

Imperva Web Application Attack Report

Edition #4 – July 2013

Table of Contents

1 Abstract	3
2 Executive Summary	4
3 Analysis Methodology	6
3.1 Data Corpus	6
3.2 Updates in Data Analysis and Presentation Methodology	6
3.3 Analysis Glossary	6
4 Analysis Results	7
4.1 Overview	7
4.2 Comparative Attack Overview	7
4.2.1 Number of Attack Incidents	8
4.2.2 Attack Incident Magnitude	8
4.2.3 Attack Incident Duration	9
4.2.4 Battle Days	10
4.2.5 Requests per Battle Day	11
4.3 Worst Case Scenarios	12
4.4 Comparing Vertical Industries	12
4.5 Geographic Dispersion	13
5 Conclusions and Recommendations	15
6 Attack Glossary	16
6.1 SQL Injection	16
6.2 Remote File Inclusion	16
6.3 Local File Inclusion	16
6.4 Directory Traversal	16
6.5 Cross-Site Scripting	16
6.6 Email Extraction	16
6.7 Comment Spamming	16

① Abstract

The application threat landscape can be described as a cyber war. In this report, we explore the technical details of this war – the methods, intensity, and duration of attacks witnessed in cyber battlefields across the globe. Armed with this report, security officers can understand the threats they face and prepare for future attacks.

This Web Application Attack Report (WAAR) identifies how many attacks a typical application can expect to suffer annually. In addition, it exposes which countries perpetrated the most attacks and compares application risks by industry. Most importantly, this report reveals the underlying distribution of attacks, presenting an accurate picture of today's application threat landscape.

Key findings:

- › Retailers suffer 2x as many SQL injection attacks as other industries.
- › While most web applications receive 4 or more web attack campaigns per month, some websites are constantly under attack.
 - One observed website was under attack 176 out of 180 days, or 98% of the time.
- › Imperva observed a single website receive 94,057 SQL injection attack requests in one day.
 - 94,057 equates to 1,567 SQL injection attacks per hour or 26 attack requests per minute, on average
- › The United States retains its rank as the number one source of web attacks.

2 Executive Summary

Our key web application attack statistics are summarized in Table 1.

Table 1 – Key Statistics Summary

	Median	Maximum
Application battle days during a six month period	12	176
Attack incident duration (in minutes)	5	935

While most applications suffered high-severity attacks for 12 days during a 6-month period, or once every 15 days, the duration and the number of attack campaigns varied widely, with some web applications under attack virtually every day. Each attack incident could consist of hundreds or even thousands of individual attack requests.

Table 2 – Comparison of Key Statistics to WAAR 3

	WAAR 4 July 2013		WAAR 3 July 2012	
	Median	Maximum	Median	Maximum
Number of attack incidents per application during a six month period	25	3,006	137	1,383
Number of individual attacks per incident	85	148,089	195	359,390

When compared to Imperva's previous WAAR report, published in July 2012, the maximum number of attack incidents grew, while the median number of attack incidents and the intensity of the attacks lessened year over year. This decrease can be attributed to the addition of more applications to our data corpus; specifically, applications with smaller traffic volumes. The results are summarized here and discussed in depth in Section 5 of this report.

The growth of our data corpus enabled us to investigate and compare attack patterns within different vertical industries.

Figure 1 – Relative Portions of Each Attack Type in Retail Web Applications vs. Other Web Applications

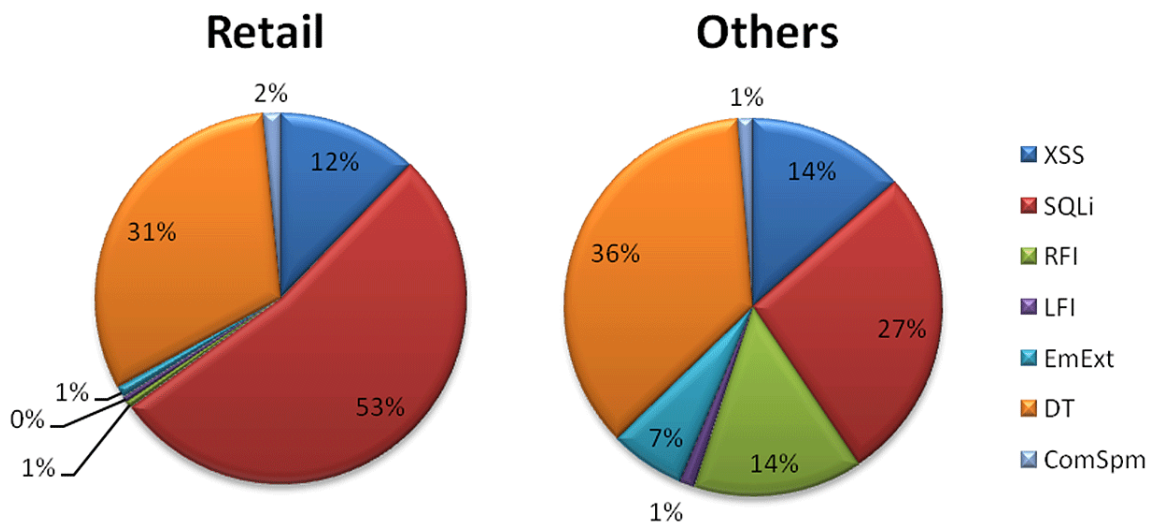


Figure 1 illustrates the types of web attacks targeting retail web applications compared to attacks targeting other industries. When compared to other industries, retail applications suffered twice as many SQL injection attacks, but fewer Remote File Inclusion (RFI) attacks.

Following these results, we further investigated the characteristics of SQL injection attacks.

Our analysis, shown in Table 3, revealed that SQL injection attacks on retail applications were more intense, both in terms of number of attacks per incident and duration of an incident. In fact, retail applications received 749 individual attack requests per attack campaign.

Table 3 – Analysis of Magnitude and Duration of SQL Injection Attack Incidents

	Magnitude per Incident (Requests)		Duration per Incident (Minutes)	
	Average	Maximum	Average	Maximum
Retail	749	46,027	22	575
Other Industries	298	7,700	12	260

The geographic attack trends reported in the previous WAAR remain relevant, with the majority of requests and attackers originating in the United States, Western European countries, China, and Brazil. For business logic attacks, email extraction is still widely dominated by African countries, such as Senegal, Nigeria, Ghana, and the Ivory Coast. The field of comment spamming shows an unusual proportion toward Eastern European countries, such as Russia, Ukraine, Latvia, and Poland. Besides this geographic dominance, more and more business logic attacks are originating from Asia and South America. Countries generating an increasing number of business logic attacks include Malaysia, Thailand, Pakistan, Mexico, Brazil, and Argentina.

③ Analysis Methodology

3.1 Data Corpus

This security summary report is based on observing and analyzing Internet traffic to 70 web applications during a period of six months. Compared to the 2012 WAAR, the number of analyzed applications doubled.

We identified security attacks targeting these applications, categorized them according to the attack method, and identified patterns and trends within these attacks.

Automated tools monitored and recorded web application traffic. Security event logs were analyzed using Imperva's special-purpose software and knowledgebase. This analysis used several measures, including matching events to known attack signatures, comparing attack sources to black lists of malicious hosts, and reviewing specific attributes of malicious traffic. Imperva's security experts performed additional detailed analysis of important events and patterns.

3.2 Updates in Data Analysis and Presentation Methodology

The analysis and presentation methodology in this report follows the changes we applied starting with our July 2012 WAAR.¹

Similar to our last report, we defined an attack incident to consist of at least 30 requests in five minutes (on average, one attack request every 10 seconds). Of course, a single attack may span several consecutive five-minute periods. We also defined an even broader concept of "battle days." Battle days 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 of which have very 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?
- › How many "battle days" should a company expect during the next six to twelve months?
- › If a company 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, starting with WAAR 3, we use 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. For a more thorough discussion, see the Methodology section in WAAR 3².

After analyzing six 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.

3.3 Analysis Glossary

- › **Attack request** – A single, malicious HTTP request.
- › **Attack incident** – Attacks are burst-like in nature. Each burst that exceeded the rate of 30 attack requests per a five minute period was defined as an attack incident.
- › **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.

¹ http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed3.pdf

² http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed3.pdf

④ Analysis Results

4.1 Overview

Our key statistics on web application attacks are summarized in Table 4. The statistics are summarized for all tested applications and for all attack types.

Table 4 – Key Statistics Summary

	Median	Maximum
Application battle days during a 6-month period	12	176
Attack incident duration (in minutes)	5	935

A typical application experienced 12 battle days, that is, days in which at least one attack incident occurred. In comparison, the worst case was 176 battle days in six months, meaning one of the tested applications suffered attack incidents in almost each and every day within this time period. Another interesting finding was that while the typical attack incident lasted around five minutes, the worst case incident was about 100 times longer, lasting more than 15 hours.

Table 5 – Comparison of Key Statics to WAAR 3

	WAAR 4 July 2013		WAAR 3 July 2012	
	Median	Maximum	Median	Maximum
Number of attack incidents per application during a six month period	25	3,006	137	1,383
Number of individual attacks per incident	85	148,089	195	359,390

Comparing the results of this report to those of the July 2012 WAAR, the maximal values, or worst cases, are in the same order of magnitude. The maximal number of battle days a single application suffered was higher, as was the number of attack incidents. Moreover, the application that suffered the maximal amount of attack incidents and battle days remained the same application. This suggests stability in the attack patterns of a single application, as well as in traffic volume.

In contrast, the maximal attack magnitude and attack duration are smaller in this report than in the previous analysis. The typical values in our data represent the changes in the web applications that compose our data corpus, and not an actual decrease in attack occurrence.

When this analysis is compared to that of the previous WAAR, despite the addition of new applications, some of the applications with the highest attack incident numbers remained the same as in the previous period. In other words, some applications are attacked more than others, and these remain stable over time.

4.2 Comparative Attack Overview

In this section, we summarize the characteristics of each attack type across the monitored applications. In the previous report, only applications that suffered a substantial volume of attacks were included in our analyses (more than 1,000 malicious HTTP requests in six months). In this report, we decided to include all monitored applications in the analysis. This change might be responsible in part for the decrease in the calculated median values of attack incidents and their magnitude, as applications with very little traffic were also taken into account. We believe that this provides us with a more accurate picture.

The relatively low number of business logic attack incidents identified in this report can be attributed to the nature of the attacks, which often don't require high rate bursts. Therefore it may be that such traffic didn't exceed our threshold for defining an attack incident. In future research, it might be beneficial to treat business logic attacks with different criteria, e.g., lower thresholds, slower rates, etc. 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, such as social networks, forums, and blogs.

4.2.1 Number of Attack Incidents

Table 6 – Number of Attack Incidents

Attack Incidents During 6-Month Period								
	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting	HTTP Violations	Email Extrusion	Comment Spam
Median	10	3	1	7	7	11	3	2
Maximum	209	98	8	193	85	2898	81	13
1st Quartile	4	3	1	3	2	2	2	2
3rd Quartile	30	10	3	24	14	43	6	6

Figure 2 – Number of Attack Incidents

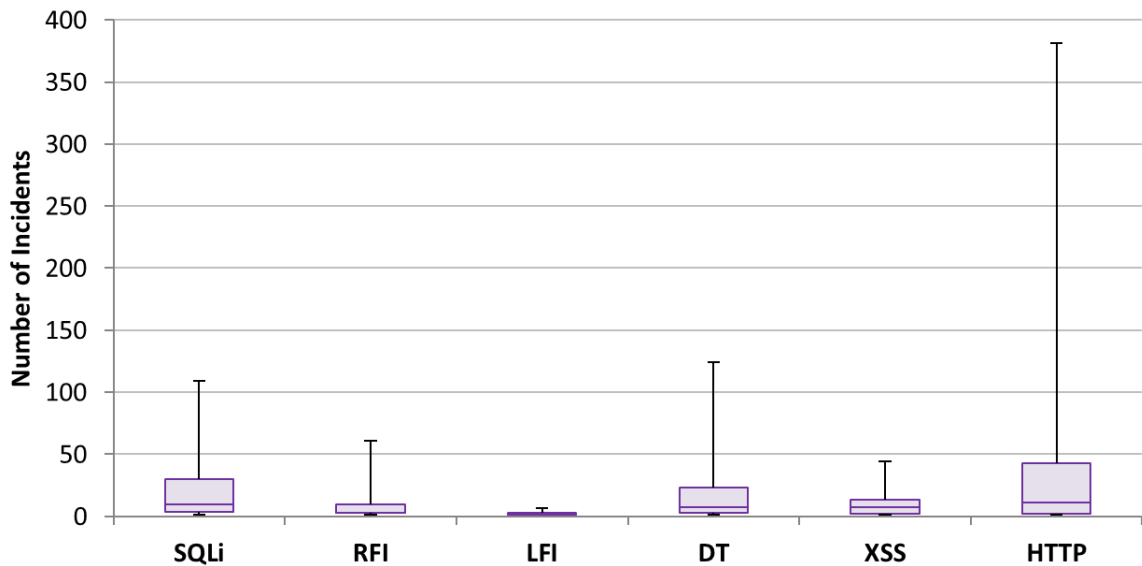


Figure 2 illustrates the maximum number of attack incidents identified per application during a 6-month period, as well as the median number of attacks (the middle line in the box) and the first and third quartile figures (the bottom and top lines of each box, respectively). The most prevalent types of attacks are SQL injection, and directory traversal HTTP protocol violations, which often indicate automated threats, evasion techniques, and denial of service attacks.

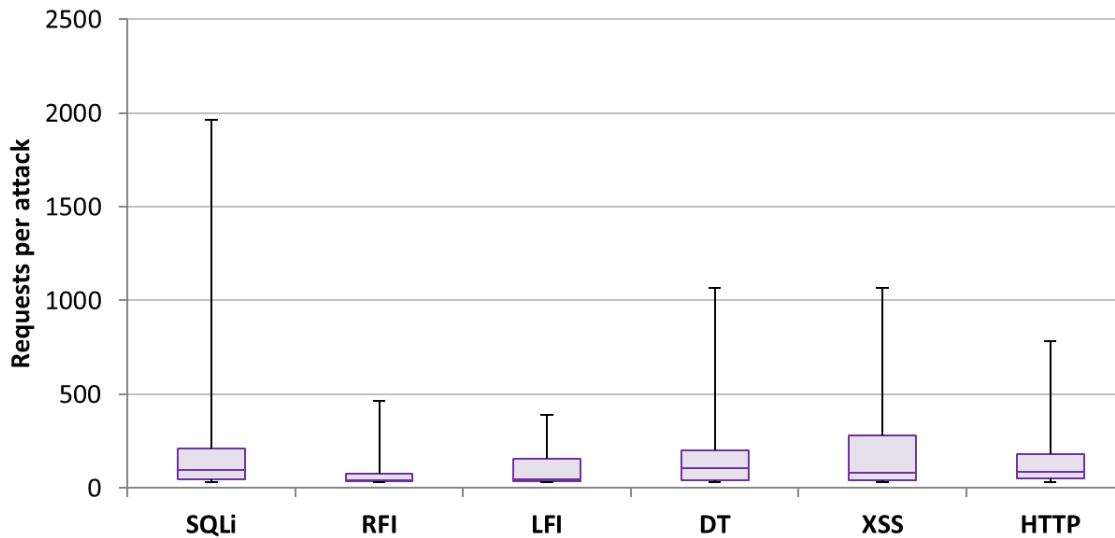
4.2.2 Attack Incident Magnitude

Table 7 – Magnitude of Attack Incidents

Attack Requests Per Incident								
	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting	HTTP Violations	Email Extrusion	Comment Spam
Median	98	42	49	75	80	87	72	224
Maximum	46,027	2,472	867	11,756	19,324	148,089	4,860	36,390
1st Quartile	48	35	35	39	40	50	53	68
3rd Quartile	212	76	158	200	278	178	264	232

In the previous report, we calculated the average number of requests per attack incident for each application, and then performed statistical analyses of the averages. For this report, we describe the distribution of all attack incidents, regardless of the application in which they had occurred, which we believe provides a more thorough assessment.

Figure 3 – Magnitude of Attack Incidents



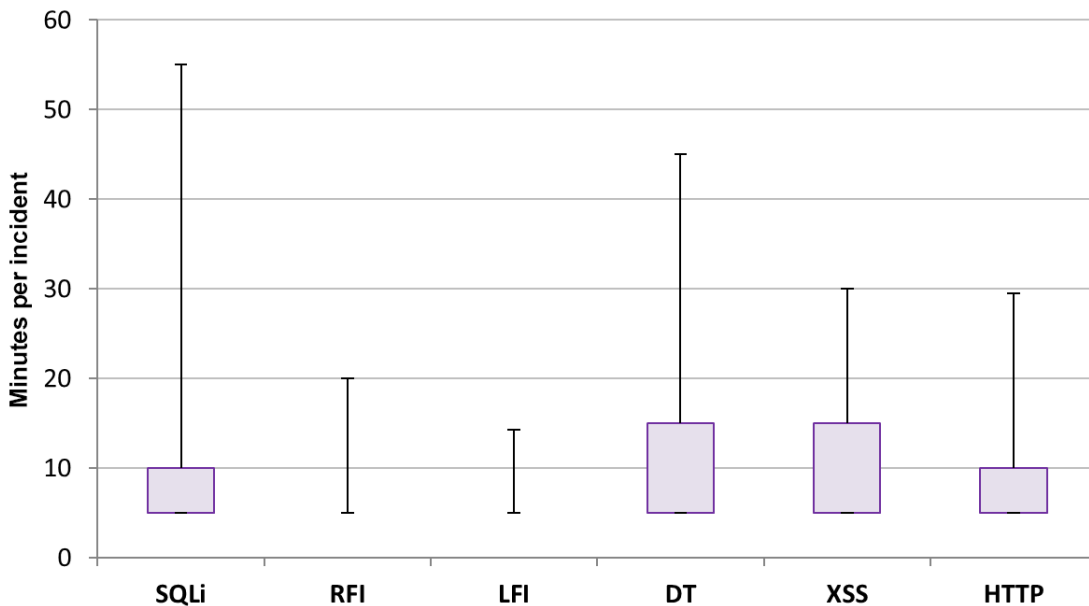
4.2.3 Attack Incident Duration

Table 8 – Duration of Attack Incidents

Attack Duration in Minutes per Incident								
	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting	HTTP Violations	Email Extrusion	Comment Spam
Median	5	5	5	5	5	5	5	5
Maximum	575	135	20	360	195	575	50	935
1st Quartile	6	5	5	5	5	5	5	5
3rd Quartile	11	5	5	15	15	10	10	10

In Table 8, it is evident that attacks are usually a burst event, with the medians number of attack incidents lasting between 5 to 10 minutes. As the maximal values show, there certainly are longer attacks lasting several hours, but most attacks are well below 15 minutes.

Figure 4 – Duration of Attack Incidents



For Remote File Inclusion (RFI) and Local File Inclusion (LFI) attacks, the distribution of attack duration is so narrow that more than 75% of the attack incidents lasted five minutes or less. These categories don't have a proper "box" in Figure 4, since the first and third quartiles have the same value. For the other attack types, the median is also five minutes, which is our minimal value for calculation. Thus, the distribution is skewed towards the bottom.

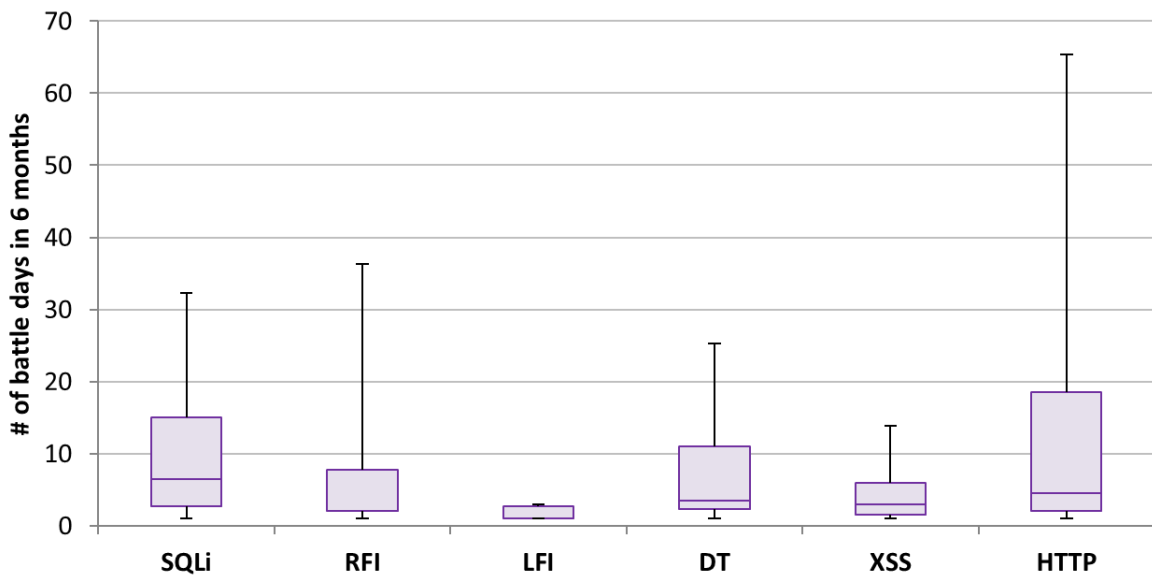
4.2.4 Battle Days

Table 9 and Figure 5, together, show the number of days over a 6-month period which an application suffered an attack incident.

Table 9 – Battle Days per Six Months

Battle Days During a 6-Month Period								
	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting	HTTP Violations	Email Extrusion	Comment Spam
Median	7	2	1	4	3	5	3	1
Maximum	96	49	3	69	14	176	53	2
1st Quartile	3	2	1	2	2	2	2	1
3rd Quartile	15	8	3	11	6	19	4	2

Figure 5 – Battle Days per Six Months



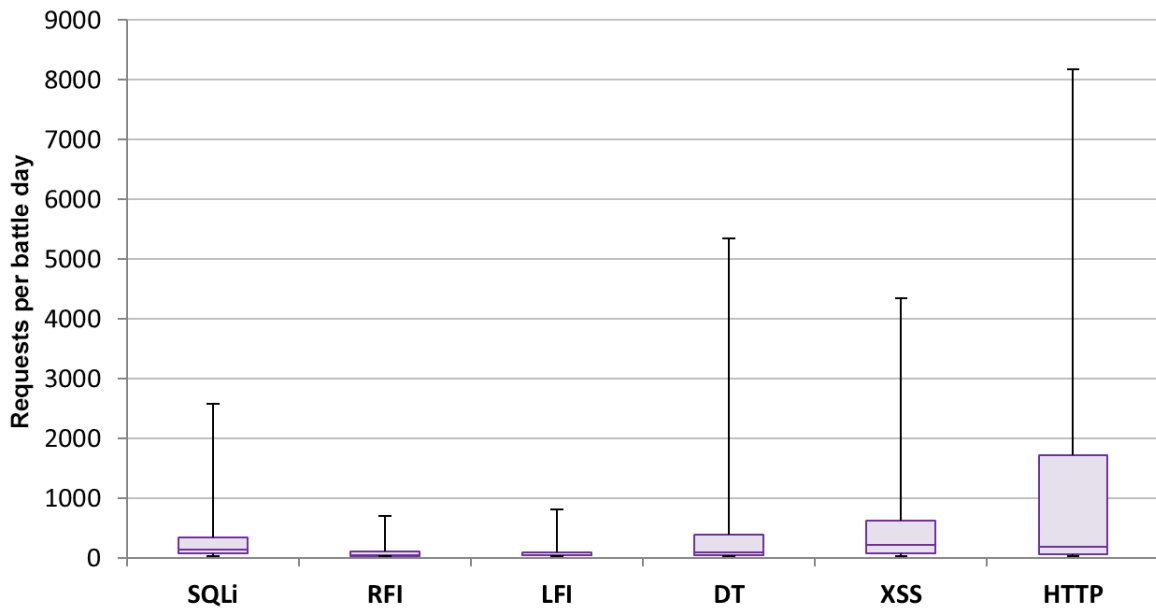
4.2.5 Requests per Battle Day

Table 10 and Figure 6 portray the number of malicious HTTP requests sent to an application during a single battle day.

Table 10 – Attacks per Battle Day

Attack Requests Battle Day								
	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting	HTTP Violations	Email Extrusion	Comment Spam
Median	144	48	47	94	210	189	110	415
Maximum	94,057	4,255	1,366	22,013	19,977	344,059	5,703	37,473
1st Quartile	69	35	37	43	80	61	55	93
3rd Quartile	349	110	89	385	629	1715	383	1810

Figure 6 – Attack Requests per Battle Day



4.3 Worst Case Scenarios

Table 11 summarizes the maximal values observed for each attack type. The maximal values in the table represent the largest attack observed in the entire dataset; that is, “the worst case scenario” of our data.

Table 11 – Worst Case Scenarios, Maximal Observed Values

	SQL Injection	Remote File Inclusion	Local File Inclusion	Directory Traversal	Cross-Site Scripting
Maximum Attacks per Incident	46,027	2,472	8,67	11,756	19,324
Attack Rate (Attacks per Minute)	185	124	173	191	201
Maximum Attacks per Battle Day	94,057	4,255	1,366	22,013	19,977

4.4 Comparing Vertical Industries

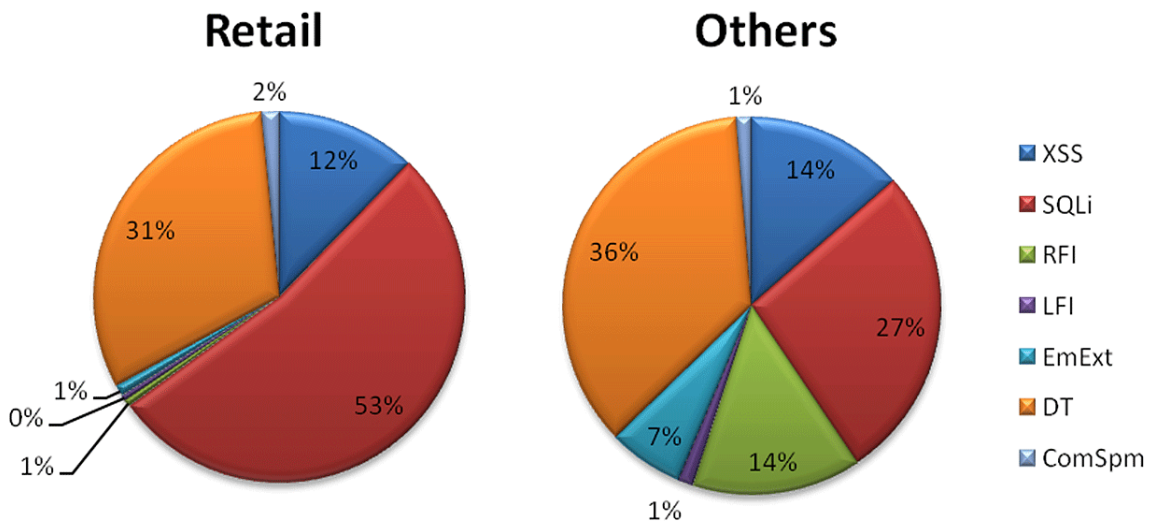
The growth of our data corpus enabled us to investigate and compare attack patterns within different vertical industries. Not all applications are the same, and different industries tend to have different characteristics and attract different attackers. A recent study by Whitehat Security³ revealed that websites from the retail industry have more security vulnerabilities than any of the other 11 tested industries, including banking, financial services, healthcare, energy, and education. Since they often process credit card data and they contain more serious vulnerabilities than other industries, retail applications are the perfect targets for cyber-attacks.

Our data provides an opportunity to look at differences between industries from another angle. While Whitehat statistics are based on vulnerabilities found, but not necessarily exploited, our data represents actual exploitation attempts – regardless of the existence of the vulnerability in the targeted web application.

Figure 7 illustrates the differences in the relative portions of attack types in retail web applications versus applications of other industries. When compared to other industries, retail applications suffered twice as many SQL injection attacks. They received fewer Directory Traversal and Cross-site Scripting (XSS) attacks, and the portion of RFI attacks is dramatically lower: 1% of attacks in retail applications, compared to 14% of attacks in other tested applications.

It is also interesting to note that although still small, the relative amount of Comment Spam activity in retail applications is twice as high as other industries’ figures. This can be attributed to the nature of retail websites, which often provide room for user comments, product reviews, recommendations, and complaints.

Figure 7 – Relative Portions of Each Attack Type in Retail Web Applications vs. Other Applications



³ <https://www.whitehatsec.com/resource/stats.html>

Following these results, we further investigated the characteristics of SQL injection attacks. We compared the distribution of SQL injection attack incidents on retail applications to other applications, with regard to attack magnitude and duration. Our analysis revealed that SQL injection attacks on retail applications consisted of more HTTP requests and lasted longer than SQL injection attacks on other applications. This finding can be attributed to the design and size of the applications. For example, it is plausible to assume that retail applications contain a relatively large number of pages in the form of online catalogs, and that this factor may have contributed to the length and the intensity of SQL injection attacks.

Table 12 – Analysis of Magnitude and Duration of SQL Injection Attack Incidents

	Magnitude per Incident (Requests)		Duration per Incident (Minutes)	
	Average	Maximum	Average	Maximum
Retail	749	46,027	22	575
Other Industries	298	7,700	12	260

4.5 Geographic Dispersion

We have analyzed the geographic distribution of the attack initiating hosts, as determined by their IP addresses. Tables 13 and 14 summarize the top 10 countries where the largest volume of HTTP requests initiated.

*Table 13 – Countries Where Most Attack Requests Were Initiated
(Requests in Thousands)*

Remote File Inclusion		SQL Injection		Directory Traversal		Local File Inclusion		Email Extrusion		Comment Spam	
Country	Requests	Country	Requests	Country	Requests	Country	Requests	Country	Requests	Country	Requests
United States	82	United States	803	United States	594	United States	20	Senegal	50	United States	42
France	22	China	46	Philippines	26	France	11	China	34	China	5
Germany	9	Netherlands	22	China	18	Republic of Korea	6	United States	32	Mexico	4
Brazil	8	Germany	17	Germany	13	Bangladesh	2	Ivory Coast	21	Turkey	4
Republic of Korea	7	India	16	Canada	8	Brazil	2	European Union	19	Pakistan	3
United Kingdom	6	Indonesia	12	Russian Federation	7	United Kingdom	2	Malaysia	12	Ukraine	3
Netherlands	6	Russian Federation	12	Sweden	6	Germany	1	Ukraine	10	Russian Federation	2
Turkey	6	United Kingdom	8	United Kingdom	4	Russian Federation	1	Brazil	8	Ireland	2
Sweden	6	Canada	7	Indonesia	3	Singapore	1	Germany	4	European Union	1
Russian Federation	6	Turkey	7	France	2	Turkey	1	Ghana	4	Argentina	1

Table 14 – Countries with the Most Distinct Attacking Hosts

Remote File Inclusion		SQL Injection		Directory Traversal		Local File Inclusion		Email Extrusion		Comment Spam	
Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers	Country	Attackers
United States	2893	United States	11662	United States	9366	United States	507	Senegal	2472	China	1676
Brazil	551	China	1761	China	2944	France	85	Ivory Coast	2318	United States	753
Germany	357	United Kingdom	1192	Russian Federation	930	Germany	80	United States	881	Russian Federation	434
Russian Federation	252	Ukraine	1182	Ukraine	478	Republic of Korea	74	European Union	747	Ukraine	433
France	249	India	1153	Germany	423	Brazil	64	Ghana	489	European Union	176
United Kingdom	240	Canada	1061	Canada	410	European Union	32	Malaysia	291	Sweden	147
Netherlands	174	Belarus	915	Mexico	391	United Kingdom	30	Thailand	286	France	117
Czech Republic	146	Thailand	751	United Kingdom	390	Turkey	28	Egypt	197	United Kingdom	96
Republic of Korea	139	Turkey	643	India	265	Spain	25	Nigeria	153	Poland	94
Canada	135	Mexico	481	Brazil	248	Russian Federation	19	Indonesia	142	Brazil	81

There were no significant differences in countries of origin between the retail group and the rest of the tested applications.

The geographic attack trends reported in the previous WAAR remain relevant, with the majority of requests and attackers originating in the United States, Western European countries, China, and Brazil. For business logic attacks, email extraction is still widely dominated by African countries, such as Senegal, Nigeria, Ghana, and the Ivory Coast. The field of comment spamming shows an unusual proportion toward Eastern European countries, such as Russia, Ukraine, Latvia, and Poland. Besides this geographic dominance, more and more business logic attacks are originating from Asia and South America. Countries generating an increasing number of business logic attacks include Malaysia, Thailand, Pakistan, Mexico, Brazil, and Argentina.

⑤ Conclusions and Recommendations

To mitigate the attacks in this WAAR, organizations should:

- › Deploy security solutions that prevent automated attacks. To stop automated attacks, security solutions should recognize known automated sources, differentiate between bots and human clients, and 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 experience about twice as many SQL injection attacks, and fewer RFI attacks than the general application population. Moreover, the SQL injection attacks experienced by online retail applications were much more intensive.
- › Detect and block attacks that target known vulnerabilities. The knowledgebase of exploitable weaknesses in an application must be frequently updated.
- › 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 to be effective.
- › Participate in a security community and share threat intelligence. The increased automation and scale of attacks leave a large footprint that can only be seen by analyzing data gathered from a large set of potential victims.
- › Attack distribution is burst-orientated 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'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.

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 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 are contained within it. These emails addresses are then used for promotional campaigns and similar marketing purposes. Email extraction is one of several activities that harvest data from web applications without the consent of data owners and the application administrators.

6.7 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 this 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.

Hacker Intelligence Initiative Overview

Web Application Attack Reports (WAARs) are part of the Imperva Hacker Intelligence Initiative (HII). This initiative goes inside the cyber-underground and provides analysis of the trending hacking techniques and interesting attack campaigns from the past month. A part of Imperva's Application Defense Center research arm, the HII, is focused on tracking the latest trends in attacks, Web application security and cyber-crime business models with the goal of improving security controls and risk management processes.

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

© Copyright 2013, Imperva

All rights reserved. Imperva and SecureSphere are registered trademarks of Imperva.

All other brand or product names are trademarks or registered trademarks of their respective holders. #HII-SA-SECURITY-SUMMARY#4-0713rev1

