Webapps Executive Report

January 13, 2017 at 4:45 PM

This report provides summarized information of every vulnerable web page found by Core Impact. This report is a brief summary divided in two sections, the first one shows the numbers comprising the test, and the second is a table showing the most exploited web pages.

Workspace(s) information

Name	Company/Test Area	Started	Finished	Exact Time	Running Time
Demo		01/13/17 12:08 PM	01/13/17 04:44 PM	4h 36m 19s	23m 56s

Starting URLs

Workspace	Scenario	Starting URL
Demo	vmcorelab	http://www.vmcorelab.com/

Summary

Risks

Vulnerabilities	Total
Successfully exploited	17
Exploited in webpages	17
Exploited in hosts	0

Assets

URLs	Total
Found	32
At Risk (confirmed vulnerabilities)	14
Broken links	1

Hosts	Total
Found (by 'WebApps Web Server Network Vulnerability Test' module)	0
At Risk (confirmed vulnerabilities)	0

General

Effort	Total
Modules run	65

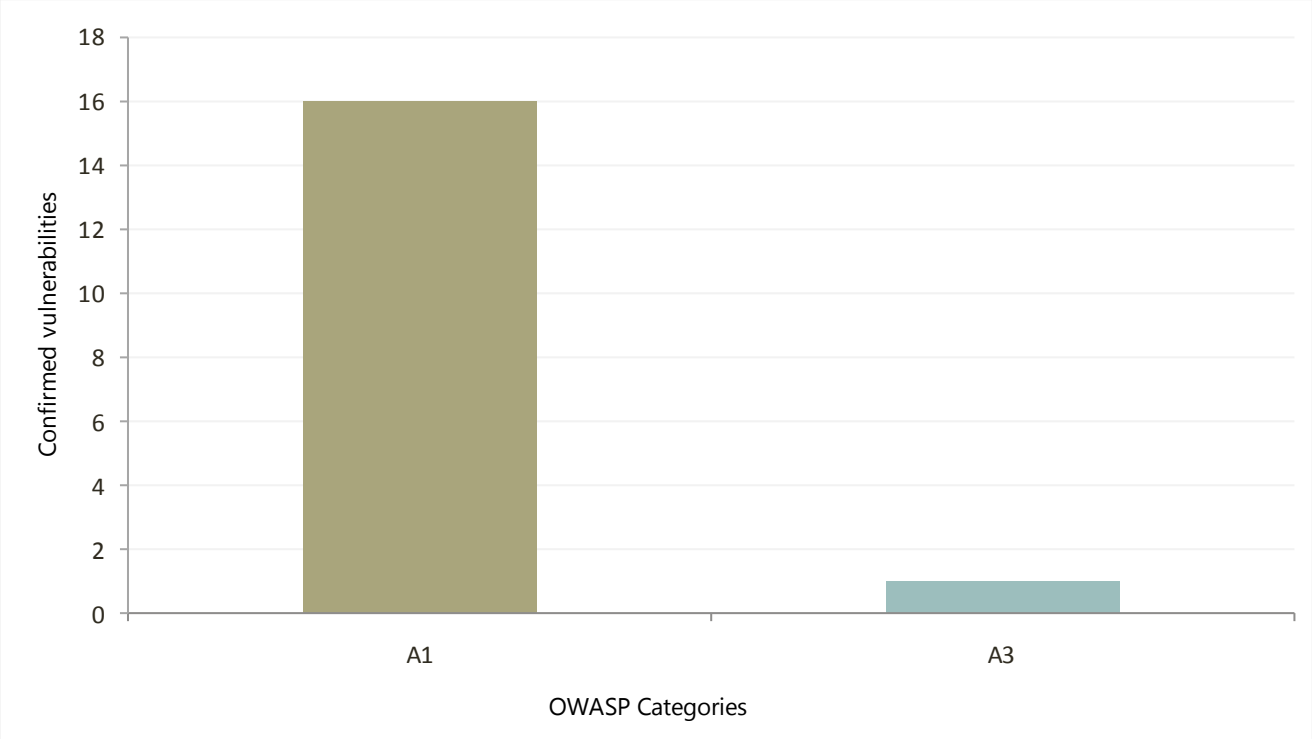
Vulnerabilities breakdown

All

OWASP Top Ten Categories	Pages Tested	Vulnerabilities
A1. Injection	**32**	**16**
SQL Injection	32	16
HTML	-	16
Web Service	-	0
OS Command Injection	32	0
HTML	-	0
Web Service	-	0
A2. Broken Authentication and Session Management	**0**	**0**
Weak Credentials	0	0
A3. Cross-Site Scripting (XSS)	**32**	**1**
HTML	32	1
Reflected	-	1
Persistent	-	0
Flash	0	0
A4. Insecure Direct Object References	**0**	**0**
Hidden Web Pages	0	**0**
A5. Security Misconfiguration	0	0
WebDAV	0	0
Default Host Credentials	0	0
A6. Sensitive Data Exposure	**0**	**0**
Sensitive Information	0	0
Database	0	0
Web Pages	0	0
Weak SSL Ciphers	0	0
A7. Missing Function Level Access Control	0	**0**
A8. Cross-Site Request Forgery (CSRF)	0	**0**
A9. Using Components with Known Vulnerabilities	0	**0**
Host Vulnerabilities	0	0
A10. Unvalidated Redirects and Forwards	0	**0**
Other Vulnerabilities	**0**	**0**
RFI for PHP	0	0
LFI for PHP	0	0

Vulnerability analysis

Confirmed vulnerabilities grouped by OWASP categories



Top ten most vulnerable urls

Workspace	Scenario	URL	Vulns
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/blind/sql_injection_string_error_rewrite.aspx?filter=	2
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_linq_example_orderby_1.aspx?filter=&order=LastName	2
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_linq_example_nested_1.aspx?filter=	2
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_linq_example_min_1.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_linq_example_join_1.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_linq_example_conditional_1.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/verbose/sql_injection_integer.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/blind/sql_injection_string_redirect.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/blind/sql_injection_string.aspx?filter=	1
Demo	vmcorelab	http://www.vmcorelab.com/testcases/sqli/blind/remoteonly/sql_injection_string.aspx?filter=	1

References

A1 - Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2 - Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

A3 - Cross Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A4 - Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

A5 - Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

A6 - Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

A7 - Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

A8 - Cross Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

A9 - Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

A10 - Unvalidated Redirects and Forwards

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.