www.webappsec.org
contact@webappsec.org

WEB APPLICATION SECURITY STATISTICS 2008

CONTENTS

1. INTRODUCTION

Web Application Security Consortium (WASC) presents web application vulnerability statistics for 2008 (WASC Web Application Security Statistics Project 2008) and expresses gratitude to the experts and the companies that have contributed to the development of the project:

Sergey Gordeychik* (POSITIVE TECHNOLOGIES)
Jeremiah Grossman (WHITEHAT SECURITY)
Mandeep Khera (CENZIC)
Matt Lantinga (HP APPLICATION SECURITY CENTER)
Chris Wysopal (VERACODE)
Chris Eng (VERACODE)
Shreeraj Shah (BLUEINFY)
Lawson Lee (dns)
Campbell Murray (ENCRIPTION LIMITED)
Dmitry Evteev (POSITIVE TECHNOLOGIES)
**Project Leader*

2. PROJECT GOALS

The Web Application Security Consortium (WASC) is pleased to announce the WASC Web Application Security Statistics Project 2008. This initiative is a collaborative industry wide effort to pool together sanitized website vulnerability data and to gain a better understanding about the web application vulnerability landscape. We ascertain which classes of attacks are the most prevalent regardless of the methodology used to identify them. Industry statistics such as those compiled by Mitre CVE project provide valuable insight into the types of vulnerabilities discovered in open source and commercial applications, this project tries to be the equivalent for custom web applications.

The main Project goals are:

- Identify the prevalence and probability of different vulnerability classes
- Compare testing methodologies against what types of vulnerabilities they are likely to identify

3. METHODOLOGY

This article contains Web application vulnerability statistics which was collected during penetration testing, security audits and other activities made by companies which were members of WASC in 2008. The statistics includes data about 12186 sites with 97554 detected vulnerabilities.

As a result, we now have 4 data sets:

- Overall statistics by all kinds of activities;
- Automatic scanning statistics;
- Black box method security assessment statistics;
- White box method security assessment statistics.

Automatic scanning data is collected in fully automated scanning process without any preliminary settings (with standard profile) of hosting provider sites. Remember that not all the sites include interactive elements, and additional settings made by an expert considering certain Web application, allows to greatly improve the efficiency of vulnerability detection.

Black box method security assessment statistics includes the results of manual and automated Web application analysis without any preliminary known data about the application. As a rule, this includes scanning with standard settings and manual search of vulnerabilities unavailable for automatic scanners.

White box method security assessment statistics includes the results of the deep Web application analysis which contains application analysis done as an authorized user. It also includes static source code and binary analysis. Detected vulnerabilities are classified according to Web Application Security Consortium Web Security Threat Classification (WASC WSTCv2). Vulnerability risk level is determined by contributors or assessed according to CVSSv2 (Common Vulnerability Scoring System version 2). Then the level was brought to PCI DSS (Payment Card Industry Data Security Standard) risk levels as described in the methodology (see appendix 1).

4. SUMMARY

The statistics includes data about 12186 web applications with 97554 detected vulnerabilities of different risk levels. The analysis shows that **more than 13%**[1] of all reviewed sites can be compromised **completely automatically**. About 49% of web applications contain vulnerabilities of high risk level (Urgent and Critical) detected during automatic scanning (T. 1). However, detailed manual and automated assessment by white box method allows **to detect these high risk level vulnerabilities with probability up to 80-96%.** The probability to detect vulnerabilities with risk level more than medium (PCI DSS compliance level) is more than 86% by any method. At the same time, detailed analysis shows that **99% of web applications are not compliant with PCI DSS standard** (T. 6, P. 13).

The following conclusions can be drawn based on the analysis:

- The most wide spread vulnerabilities are Cross-site Scripting, different types of Information Leakage, SQL Injection, HTTP Response Splitting;
- The probability to detect a urgent or critical error in dynamic web application is about 49% by automatic scanning and 96% by comprehensive expert analysis (white box method);
- Administration issues are 20% more frequent cause of a vulnerability than system development errors;
- 99% of web application are not compliant with PCI DSS standard requirements, and 48% of web applications are not compliant with criteria of ASV scanning by PCI DSS;
- Detailed white box method analysis allows to detect up to 91 vulnerabilities per web application, while automatic scanning – only 3;
- Compared to 2007, the number of sites with wide spread SQL Injection and Cross-site Scripting vulnerabilities fell by 13% and 20%, respectively, however, the number of sites with different types of Information Leakage rose by 24%. On the other hand, the probability to compromise a host automatically rose from 7 to 13 %.

5. DATA ANALYSIS

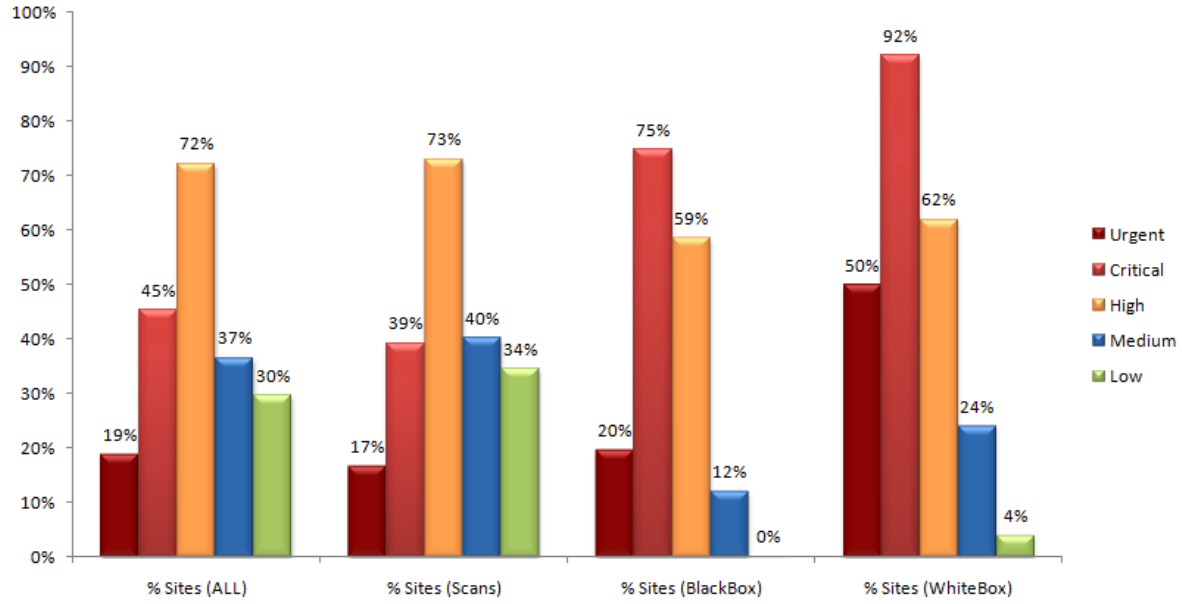
5.1. General analysis

T. 1 and P. 1 show the probability to detect vulnerabilities of different risk levels detected during audits and automatic scanning.

Thus, automatic scanning detected up to 86% sites with one or some vulnerabilities of medium (or higher) risk level (Urgent-High). Black box and white box analysis methods increase it to 92-98%, respectively.

[1] Web applications with Brute Force Attack, Buffer Overflow, OS Commanding, Path Traversal, Remote File Inclusion, SSI Injection, Session Fixation, SQL Injection, Insufficient Authentication, Insufficient Authorization vulnerabilities detected by automatic scannings.

These results are greatly depend on the fact that detailed risk assessment analysis is more adequate and consider not only vulnerability type but its exploitation consequences and application design and implementation. Another important fact is that automatic scanning was made for hosting provider sites which in some cases have no active content, while security assessment is usually done for application with complicated business logic. That is that automatic scanning results can be interpret as typical Internet site scanning results, while black box and white box methods results are scanning results of interactive corporate web applications.

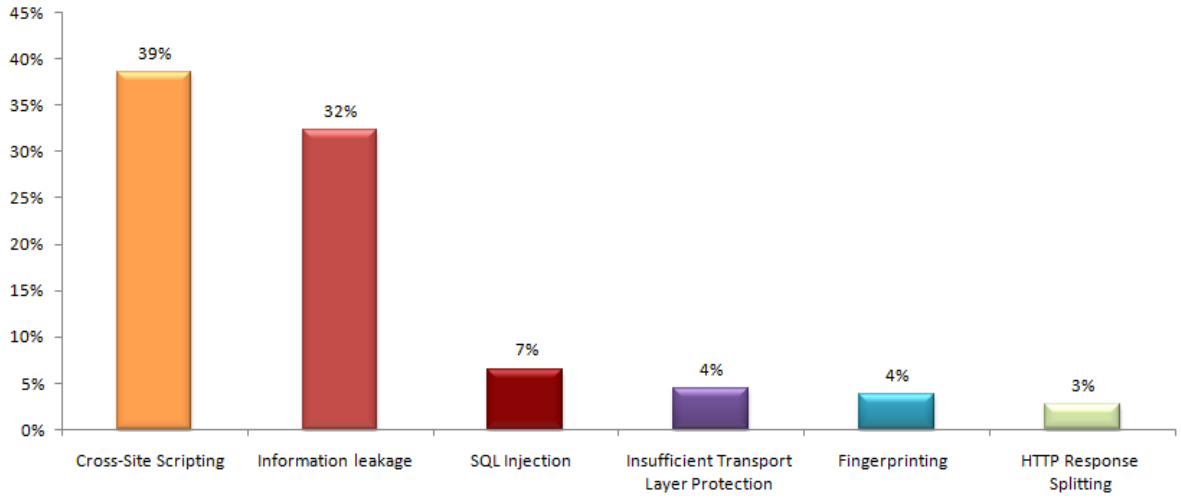


P. 1.The probability to detect vulnerabilities of different risk levels

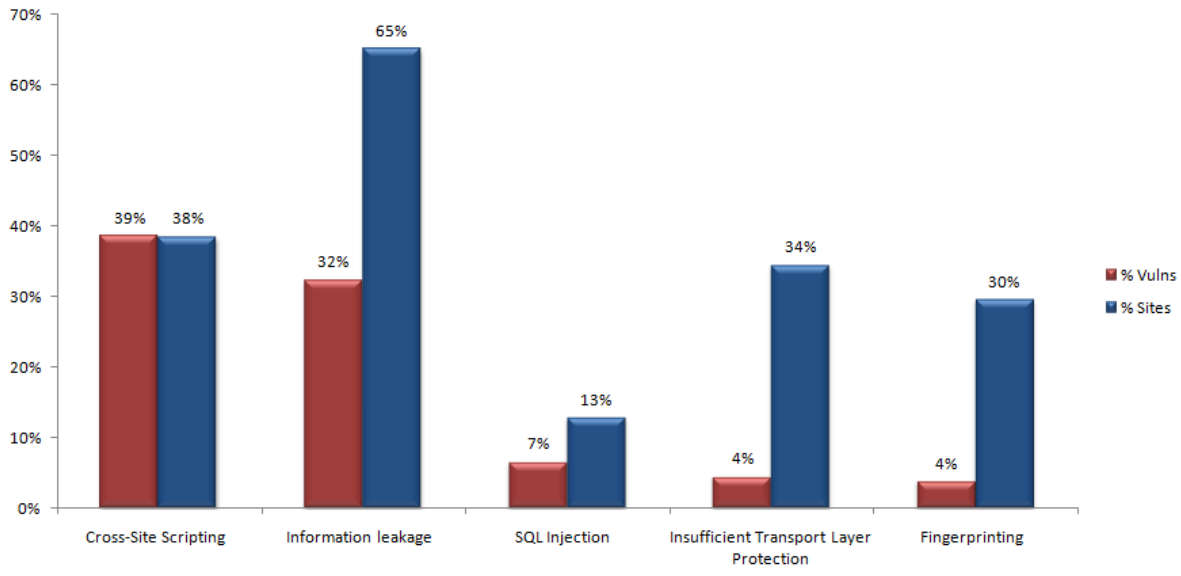
T. 1 The probability to detect vulnerabilities of different risk levels

	ALL	Scans	BlackBox	WhiteBox
Urgent	18.77%	16.70%	19.69%	50.00%
Critical	45.22%	39.25%	74.76%	92.00%
High	72.27%	73.09%	58.51%	62.00%
Medium	36.56%	40.19%	12.05%	24.00%
Low	29.69%	34.45%	0.10%	4.00%
U+C	55,50%	49,40%	79,73%	96,00%
U+C+H	87,66%	86,38%	91,59%	98,48%

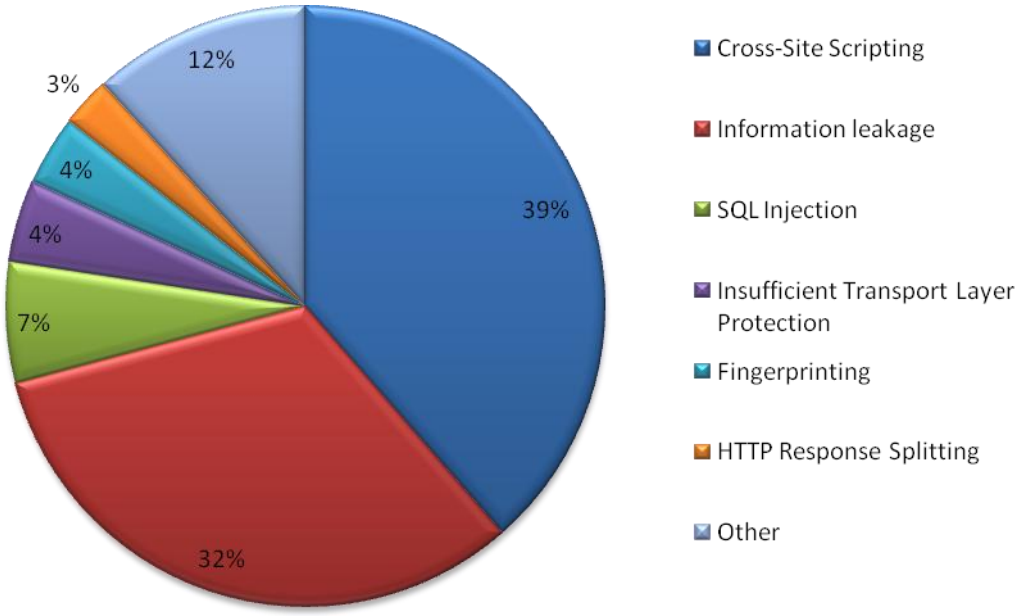
The most widespread vulnerabilities are Cross-Site Scripting, Information Leakage, SQL Injection, Insufficient Transport Layer Protection, Fingerprinting и HTTP Response Splitting (P. 2). As a rule, Cross-Site Scripting, SQL Injection and HTTP Response Splitting vulnerabilities are caused by design errors, while Information Leakage, Insufficient Transport Layer Protection and Fingerprinting are often caused by insufficient administration (e.g., access control).



P. 2. The most widespread vulnerabilities in web applications (% Vulns ALL)



P. 3. The probability to detect the most widespread vulnerabilities in web applications (% Sites ALL)

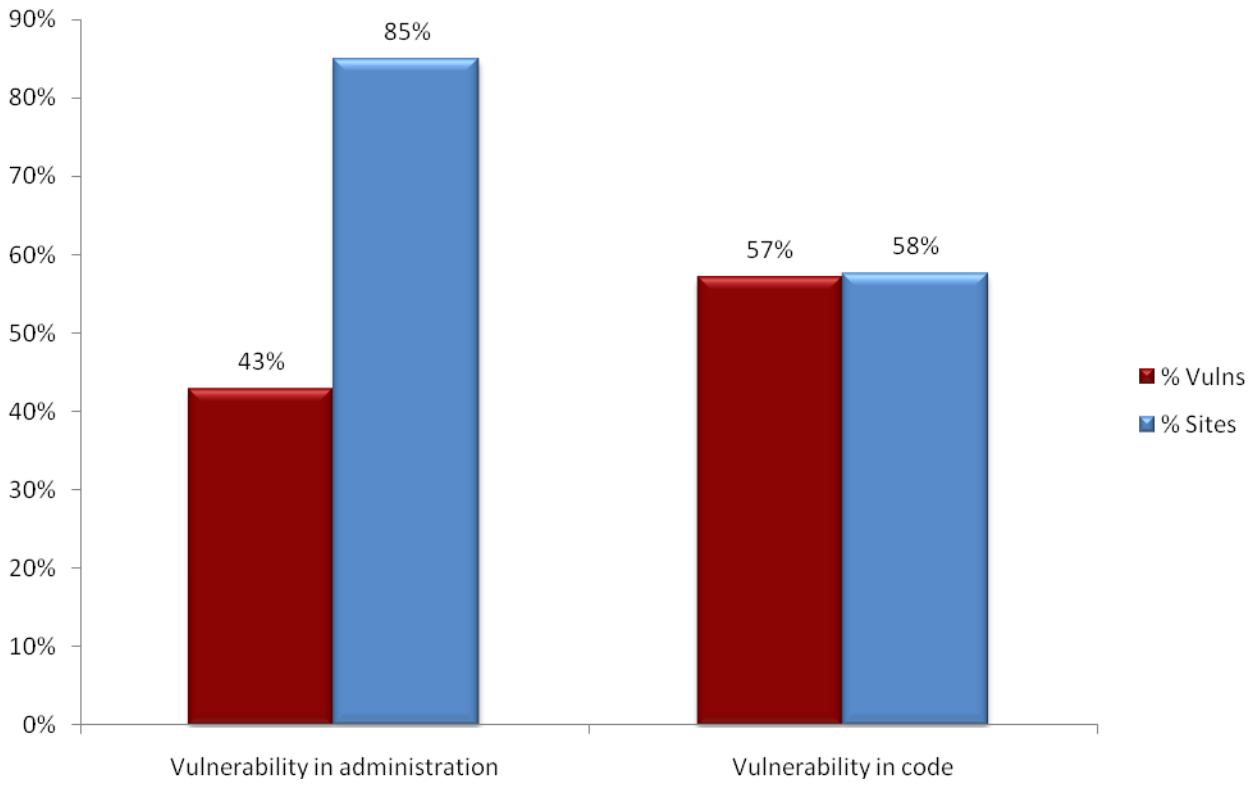


P. 4. Percent of vulnerabilities out of total number of vulnerabilities (% Vulns ALL)

If we consider vulnerability origin as a whole (according to classification in Appendix 2) we'll see that vulnerabilities caused by insufficient administration are 20% more frequent (P. 5). At the same time, there are up to 4 issues per site caused by administration flaws and up to 8 vulnerabilities caused by design errors (T. 2).

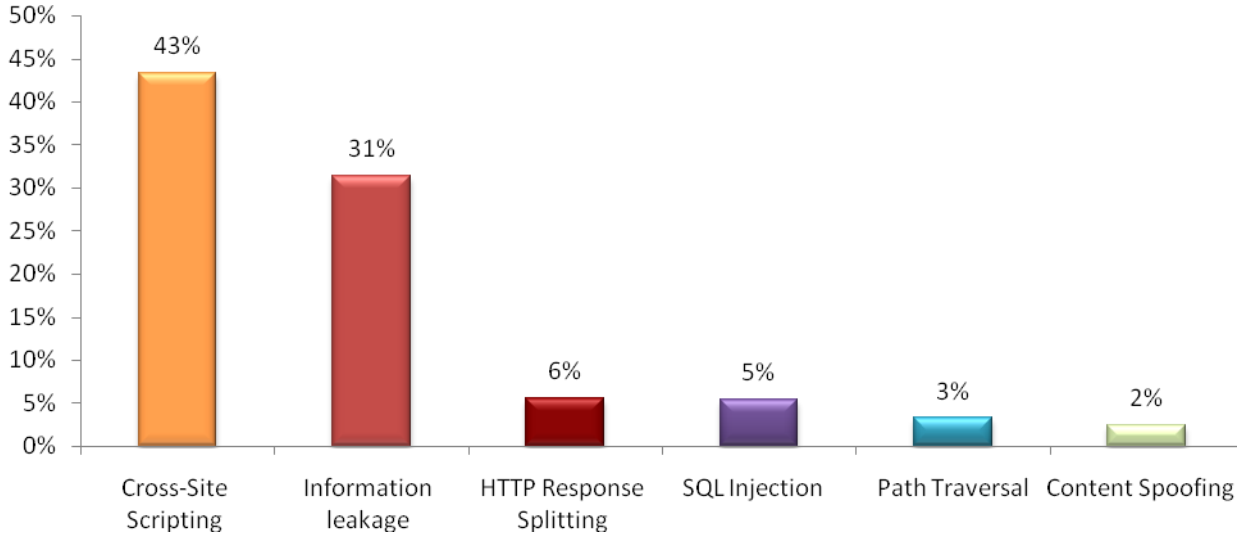
T. 2 The probability to detect vulnerabilities depending on vulnerability origin

	No. of Vulns	No. of Sites	% Vulns	% Sites	No. Vulns on Site
Vulnerability in administration	41859	10347	42.91%	84.91%	4.05
Vulnerability in code	55695	7023	57.09%	57.63%	7.93

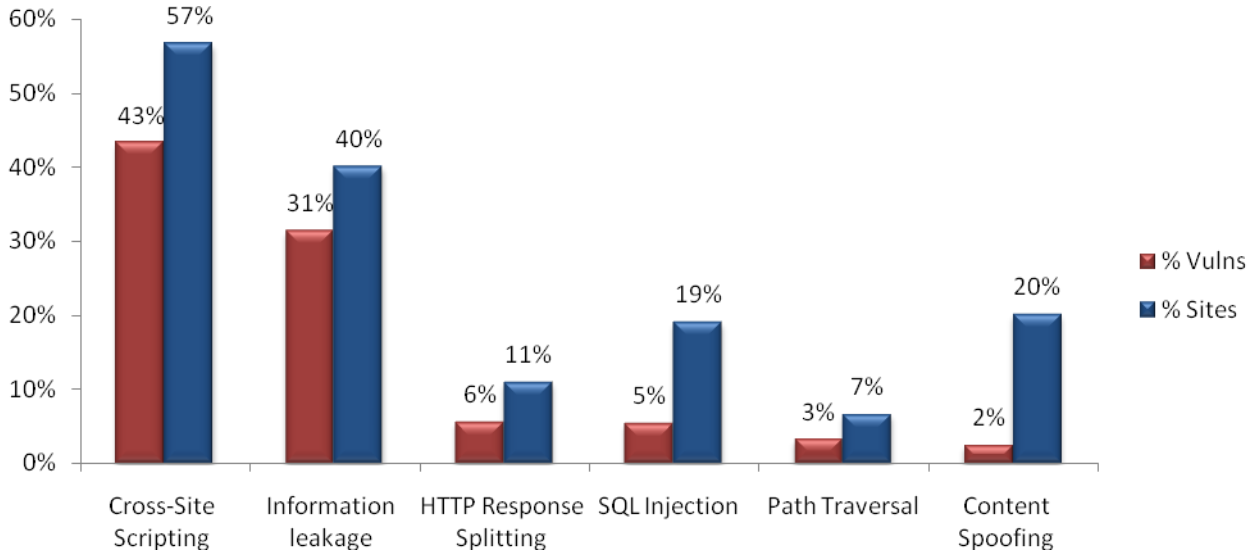


P. 5. The probability to detect vulnerabilities depending on their origin

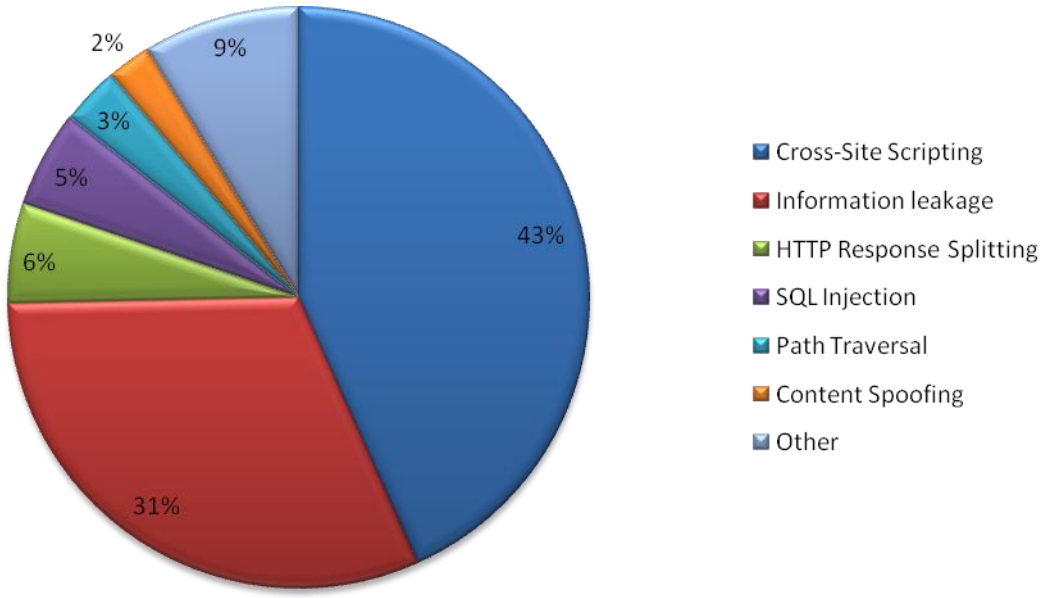
Detailed web application analysis by black box and white box methods shows that appreciable percent of sites are vulnerable to Content Spoofing and Path Traversal (P. 6), and the probability to detect a vulnerability of SQL Injection type reaches 19% in this approach (P. 7).



P. 6. The most widespread vulnerabilities in web applications (% Vulns BlackBox & WhiteBox)

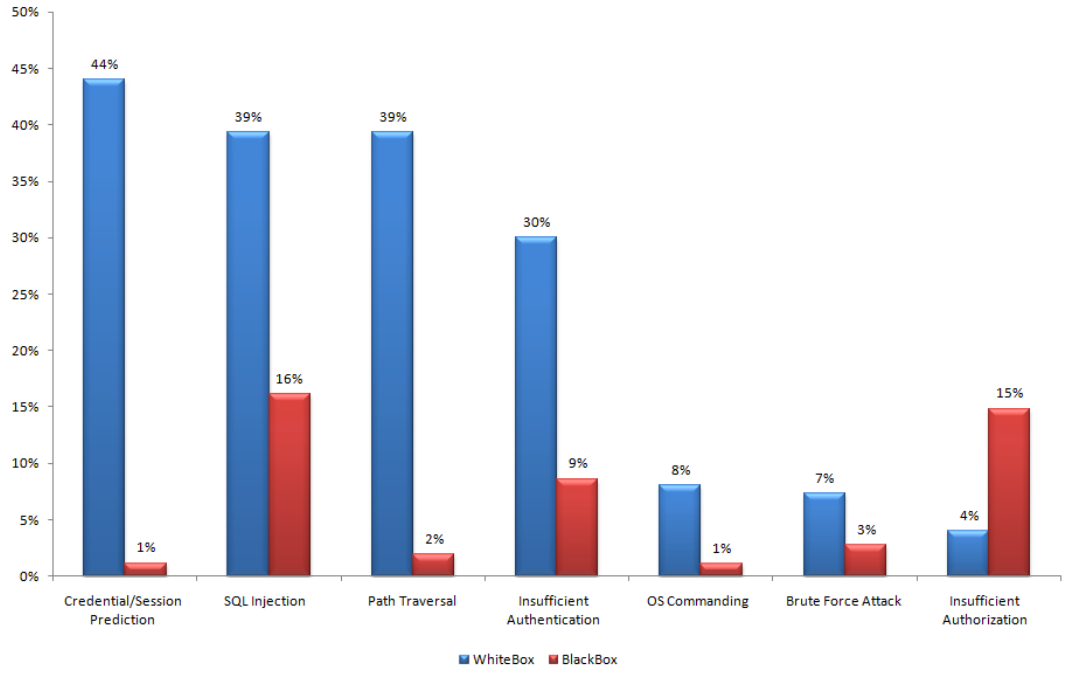


P. 7.The probability to detect the most widespread vulnerabilities in web applications (% Sites BlackBox & WhiteBox)



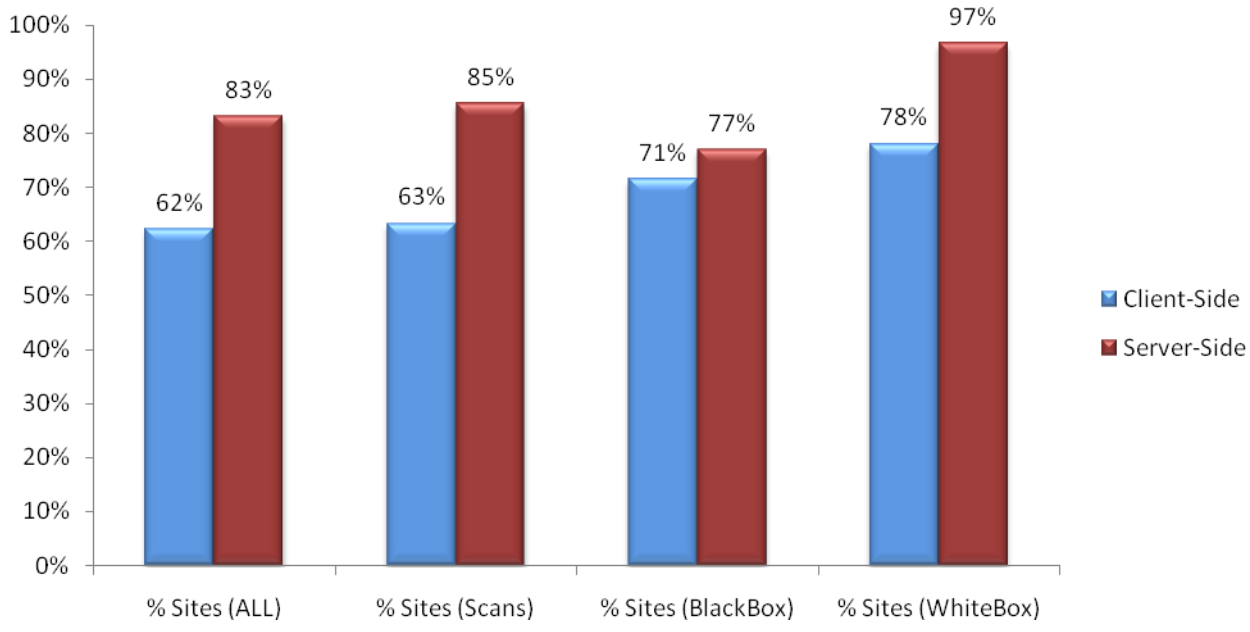
P. 8. Percent of vulnerabilities out of total number of vulnerabilities (% Vulns BlackBox & WhiteBox)

If we consider the prevalence of high risk level vulnerabilities in detailed web application analysis (P. 9) we'll see that the most widespread is Credential/Session Prediction errors. SQL Injection, Path Traversal and implementation and configuration errors in authentication and authorization systems are also widespread.

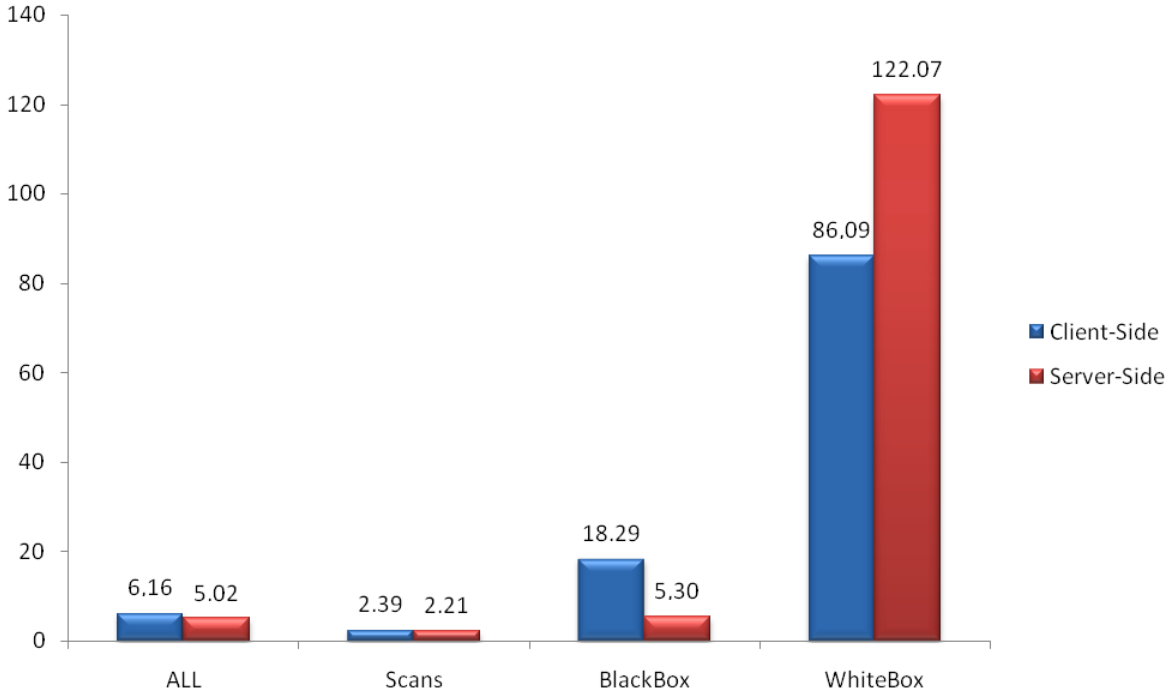


P. 9. The probability to detect the most risky vulnerabilities in Web applications (% Sites BlackBox & WhiteBox)

If we consider the probability to detect vulnerabilities in terms of web resource visitors and web server impact (according to classification in appendix 2), the server-side vulnerabilities are the most widespread (P. 10). But the vulnerability distribution by impact type per site is irregular and greatly depends on used vulnerability search method (P. 11).



P. 10. The probability to detect vulnerability by impact type



P. 11.Vulnerabilities per site by different search methods (No. Vulns on Site)

T. 3 Vulnerabilities by impact

	No. of Vulns	No. of Sites	% Vulns	% Sites	No. Vulns on Site
ALL Stat (Server-Side)	50856	10125	52.13%	83.09%	5.02
ALL Stat (Client-Side)	46698	7580	47.87%	62.20%	6.16
Scans (Server-Side)	19746	8922	55.60%	85.40%	2.21
Scans (Client-Side)	15767	6607	44.40%	63.24%	2.39
BlackBox (Server-Side)	4260	804	23.77%	76.86%	5.30
BlackBox (Client-Side)	13665	747	76.23%	71.41%	18.29
WhiteBox (Server-Side)	17700	145	63.73%	96.67%	122.07
WhiteBox (Client-Side)	10072	117	36.27%	78.00%	86.09

5.2. Data analysis according to PCI DSS requirements

If we consider data sets about vulnerable Web applications according to PCI DSS requirements, we can easily sort (T. 4) those that are about certain vulnerability elimination in Web applications. In addition, PCI DSS Technical and Operational Requirements for Approved Scanning Vendors (ASVs) includes similar requirements but affects only ASV scanning by PCI (T. 5).

T. 4 PCI DSS requirements for Web application security

PCI DSS v.1.2 requirements	Procedure
6.5 Develop all web applications (internal and external, and including web administrative access to application) based on secure coding guidelines such as the Open Web Application Security Project Guide. Cover prevention of common coding vulnerabilities in software development processes, to include the following:	-
6.5.1 Cross-site scripting (XSS)	6.5.1 Cross-site scripting (XSS) (Validate all parameters before inclusion.)
6.5.2 Injection flaws, particularly SQL injection. Also consider LDAP and Xpath injection flaws as well as other injection flaws.	6.5.2 Injection flaws, particularly SQL injection (Validate input to verify user data cannot modify meaning of commands and queries.)
6.5.3 Malicious file execution	6.5.3 Malicious file execution (Validate input to verify application does not accept filenames or files from users.)
6.5.5 Cross-site request forgery (CSRF)	6.5.5 Cross-site request forgery (CSRF) (Do not reply on authorization credentials and tokens automatically submitted by browsers.)
6.5.6 Information leakage and improper error handling	6.5.6 Information leakage and improper error handling (Do not leak information via error messages or other means.)
6.5.7 Broken authentication and session management	6.5.7 Broken authentication and session management (Properly authenticate users and protect account credentials and session tokens.)
6.5.9 Insecure communications	6.5.9 Insecure communications (Properly encrypt all authenticated and sensitive communications.)

PCI DSS v.1.2 requirements	Procedure
6.6 For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods: - Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes - Installing a web-application firewall in front of public-facing web applications	-

T. 5 PCI DSS Technical and Operational Requirements for Approved Scanning Vendors (ASVs) for WEB

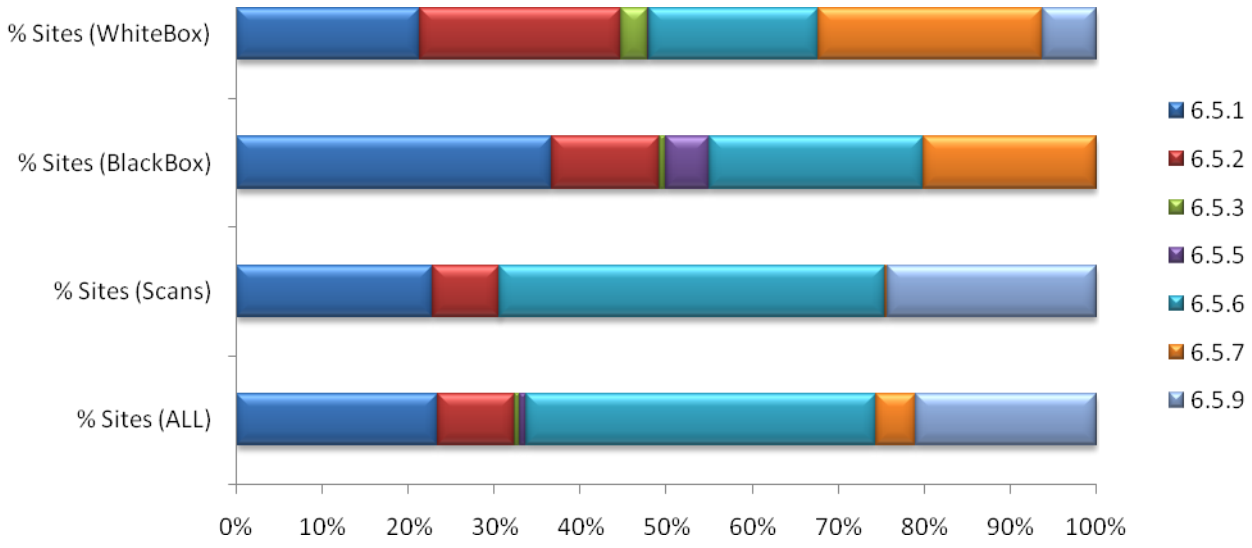
Technical and Operational Requirements for Approved Scanning Vendors (ASVs) v.1.1	Procedure
Web Server Check	The ASV scanning solution must be able to test for all known vulnerabilities and configuration issues on web servers. New exploits are routinely discovered in web server products. The ASV scanning solution must be able to detect and report known exploits. Browsing of directories on a web server is not a good practice. The ASV scanning solution must be able to scan the web site and verify that directory browsing is not possible on the server. The ASV scanning solution must be able to detect all known CGI vulnerabilities.
Custom Web Application Check	The ASV scanning solution must be able to detect the following application vulnerabilities and configuration issues: • Unvalidated parameters which lead to SQL injection attacks • Cross-site scripting (XSS) flaws

Assessing collected data statistics by criteria from T. 4 and T. 5, we conclude the following (see T. 6 and P. 12 – 14).

T. 6 % of sites which are not complaint to PCI DSS requirements in Web application scanning methods

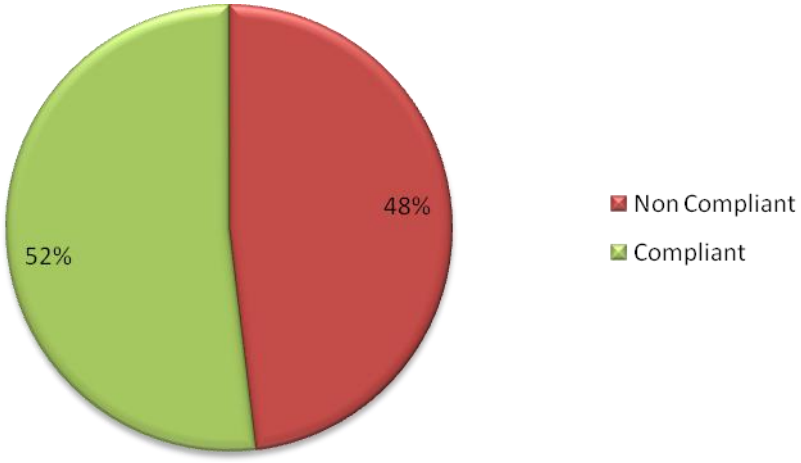
PCI DSS v.1.2 requirement	Non compliant. ALL (% Sites)	Non compliant. Scans (% Sites)	Non compliant. BlackBox (% Sites)	Non compliant. WhiteBox (% Sites)
6.5.1 Cross-site scripting (XSS)	38.45%	37.66%	56.41%	58.67%
6.5.2 Injection flaws. particularly SQL injection. Also consider LDAP and Xpath injection flaws as well as other injection flaws.	14.55%	12.70%	19.31%	64.00%
6.5.3 Malicious file execution	0.94%	0.08%	1.05%	8.67%
6.5.5 Cross-site request forgery (CSRF)	1.32%	0.02%	7.93%	0.67%
6.5.6 Information leakage and improper error handling	66.67%	74.05%	38.24%	54.00%
6.5.7 Broken authentication and session management	7.62%	0.52%	30.98%	71.33%
6.5.9 Insecure communications	34.42%	39.96%	0.00%*	17.33%
Technical and Operational Requirements for Approved Scanning Vendors (ASVs) v.1.1				
Web Server Check	Inapplicable	5.73%	Inapplicable	Inapplicable
Custom Web Application Check	Inapplicable	44.92%	Inapplicable	Inapplicable

* Vulnerability of this class are not incuded into reports during web application security assessment by black box method.



P 12. The distribution of sites non compliant to PCI DSS

P 13.Compliance level of Web application to PCI DSS (QSA) requirements



P 14. Compliance level of Web application to PCI DSS (ASV) requirements

Thus, more than 48 % of scanned Web applications are not compliant to PCI DSS requirements by ASV scanning. Meanwhile, deeper analysis shows that 99% of Web applications are not complaint to the standard requirements.

6. PARTICIPATION

If you represent an organization that performs vulnerability assessments on websites, particular in those in custom web applications, through a manual or automated process and would like to participate please let us know. Once statistics are compiled, a report will be distributed, and all contributors will receive a logo on the project pages as well as on other deliverables in appreciate of their contribution. Please contact Sergey Gordeychik (gordey@ptsecurity.ru).

7. APPENDIX 1: RISK ASSESSMENT METHODOLOGY

T. 8 Risk level assessment routine

Threat Classification	Basic CVSS Score	PCI DSS Risk
Abuse of Functionality	4 (AV:N/AC:H/Au:N/C:P/I:P/A:N)	Medium
Brute Force Attack	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical
Buffer Overflow	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Content Spoofing	5 (AV:N/AC:L/Au:N/C:N/I:P/A:N)	High
Credential/Session Prediction	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical
Cross-Site Scripting	6.4 (AV:N/AC:L/Au:N/C:P/I:P/A:N)	Critical
Cross-Site Request Forgery	5 (AV:N/AC:L/Au:N/C:N/I:P/A:N)	High
Denial of Service	7.8 (AV:N/AC:L/Au:N/C:N/I:N/A:C)	High
Format String Attack	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
HTTP Request Splitting	6.4 (AV:N/AC:L/Au:N/C:P/I:P/A:N)	Critical
HTTP Response Splitting	6.4 (AV:N/AC:L/Au:N/C:P/I:P/A:N)	Critical
HTTP Request Smuggling	6.4 (AV:N/AC:L/Au:N/C:P/I:P/A:N)	Critical
HTTP Response Smuggling	6.4 (AV:N/AC:L/Au:N/C:P/I:P/A:N)	Critical
Integer Overflow	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
LDAP Injection	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Mail Command Injection	5 (AV:N/AC:L/Au:N/C:N/I:P/A:N)	High
OS Commanding	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Path Traversal	7.8 (AV:N/AC:L/Au:N/C:C/I:N/A:N)	Critical
Predictable Resource Location	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High

Remote File Inclusion	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Routing Detour	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
SOAP Array Abuse	7.8 (AV:N/AC:L/Au:N/C:N/I:N/A:C)	High
SSI Injection	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Session Fixation	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical
SQL Injection	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
URL Redirectors	2.6 (AV:N/AC:H/Au:N/C:N/I:P/A:N)	Medium
XPath Injection	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
XML Attribute Blowup	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
XML External Entity	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
XML Entity Expansion	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
XML Injection	7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)	Critical
XQuery Injection	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Application Misconfiguration	5.1 (AV:N/AC:H/Au:N/C:P/I:P/A:P)	Medium
Directory Indexing	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
Fingerprinting	0 (AV:N/AC:L/Au:N/C:N/I:N/A:N)	Low
Improper Parsing	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Improper Permissions	10 (AV:N/AC:L/Au:N/C:C/I:C/A:C)	Urgent
Information leakage	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
Insecure Indexing	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
Insufficient Anti-automation	4 (AV:N/AC:H/Au:N/C:P/I:P/A:N)	Medium
Insufficient Authentication	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical
Insufficient Authorization	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical

Insufficient Data Protection	5 (AV:N/AC:L/Au:N/C:P/I:N/A:N)	High
Insufficient Process Validation	4 (AV:N/AC:H/Au:N/C:P/I:P/A:N)	Medium
Insufficient Session Expiration	6.8 (AV:N/AC:M/Au:N/C:P/I:P/A:P)	Critical
Insufficient Transport Layer Protection	4 (AV:N/AC:H/Au:N/C:P/I:P/A:N)	Medium
Server Misconfiguration	5.1 (AV:N/AC:H/Au:N/C:P/I:P/A:P)	Medium

8. APPENDIX 2: ADDITIONAL VULNERABILITY CLASSIFICATION

T. 9 Vulnerability classification by origin and impact

Threat Classification	Vulnerability in	Impact
Abuse of Functionality	code	server-side
Brute Force Attack	administration	server-side
Buffer Overflow	code	server-side
Content Spoofing	code	client-side
Credential/Session Prediction	code	server-side
Cross-Site Scripting	code	client-side
Cross-Site Request Forgery	code	client-side
Denial of Service	administration	server-side
Format String Attack	code	server-side
HTTP Request Splitting	code	client-side
HTTP Response Splitting	code	client-side
HTTP Request Smuggling	administration	client-side
HTTP Response Smuggling	administration	client-side

Integer Overflow	code	server-side
LDAP Injection	code	server-side
Mail Command Injection	code	server-side
OS Commanding	code	server-side
Path Traversal	code	server-side
Predictable Resource Location	administration	server-side
Remote File Inclusion	code	server-side
Routing Detour	code	server-side
SOAP Array Abuse	code	server-side
SSI Injection	code	server-side
Session Fixation	code	server-side
SQL Injection	code	server-side
URL Redirectors	code	client-side
XPath Injection	code	server-side
XML Attribute Blowup	code	server-side
XML External Entity	code	server-side
XML Entity Expansion	code	server-side
XML Injection	code	server-side
XQuery Injection	code	server-side
Application Misconfiguration	administration	server-side
Directory Indexing	administration	server-side
Fingerprinting	administration	server-side
Improper Parsing	code	server-side

Improper Permissions	administration	server-side
Information leakage	administration	server-side
Insecure Indexing	administration	server-side
Insufficient Anti-automation	code	server-side
Insufficient Authentication	code	server-side
Insufficient Authorization	code	server-side
Insufficient Data Protection	administration	server-side
Insufficient Process Validation	code	server-side
Insufficient Session Expiration	code	server-side
Insufficient Transport Layer Protection	administration	client-side
Server Misconfiguration	administration	server-side

9. APPENDIX 3: STATISTICS

Overall Data

T. 10 General statistics Threat Classification

Threat Classification	N of Vulns	N of Sites	% Vulns	% Sites
Abuse of Functionality	153	83	0.16%	0.68%
Brute Force Attack	79	51	0.08%	0.42%
Buffer Overflow	537	84	0.55%	0.69%
Content Spoofing	1564	304	1.60%	2.49%
Credential/Session Prediction	794	147	0.81%	1.21%

Cross-Site Scripting	37624	4686	38.57%	38.45%
Cross-Site Request Forgery	285	161	0.29%	1.32%
Denial of Service	42	36	0.04%	0.30%
Format String Attack	52	43	0.05%	0.35%
HTTP Request Splitting	311	162	0.32%	1.33%
HTTP Response Splitting	2592	161	2.66%	1.32%
HTTP Request Smuggling	0	0	0.00%	0.00%
HTTP Response Smuggling	0	0	0.00%	0.00%
Integer Overflow	79	46	0.08%	0.38%
LDAP Injection	41	16	0.04%	0.13%
Mail Command Injection	1	1	0.00%	0.01%
OS Commanding	76	30	0.08%	0.25%
Path Traversal	1563	139	1.60%	1.14%
Predictable Resource Location	1507	295	1.54%	2.42%
Remote File Inclusion	99	44	0.10%	0.36%
Routing Detour	0	0	0.00%	0.00%
SOAP Array Abuse	2	1	0.00%	0.01%
SSI Injection	157	33	0.16%	0.27%
Session Fixation	137	123	0.14%	1.01%
SQL Injection	6345	1555	6.50%	12.76%
URL Redirectors	5	4	0.01%	0.03%

XPath Injection	64	19	0.07%	0.16%
XML Attribute Blowup	0	0	0.00%	0.00%
XML External Entity	0	0	0.00%	0.00%
XML Entity Expansion	0	0	0.00%	0.00%
XML Injection	0	0	0.00%	0.00%
XQuery Injection	0	0	0.00%	0.00%
Application Misconfiguration	85	60	0.09%	0.49%
Directory Indexing	370	184	0.38%	1.51%
Fingerprinting	3663	3604	3.75%	29.57%
Improper Parsing	1464	524	1.50%	4.30%
Improper Permissions	4	4	0.00%	0.03%
Information leakage	31527	7942	32.32%	65.17%
Insecure Indexing	8	7	0.01%	0.06%
Insufficient Anti-automation	108	36	0.11%	0.30%
Insufficient Authentication	806	304	0.83%	2.49%
Insufficient Authorization	615	286	0.63%	2.35%
Insufficient Data Protection	64	21	0.07%	0.17%
Insufficient Process Validation	52	34	0.05%	0.28%
Insufficient Session Expiration	169	71	0.17%	0.58%
Insufficient Transport Layer Protection	4317	4195	4.43%	34.42%
Server Misconfiguration	193	113	0.20%	0.93%

Total	97554	12186

T. 11 Vulnerabilities distribution by risk Threat rank

Threat rank	N of Vulns	N of Sites	N of Sites	% Sites
Urgent	8918	2287	9.14%	18.77%
Critical	44669	5511	45.79%	45.22%
High	35375	8807	36.26%	72.27%
Medium	4908	4455	5.03%	36.56%
Low	3663	3618	3.75%	29.69%

Automatic scans

T. 12 General statistics Threat Classification

Threat Classification	N of Vulns	N of Sites	% Vulns	% Sites
Abuse of Functionality	1	1	0.00%	0.01%
Brute Force Attack	5	5	0.01%	0.05%
Buffer Overflow	6	3	0.02%	0.03%
Content Spoofing	29	22	0.08%	0.21%
Credential/Session Prediction	9	9	0.03%	0.09%
Cross-Site Scripting	11230	3934	31.62%	37.66%
Cross-Site Request Forgery	2	2	0.01%	0.02%
Denial of Service	30	25	0.08%	0.24%
Format String Attack	0	0	0.00%	0.00%
HTTP Request Splitting	311	162	0.88%	1.55%

HTTP Response Splitting	0	0	0.00%	0.00%
HTTP Request Smuggling	0	0	0.00%	0.00%
HTTP Response Smuggling	0	0	0.00%	0.00%
Integer Overflow	0	0	0.00%	0.00%
LDAP Injection	0	0	0.00%	0.00%
Mail Command Injection	0	0	0.00%	0.00%
OS Commanding	28	5	0.08%	0.05%
Path Traversal	82	56	0.23%	0.54%
Predictable Resource Location	16	15	0.05%	0.14%
Remote File Inclusion	86	36	0.24%	0.34%
Routing Detour	0	0	0.00%	0.00%
SOAP Array Abuse	0	0	0.00%	0.00%
SSI Injection	157	33	0.44%	0.32%
Session Fixation	3	3	0.01%	0.03%
SQL Injection	2969	1217	8.36%	11.65%
URL Redirectors	1	1	0.00%	0.01%
XPath Injection	0	0	0.00%	0.00%
XML Attribute Blowup	0	0	0.00%	0.00%
XML External Entity	0	0	0.00%	0.00%
XML Entity Expansion	0	0	0.00%	0.00%
XML Injection	0	0	0.00%	0.00%
XQuery Injection	0	0	0.00%	0.00%
Application Misconfiguration	48	37	0.14%	0.35%

Directory Indexing	12	11	0.03%	0.11%
Fingerprinting	3604	3587	10.15%	34.34%
Improper Parsing	1463	523	4.12%	5.01%
Improper Permissions	2	2	0.01%	0.02%
Information leakage	11134	7593	31.35%	72.68%
Insecure Indexing	8	7	0.02%	0.07%
Insufficient Anti-automation	0	0	0.00%	0.00%
Insufficient Authentication	24	15	0.07%	0.14%
Insufficient Authorization	14	14	0.04%	0.13%
Insufficient Data Protection	10	10	0.03%	0.10%
Insufficient Process Validation	12	11	0.03%	0.11%
Insufficient Session Expiration	1	1	0.00%	0.01%
Insufficient Transport Layer Protection	4194	4175	11.81%	39.96%
Server Misconfiguration	22	22	0.06%	0.21%
Total	35513	10447		

T. 13 Vulnerabilities distribution by risk Threat rank

Threat rank	N of Vulns	N of Sites	N of Sites	% Sites
Urgent	4711	1745	13.27%	16.70%
Critical	11679	4100	32.89%	39.25%
High	11257	7636	31.70%	73.09%
Medium	4294	4199	12.09%	40.19%
Low	3625	3599	10.21%	34.45%

Black Box

T. 14 General statistics Threat Classification

Threat Classification	N of Vulns	N of Sites	% Vulns	% Sites
Abuse of Functionality	135	75	0.75%	7.17%
Brute Force Attack	34	29	0.19%	2.77%
Buffer Overflow	0	0	0.00%	0.00%
Content Spoofing	1110	241	6.19%	23.04%
Credential/Session Prediction	15	12	0.08%	1.15%
Cross-Site Scripting	11768	590	65.65%	56.41%
Cross-Site Request Forgery	185	83	1.03%	7.93%
Denial of Service	9	8	0.05%	0.76%
Format String Attack	2	2	0.01%	0.19%
HTTP Request Splitting	0	0	0.00%	0.00%
HTTP Response Splitting	601	77	3.35%	7.36%
HTTP Request Smuggling	0	0	0.00%	0.00%
HTTP Response Smuggling	0	0	0.00%	0.00%
Integer Overflow	9	6	0.05%	0.57%
LDAP Injection	0	0	0.00%	0.00%
Mail Command Injection	0	0	0.00%	0.00%
OS Commanding	16	11	0.09%	1.05%
Path Traversal	29	20	0.16%	1.91%
Predictable Resource Location	855	155	4.77%	14.82%
Remote File Inclusion	3	3	0.02%	0.29%

Routing Detour	0	0	0.00%	0.00%
SOAP Array Abuse	0	0	0.00%	0.00%
SSI Injection	0	0	0.00%	0.00%
Session Fixation	83	79	0.46%	7.55%
SQL Injection	1556	169	8.68%	16.16%
URL Redirectors	1	1	0.01%	0.10%
XPath Injection	59	17	0.33%	1.63%
XML Attribute Blowup	0	0	0.00%	0.00%
XML External Entity	0	0	0.00%	0.00%
XML Entity Expansion	0	0	0.00%	0.00%
XML Injection	0	0	0.00%	0.00%
XQuery Injection	0	0	0.00%	0.00%
Application Misconfiguration	31	20	0.17%	1.91%
Directory Indexing	104	42	0.58%	4.02%
Fingerprinting	1	1	0.01%	0.10%
Improper Parsing	1	1	0.01%	0.10%
Improper Permissions	2	2	0.01%	0.19%
Information leakage	745	399	4.16%	38.15%
Insecure Indexing	0	0	0.00%	0.00%
Insufficient Anti-automation	6	4	0.03%	0.38%
Insufficient Authentication	158	90	0.88%	8.60%
Insufficient Authorization	312	155	1.74%	14.82%
Insufficient Data Protection	2	2	0.01%	0.19%

Insufficient Process Validation	5	5	0.03%	0.48%
Insufficient Session Expiration	30	27	0.17%	2.58%
Insufficient Transport Layer Protection	0	0	0.00%	0.00%
Server Misconfiguration	58	38	0.32%	3.63%
Total	17925	1046		

T. 15 Vulnerabilities distribution by risk Threat rank

Threat rank	N of Vulns	N of Sites	N of Sites	% Sites
Urgent	1648	206	9.19%	19.69%
Critical	13030	782	72.69%	74.76%
High	3011	612	16.80%	58.51%
Medium	235	126	1.31%	12.05%
Low	1	1	0.01%	0.10%

White Box

T. 16 General statistics Threat Classification

Threat Classification	N of Vulns	N of Sites	% Vulns	% Sites
Abuse of Functionality	7	4	0.03%	2.67%
Brute Force Attack	15	11	0.05%	7.33%
Buffer Overflow	421	1	1.52%	0.67%
Content Spoofing	0	0	0.00%	0.00%
Credential/Session Prediction	695	66	2.50%	44.00%
Cross-Site Scripting	8006	88	28.83%	58.67%
Cross-Site Request Forgery	2	1	0.01%	0.67%
Denial of Service	3	3	0.01%	2.00%
Format String Attack	2	1	0.01%	0.67%

HTTP Request Splitting	0	0	0.00%	0.00%
HTTP Response Splitting	1941	54	6.99%	36.00%
HTTP Request Smuggling	0	0	0.00%	0.00%
HTTP Response Smuggling	0	0	0.00%	0.00%
Integer Overflow	0	0	0.00%	0.00%
LDAP Injection	0	0	0.00%	0.00%
Mail Command Injection	1	1	0.00%	0.67%
OS Commanding	29	12	0.10%	8.00%
Path Traversal	1450	59	5.22%	39.33%
Predictable Resource Location	15	13	0.05%	8.67%
Remote File Inclusion	3	2	0.01%	1.33%
Routing Detour	0	0	0.00%	0.00%
SOAP Array Abuse	0	0	0.00%	0.00%
SSI Injection	0	0	0.00%	0.00%
Session Fixation	1	1	0.00%	0.67%
SQL Injection	898	59	3.23%	39.33%
URL Redirectors	0	0	0.00%	0.00%
XPath Injection	0	0	0.00%	0.00%
XML Attribute Blowup	0	0	0.00%	0.00%
XML External Entity	0	0	0.00%	0.00%
XML Entity Expansion	0	0	0.00%	0.00%
XML Injection	0	0	0.00%	0.00%
XQuery Injection	0	0	0.00%	0.00%

Application Misconfiguration	1	1	0.00%	0.67%
Directory Indexing	2	2	0.01%	1.33%
Fingerprinting	8	6	0.03%	4.00%
Improper Parsing	0	0	0.00%	0.00%
Improper Permissions	0	0	0.00%	0.00%
Information leakage	13598	81	48.96%	54.00%
Insecure Indexing	0	0	0.00%	0.00%
Insufficient Anti-automation	2	2	0.01%	1.33%
Insufficient Authentication	324	45	1.17%	30.00%
Insufficient Authorization	89	6	0.32%	4.00%
Insufficient Data Protection	52	9	0.19%	6.00%
Insufficient Process Validation	5	3	0.02%	2.00%
Insufficient Session Expiration	78	28	0.28%	18.67%
Insufficient Transport Layer Protection	123	26	0.44%	17.33%
Server Misconfiguration	1	1	0.00%	0.67%
Total	27772	150		

T. 17 Vulnerabilities distribution by risk Threat rank

Threat rank	N of Vulns	N of Sites	N of Sites	% Sites
Urgent	1353	75	4.87%	50.00%
Critical	12599	138	45.37%	92.00%
High	13673	93	49.23%	62.00%
Medium	139	36	0.50%	24.00%
Low	8	6	0.03%	4.00%

10. LICENSE

Terms and Conditions for Copying, Distributing, and Modifying Items other than copying, distributing, and modifying the Content with which this license was distributed (such as using, etc.) are outside the scope of this license.

1. You may copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the OC a copy of this License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use offline, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee for the OC itself. You may not charge a fee for the sole service of providing access to and/or use of the OC via a network (e.g. the Internet), whether it be via the world wide web, FTP, or any other method.

2. You may modify your copy or copies of the OpenContent or any portion of it, thus forming works based on the Content, and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
a) You must cause the modified content to carry prominent notices stating that you changed it, the exact nature and content of the changes, and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But

when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.

3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

NO WARRANTY

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.
5. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THE OC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.