpm
Median	13.50	2.00	3.50	8.00	3.00	18.00	Median	3.50	3.00
Max	110.00	50.00	30.00	84.00	32.00	140.00	Max	19.00	48.00
1st Quartile	4.50	1.00	3.00	4.00	1.00	6.50	1st Quartile	2.00	3.00
3rd Quartile	21.00	9.50	8.00	15.00	6.00	41.25	3rd Quartile	5.00	8.00

This table summarizes the amount of battle days – Days during which an application was attacked at least once, across a half-year period.

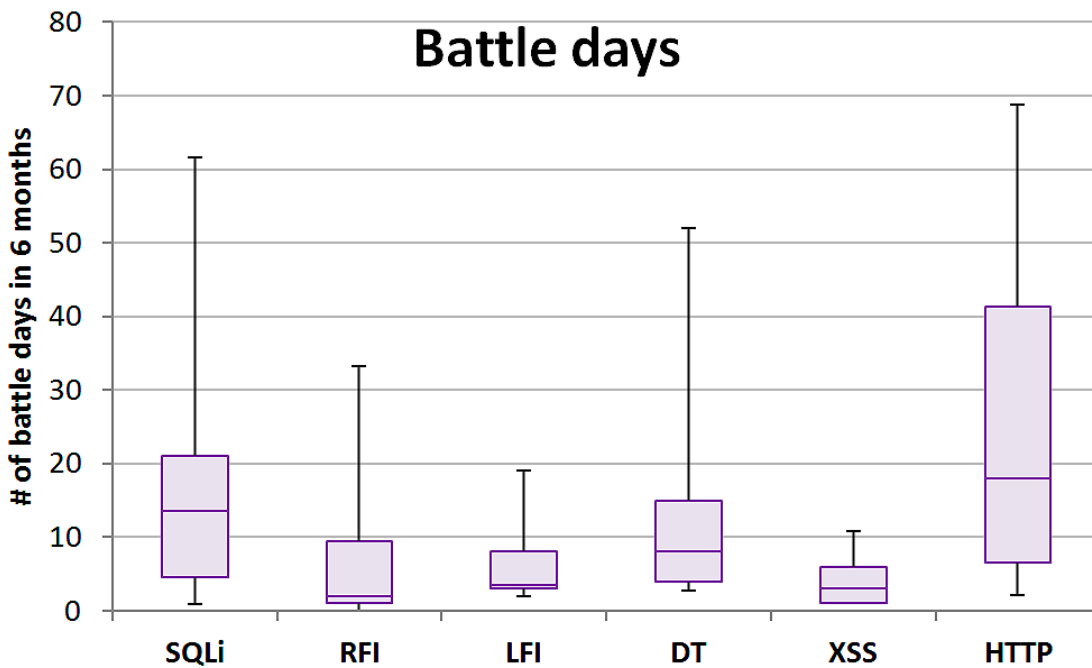
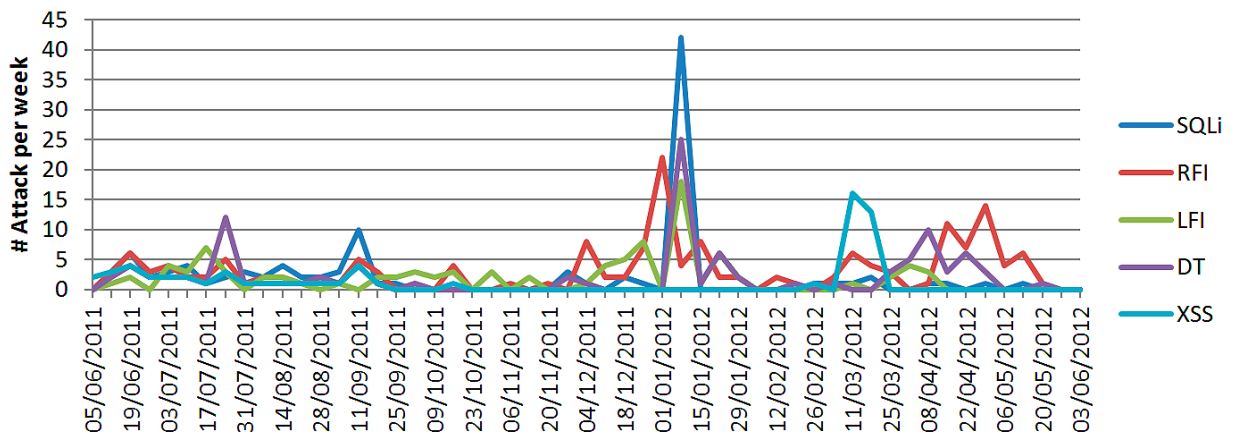


Figure 12 – Battle days in 6 months

### 4.4 Single application case study- full year

Here we have analyzed the amount of attack incidents on a single application.

Figure 13 – App #5 attack incidents in 12 months



For this application, we analyzed the traffic of a full year from June 2011 to May 2012. As the chart shows, the data is very bursty, dominated by seemingly random attack peaks. It is noteworthy to mention the campaign that took place on the beginning of January, in which the application was attacked using SQLi, directory Traversal and LFI methods simultaneously. Even though, it is also evident from the graph that the overall amount of attack incidents increased during the last 6 months. When summing the attack incidents of all attack types, during the first 6 months, the app had 322 attack incidents, compared to 576 in the next months. This difference still holds after excluding the large attack incident of the beginning of January.

In addition, the amount of HTTP violation attack incidents (not shown in graph) increased considerably in the second half of the year when compared to the first. This supports the notion that HTTP violations are often an indicator of other malicious activity and should therefore be closely monitored.

#### 4.5 Worst-case scenarios

Table 6 – Worst case scenarios, maximal observed values

	SQLi	RFI	LFI	DT	XSS
<b>Attack incident magnitude (requests)</b>	359390	35276	3941	8197	16222
<b>Attack rate (requests per minute)</b>	543.2	742.2	418.4	378	455.4
<b>Requests per battle day</b>	359465	41495	8343	11549	21113

This table summarizes the maximal values observed for each attack type. The maximal values in the table represent the largest attack observed in the entire dataset; “the worst case scenario” of our data.

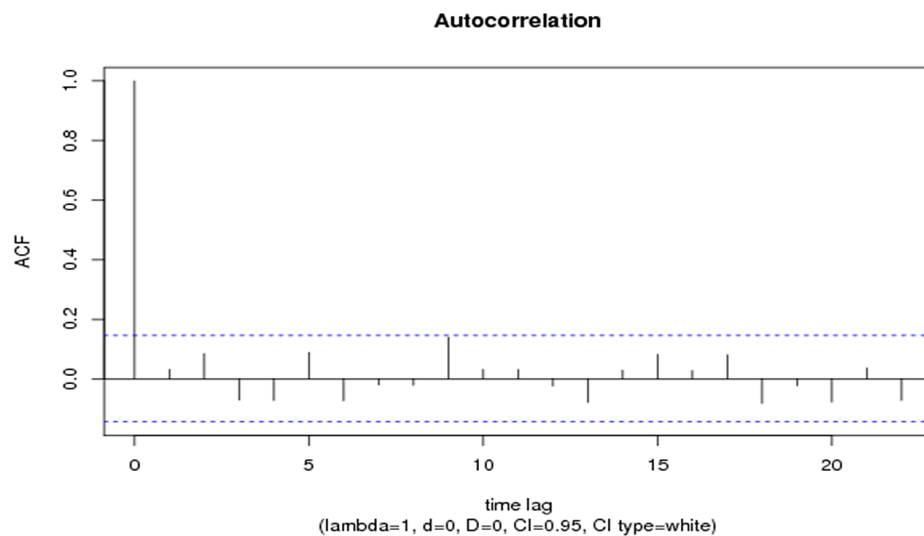
#### 4.6 Battle days predictions

Predicting an upcoming attack is highly desirable as it would enable the defender to be more prepared. We had tried to test for the applicability of such prediction for “battle days” using linear prediction, which is a common method to estimate future values of a discrete-time signal as a linear function of previous samples.<sup>3</sup>

Linear prediction is based on the signal Autocorrelation Function (ACF). Autocorrelation is the cross-correlation of a signal with itself. Informally, it is the similarity between observations as a function of the time separation between them.<sup>4</sup>

Using a freely available autocorrelation service<sup>5</sup>, we had found out that mostly battle days cannot be effectively predicted with linear prediction as the typical graph of the ACF shows that there is very little correlation between observations. The graph values on time lags other than zero are very low compared to the value on the zero lag. Which really suggest we cannot learn too much on the future battle days based on observations on battle day’s history.

Figure 14 – ACF chart for App #13



<sup>3</sup> [http://en.wikipedia.org/wiki/Linear\\_prediction](http://en.wikipedia.org/wiki/Linear_prediction)

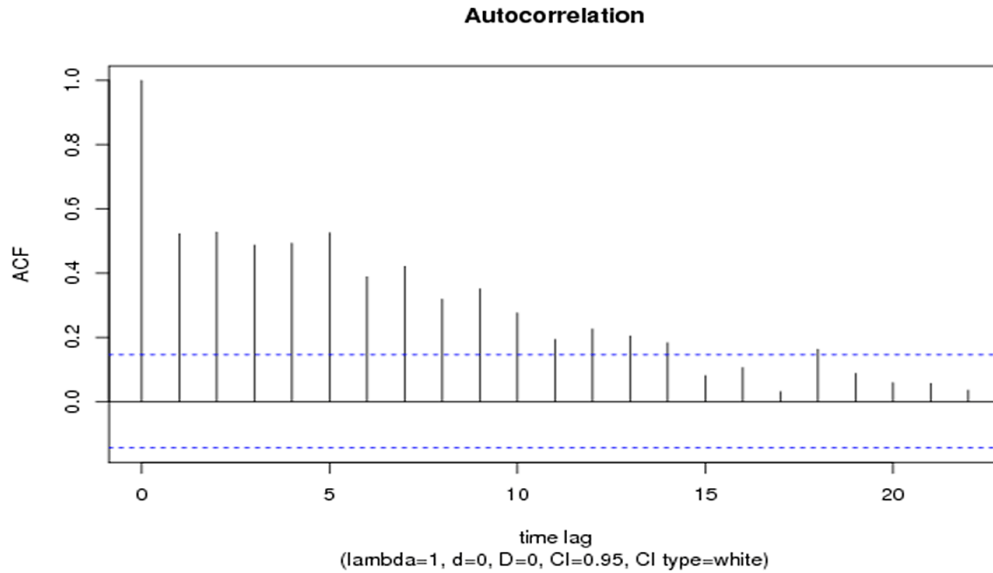
<sup>4</sup> <http://en.wikipedia.org/wiki/Autocorrelation>

<sup>5</sup> Wessa, P. (2012), Free Statistics Software, Office for Research Development and Education, version 1.1.23-r7, <http://www.wessa.net/>



However, the ACF on a specific application showed some very different results.

Figure 15 – ACF chart for App #11



Working with application owner, we were able to determine that the source for the correlation was some previously unreported friendly web application vulnerability scanning service, that generated traffic on some predictable manner. Even though this case does not represent a genuine attack, it really shows the effectiveness of the ACF analysis in finding patterns within attacks.

## 4.7 Geographic Dispersion

We have analyzed the geographic distribution of the attack initiating hosts, as determined by their IP address. The following tables summarize the top 10 countries from which the largest volume of HTTP requests were sent.

Table 7: Countries from which most attack requests were initiated (requests count in thousands)

RFI		SQLi		DT		LFI		EmExt		ComSpm	
Country	Requests (1000's)	Country	Requests (1000's)	Country	Requests (1000's)	Country	Requests (1000's)	Country	Requests (1000's)	Country	Requests (1000's)
USA	150	France	803	USA	342	USA	40	Senegal	14	Russian Federation	31
United Kingdom	47	USA	232	Canada	35	China	18	European Union	14	Ukraine	14
France	21	China	24	Germany	32	Germany	10	USA	10	Germany	9
Sweden	15	Netherlands	22	United Kingdom	18	France	10	Ivory Coast	8	USA	9
Germany	11	Mexico	21	Ukraine	9	Canada	9	Malaysia	4	China	8
Canada	9	Bulgaria	15	Brazil	7	Brazil	7	Italy	3	Latvia	8
Spain	8	Albania	10	China	6	Poland	7	Nigeria	2	United Kingdom	4
Italy	7	Ukraine	9	Japan	5	United Kingdom	5	Ghana	2	Poland	2
Turkey	5	Germany	9	France	5	Italy	5	Germany	2	Netherlands	2
Netherlands	4	United Kingdom	8	Russian Federation	3	Colombia	5	Thailand	1	France	2

Table 8: Countries with the most distinct attacking hosts

RFI		SQLi		DT		LFI		EmExt		ComSpm	
Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers
USA	17176	USA	16041	USA	7802	USA	842	Senegal	3627	USA	1470
United Kingdom	1859	China	723	United Kingdom	1605	Germany	126	Ivory Coast	1673	China	1365
Canada	1598	Russian Federation	389	China	1242	France	119	European Union	849	Russian Federation	995
Australia	789	United Kingdom	281	Russian Federation	1041	Republic of Korea	76	USA	488	Ukraine	825
Germany	572	Ukraine	205	Ukraine	532	Brazil	65	India	364	Germany	407
France	426	Canada	140	Canada	442	Canada	53	Thailand	358	Sweden	207
Sweden	294	India	105	Germany	382	United Kingdom	49	Ghana	322	European Union	206
Netherlands	233	European Union	85	Brazil	361	Netherlands	48	Malaysia	307	Thailand	185
Japan	218	Germany	71	France	171	Russian Federation	47	Indonesia	305	France	179
Republic of Korea	193	France	64	European Union	168	European Union	42	Nigeria	197	United Kingdom	174

As demonstrated by these tables, the trends reported in the previous WAAR remain relevant, with the majority of requests and attackers originating in the USA, western European countries, China, and Brazil. For business logic attack types, email extraction is still widely dominated by western African countries, like Senegal, Nigeria, Ghana, and the Ivory Coast. The field of comment spamming shows unusual bias towards former eastern-bloc countries, such as Russia, Ukraine, Latvia, and Poland.

A consistent finding throughout our reports, that more often than not, the largest amount of attack requests originates from IPs allocated to the USA. Therefore, it was very surprising to find out that France has earned the dubious title of being the leader in SQLi attacks – as attack volume of requests originated from France was almost four times greater than that of the USA.

## ⑤ Conclusions and Recommendations

Our recommendations from previous reports still remain valid:

- › Deploy security solutions that detect automated attacks. This detection must be done as early as possible during the attack.
- › Detect and block attacks against known vulnerabilities. The knowledge base of exploitable weaknesses in the application must be frequently updated.
- › Acquire intelligence on malicious sources and apply it in real time. Black lists of attacking hosts are still an efficient counter measure. However, the lists must be up to date to be effective.
- › Participate in a security community and share data on attacks. The increased automation and scale of attacks leave a large footprint on the web – but it can only be seen by looking at data gathered from a large set of potential victims.

Based on the analysis made for this report we would like to add the following recommendations:

- › Attack distribution is burst orientated and far from normally distributed.
  - Don't be fooled by relative average calm of the battlefield. As you typically would witness a "battle day" only on one day out of three, and it typically would last just a few minutes. However the way your security solution and process would perform on these minutes really determines your overall security performance. So, base your estimations for the security measures you need on the worst-case scenario and not on the average case.
  - Be certain that the security procedures and solutions are as automated as possible, since:
    - Attack volume is too overwhelming for humans
    - Typically, there will be no advance warning
  - Test your readiness to accommodate such bursty threats by simulating them. This is probably the best way to find out if your security solutions and procedures are adequate.
- › Generally, attacks are not predictable – security team should be prepared for mitigating them without any advance notice.
- › Good intelligence saves lives on the cyber battlefield.
  - Learn from your own experience – analyze the data on attacks against your application using some advanced statistical tools, such as autocorrelations to identify patterns and gain insights on the attacks your applications experience.
  - Learn from the experience of others – get reputation services to quickly detect and mitigate attacks as shown on our previous reports.

**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](http://www.imperva.com)

© Copyright 2012, Imperva  
 All rights reserved. Imperva, SecureSphere, and "Protecting the Data That Drives Business" are registered trademarks of Imperva.  
 All other brand or product names are trademarks or registered trademarks of their respective holders. #HII-SA-SECURITY-SUMMARY#3-0712rev1

