MOST VULNERABLE WEB APPLICATION IN 2013: XSS, PHP AND MEDIA SITES

Anna Breeva, Eugenia Potseluevskaya

Nowadays most commercial companies and public institutions use web-based services that handle both sensitive personal and financial information. Unfortunately, many of these companies fail to follow recommended security guidelines either created in-house or supplied by third party application developers. This has resulted in websites becoming all too often the entry point used by hackers for Denial-of-Service (DDoS) attacks, for interrupting and obstructing service availability, and for posting misinformation in an attempt to damage a company's reputation. This article outlines the most common web application vulnerabilities found by Positive Technologies in 2013.

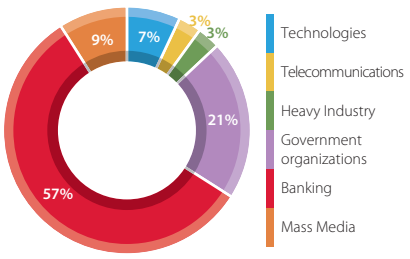
Research Methodology

The following data was collected by Positive Technologies during our analysis of 500 web applications that were used across 61 websites.

Most of the websites examined are from the **banking industry (57%);** where there is a growing demand for web application security analysis. In addition, we also see an increase in interest, for security testing, from Mass Media; possibly due to the fears related to how false news stories could impact the public.

Based on the web applications analyzed, we found the most common programming languages used were PHP (39%), Java (37%) and ASP.NET (20%). We also found that most web servers are based on Apache (39%) or Nginx (37%).

Security was assessed by means of white-, grey- and black-box testing conducted with the help of automated tools. The statistics only include data about external web applications available from the Internet and the severity of vulnerabilities found we classified in accordance with CVSSv2.



Vulnerability analysis by industry

Summary of Results

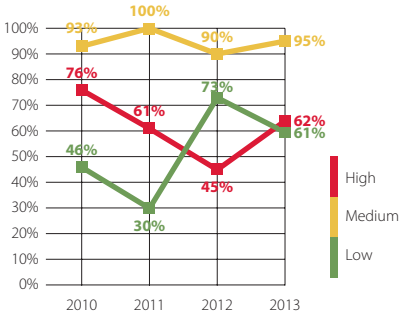
62% of the examined systems include **vulnerabilities of a high severity level;** a sharp increase from 45% of systems reported in 2012. Vulnerabilities of a medium severity level were detected in 95% of the systems tested.

In 2013, the most common vulnerability was **Cross-Site Scripting,** detected on 78% of examined sites. **ID or Password Brute Force** was the second most common vulnerability found at 69%.

The TOP-10 also included two vulnerabilities of a high severity level — **SQL Injection** (43%) and **XML External Entity** (20%). In 2012, XML External Entity failed to rank anywhere in the top ten. However, it has become more popular as new attacks like XML Out-Of-Band Data Retrieval have occurred. If exploited, an attacker could access third-party resources, read server files and even cause denial-of-service.

Authorization flaws (24%) are another notable vulnerability worth highlighting. Although a medium level severity flaw, it could cause serious consequences, such as fraudulent transactions in an e-banking system or theft within online shopping.

Vulnerabilities are mostly (89%) due to errors in application code; only 11% are caused by configuration flaws.

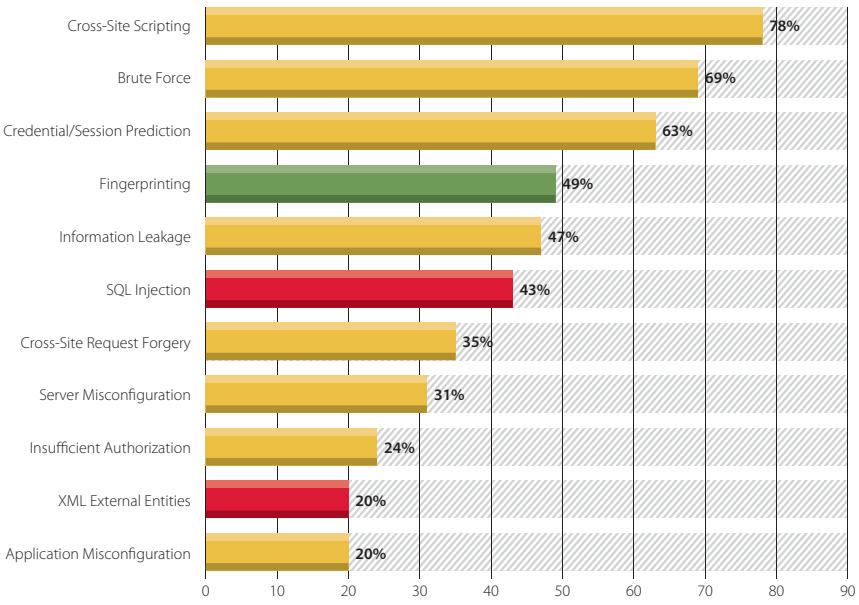


Percentage of Websites by vulnerability severity level

Most Insecure Language

76% of sites written in **PHP** have high vulnerabilities; while web applications written in Java and ASP.NET are less vulnerable (70% and 55% respectively). Applications written in all three languages were found to have medium severity level vulnerabilities.

The statistics by average number of vulnerabilities in a system prove that sites written in PHP are most vulnerable. On average, every PHP application has 12 critical vulnerabilities, whereas Java and ASP .NET applications contain 2 or less critical vulnerabilities.



Most common vulnerabilities by website (%)

POSITIVE TECHNOLOGIES

PHP	% of websites	Java	% of websites	ASP.NET	% of websites
Cross-Site Scripting	90	Cross-Site Scripting	80	Cross-Site Scripting	73
Credential/Session Prediction	86	Fingerprinting	60	Brute Force	73
Brute Force	81	Brute Force	45	Fingerprinting	55
Information Leakage	67	Credential/Session Prediction	45	Cross-Site Request Forgery	55
SQL Injection	62	Server Misconfiguration	35	Credential/Session Prediction	45
Fingerprinting	43	Information Leakage	30	Information Leakage	45
Cross-Site Request Forgery	43	XML External Entities	30	Server Misconfiguration	36
Server Misconfiguration	29	SQL Injection	25	Application Misconfiguration	36
Insufficient Authorization	29	Cross-Site Request Forgery	25	XML External Entities	36
Application Misconfiguration	19	Insufficient Authorization	15	SQL Injection	27

Most common vulnerabilities by programming language

62% of sites written in PHP include the high vulnerability "SQL Injection". This percentage is lower for other languages. One main reason is that PHP versions exist that do not allow users to create parametrized SQL queries. Therefore, many programmer guides include coding examples that use insecure database queries.

Vulnerabilities by Servers

Like in 2012, resources based on **Apache Tomcat** web server had the highest percentage (75%) of critical vulnerabilities. The percentage of critical vulnerabilities found in **Microsoft IIS** and **Nginx** systems greatly increased to 71% and 57% respectively; while they declined in Apache-based resources from 88% in 2012 to 60% in 2013.

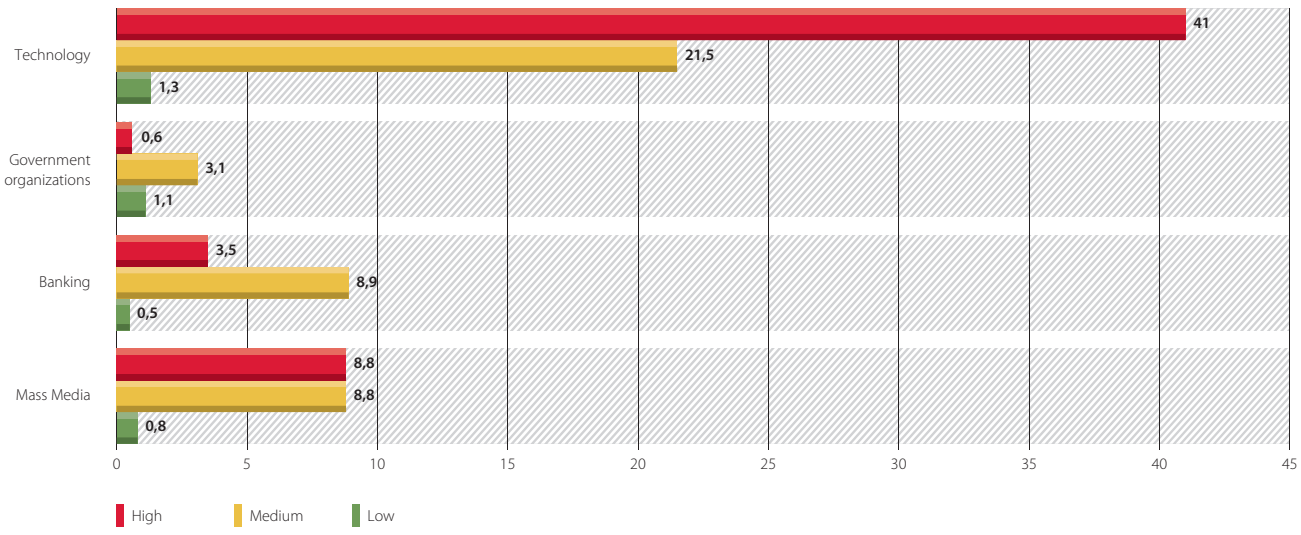
The most common administration error was **Information Leakage** which was present in 45% of all examined resources.

Vulnerabilities by Industry

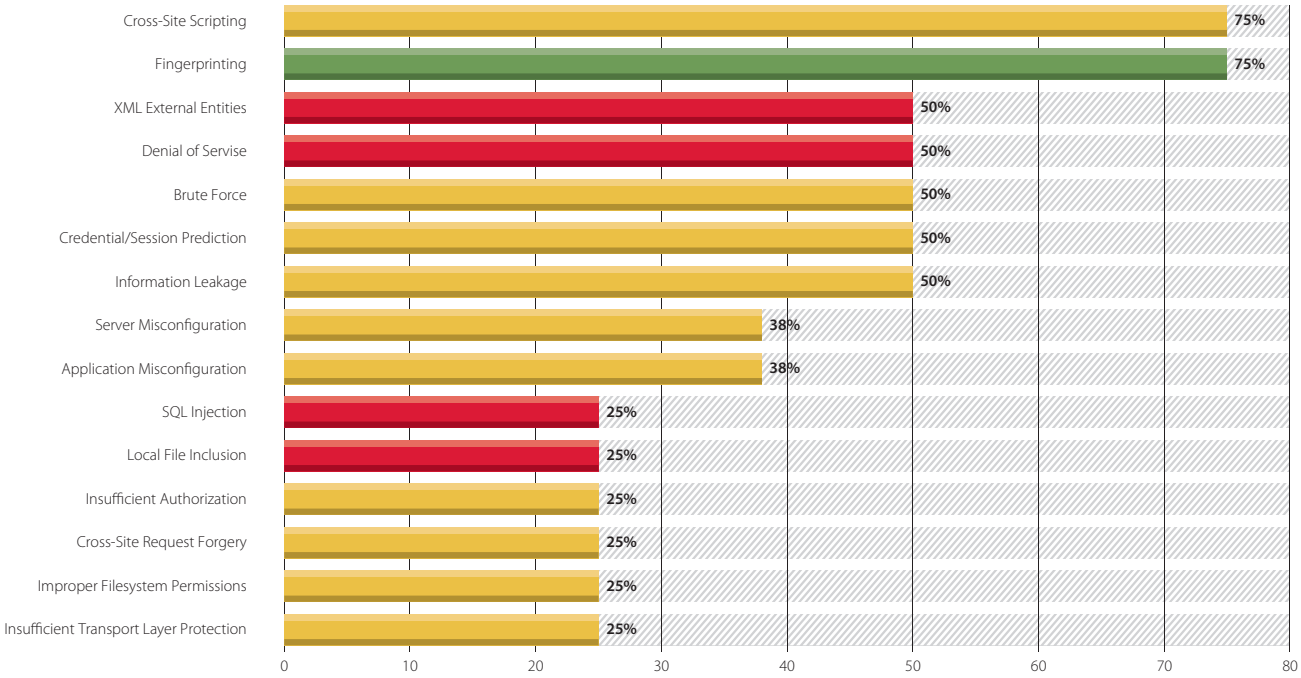
Mass Media sites contained web applications with the highest percentage (80%) of high severity level vulnerabilities. The second largest number of high severity vulnerabilities were found on Technology company sites (75%), with e-Banking websites rounded out the top three with 67%. Government sites recorded the lowest percentage of vulnerable web applications with 33%.

These results however greatly depend on the data available for analysis. For example, since Technology companies usually provided access to web application source code, than their sites were often analyzed using white-box testing; making the results more accurate.

On the other hand, the state of Government sites could actually be much worse than reported, since we were not given special permissions to these sites and therefore could only perform our analysis as an outsider with limited access to resources. We recommend using special web application protection tools like a Web Application Firewall to combat threats directed at Government organizations. In 2013, we found only one site that used such protection measures, compared with 30% of similar sites in 2012.



Average number of vulnerabilities by industry



Most common vulnerabilities in e-banking systems (% of vulnerable systems)

e-Banking Vulnerabilities

In 2013, the most common flaws found in e-banking systems were **Cross-Site Scripting** and **Application Identification.** These vulnerabilities were present in **75%** of systems tested. XML External Entity was more prevalent than in 2012 and Brute Force (2012 leading vulnerability) is now on the third most commonly found flaw.

We also assessed how compliant e-banking systems were with PCI DSSv3 (chapter 6.5). **None of the e-banking systems in our study were** fully compliant with the **PCI DSS** requirements.

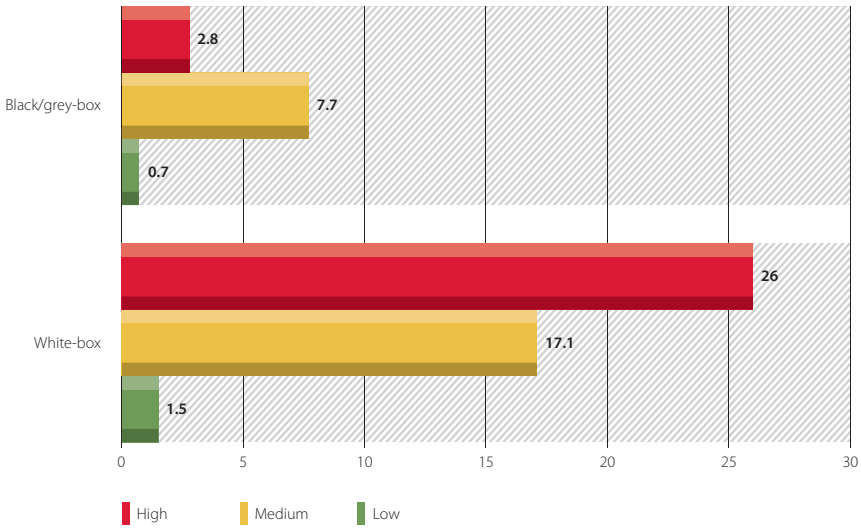
Comparison of Testing Methods

In 2013, Positive Technologies used black-, grey- and white-box test methods. Back-box testing analyzes a system without any details from its owner; grey-box testing implies an attacker with certain privileges in a system; and white-box testing, uses special privileges and access to analyze all system details including application source code.

Amongst the web sites analyzed, 60% of them were found to have critical vulnerabilities when black- and grey-box testing was used. However, the results increased to 75% when white-box testing was employed. This method is also better for detecting vulnerabilities of medium and low severity levels.

Comparing the average number of vulnerabilities for a system, we concluded that white-box testing finds almost 10 times more critical vulnerabilities and about twice as many medium and low vulnerabilities as compared to grey- and black-box methods only.

Therefore, white-box testing is preferable for source code analysis. However, companies rarely use this method: only 13% of the examined web resources were done using white-box testing.



Number of vulnerabilities by testing method (by severity level)

Positive Technologies Helped SAP to Fix Vulnerabilities

The experts of Positive Technologies regularly perform research and organize webinars to help eliminate vulnerabilities in SAP systems. SAP released an update in November 2013 that fixed vulnerabilities in its products. In particular, a flaw detected by our experts in the service-oriented integration platform NetWeaver was eliminated; which is the basis of all SAP Business Suite applications.

If this vulnerability that was detected by Dmitry Gutsko and Dmitry Sklyarov, the specialists of Positive Research Center, had been exploited than confidential information might have been disclosed. With access to the table RSECTAB of the SAP server, a malicious user can retrieve passwords of other SAP systems with RFC connections established. It is a high severity level vulnerability for SAP NetWeaver 7.20, SAP_BASIS 7.3 and their earlier versions.