Version: 1.5
Effective: December 18, 2014

Annex B - Content Management System (CMS) Qualifying Procedure

This document is an annex to the Government Web Hosting Service (GWHS) Memorandum Circular.

Content

Purpose

This document contains the qualifying process that allows agencies' custom platforms to be hosted under GWHS.

Scope

Specifically, this document shall cover the following:

A) Security audit procedure that allows custom platforms under the web hosting service.

INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

B) Description for each process.

Issuing Authority

The Security Audit Procedure has been developed and issued by the Department of Science and Technology's Information and Communications Technology Office (DOST-ICT Office) and Advance Science and Technology Institute (DOST-ASTI) through the iGovPhil Project.

Contact Information

Policies and associated publications under iGovPhil Project can be found at http://i.gov.ph/.

Queries, suggestions and clarifications with regard to this policy may be forwarded to inquiry@i.gov.ph.

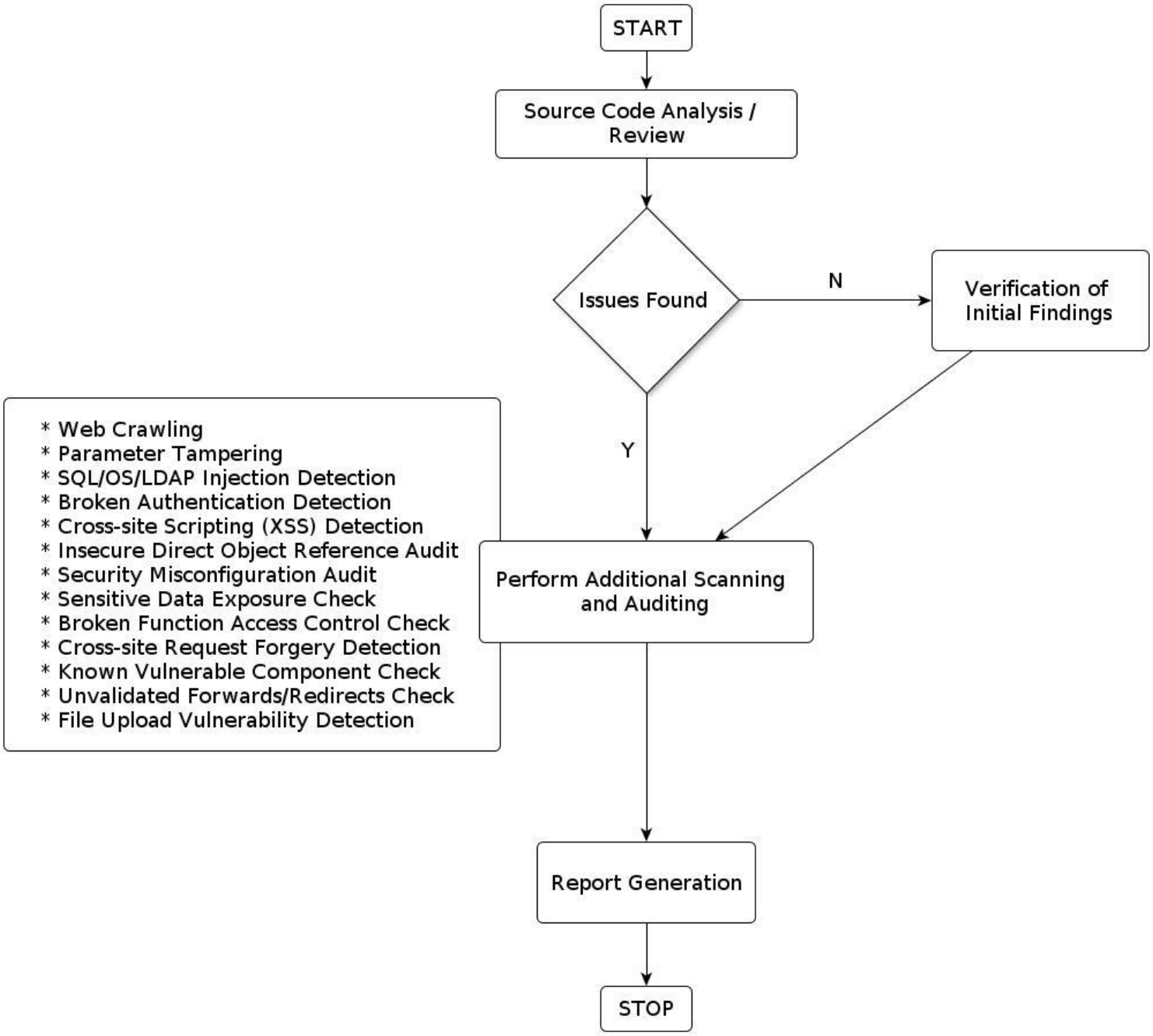
INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

1. Introduction

Agencies may use their custom platforms under the Government Web Hosting Service (GWHS). These platforms must pass the security audit procedure.

2. Security Audit Procedure

The flowchart below demonstrates the step-by-step security audit procedure.



INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

2.1 Definition of Terms:

2.1.1 Source Code Analysis and Review

The Source Code Analysis is designed to evaluate the source code for security flaws. With the help of particular tools, this review helps analysts zero in on the code's relevant portions so they can find the vulnerabilities more effectively.[i]

2.1.2 Vulnerability Scanning and Auditing

This is a combination of manual and automated processes that aims to seek out security vulnerabilities of computing systems, basically to determine if and when a system can be threatened.[ii]

2.1.3 Verification of Issues Found

The analysts verify and zoom in on the found security flaws to assess their impacts to the system and determine actions to mitigate them.

2.1.4 Web Crawling

Using tools called web robots, crawlers or spiders, web crawling is used to automate tasks on a particular website. Such tasks include identifying all links and URLs in web pages.[iii]

2.1.5 Parameter Tampering

A certain procedure is done to deal with a particular web-based attack in which parameters in the Uniform Resource Locator (URL) or web page from field data entered by the user are manipulated by performing a man-in-the-middle attack. Tampered parameters may not be handled properly by the web application. Measures to prevent parameter tampering include validation of all parameters to ensure that they adhere to the standards on minimum and maximum allowable length, character sequences and patterns, and numeric range.[iv]

2.1.6 SQL, OS and LDAP Injection Detection

Injection flaws, such as SQL, OS and LDAP injection, enable attackers to pass malicious code to another subsystem through a poorly designed web application. This is where the injection detection steps in, specifically to identify the types of injection attacks and to eliminate the risks brought about by such flaws.[v]

INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

2.1.7 Broken Authentication Detection

This procedure is performed to proficiently deal with broken authentication security risk. Through this risk, an attacker can get the user's important data (e.g., passwords), impersonate an administrator or another user, or access sensitive data without proper authentication. The risk arises when web application functions that deal with user authentication and session management are not properly implemented.[vi]

2.1.8 Cross Site Scripting (XSS)

XSS flaws need to be checked as these allow the attacker to execute scripts in the victim's browser. This attack can lead to website defacement and user sessions being hijacked. It can as well redirect the end user to a malicious website. Proper filtering and sanitation of user inputs are recommended to minimize the risk of XSS.[vii]

2.1.9 Unsecure Direct Object Reference Audit

Analysts need to test web applications for unsecure direct object references to prevent unauthorized access to data. Direct object references occur when a developer exposes a reference to an internal implementation object, like database key, file or directory.[viii]

2.1.10 Security Misconfiguration Audit

Secure settings need to be defined, implemented and maintained to mitigate the risk of security misconfiguration. This risk occurs when secure configurations for all frameworks, platforms, and servers are not defined. This also arises when the code libraries being used by the application are not updated.[ix]

2.1.11 Sensitive Data Exposure

Attackers may steal or modify weakly protected data to conduct different forms of crimes. As such, sensitive data deserve extra protection, such as encryption on storage or in transit.[x]

2.1.12 Broken Function Access Control Check

Attackers have the ability to forge requests to access functionality without proper authorization. Hence, as a countermeasure, the broken function access control needs to be properly checked or audited by analysts.[xi]

2.1.13 Request Forgery (CSRF) Detection

In this kind of security breach, attackers can send forged HTTP requests to a vulnerable web application through a logged-on victim who is unaware of the crime.[xii] The web application receiving the request has no capacity to verify if the request is sent by the

original user or by the attacker. Thus, there is a need to detect such form of security breach using a particular tool to modify the HTTP/HTTPS and POST parameters.[xiii]

2.1.14 Known Vulnerable Component Check

Known vulnerable components, such as frameworks, libraries and other software modules, need to be evaluated as these can be easily exploited. When these components are attacked, it can result in severe data loss or, worse, server takeover.[xiv]

2.1.15 Unvalidated Redirects and Forwards

Unvalidated Redirects and Forwards need to be found and eradicated in the coding, using a code analysis tool. Unvalidated redirects, in which web applications direct users to different pages and links without any validation, can lead the end-user to websites and pages that are malicious and unsecure.[xv]

2.1.16 File Upload Vulnerability Detection

Uploaded files pose threat to web server. File upload is a way for an attacker to exploit the victim's system by executing malicious codes. This then will result in server takeover and serious data loss. Hence, such vulnerability needs to be detected and processed or resolved.[xvi]

2.1.17 Report Generation

Process of consolidating information generated by the security audit and assessment.

INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

Revision History

Version	Releasing Date	Author	Status + Description
1.0	November 14, 2013	iGovPhil Project	
1.1	November 18, 2013	iGovPhil Project	Revised the flowchart and added some sections
1.2	November 26, 2013	iGovPhil Project	Incorporated the comments from Mr. Raymond Nunez and updated the workflow
1.3	November 30, 2013	iGovPhil Project	Divided the Annex A into two: List of Approved CMSs (Annex A) and CMS Qualifying Procedure (Annex B)
1.4	August 7, 2014	iGovPhil Project	Removed custom platforms, edited definitions in Section 2.1
1.5	November 13, 2014	iGovPhil Project	Changed the letterhead

INFORMATION AND COMMUNICATIONS TECHNOLOGY OFFICE
ICTO Building, C.P. Garcia Avenue, Diliman, Quezon City 1101, Philippines
+63 (02) 920-0101

[i] https://www.owasp.org/index.php/Source_Code_Analysis_Tools
[ii] http://www.webopedia.com/TERM/V/vulnerability_scanning.html
[iii] http://EzineArticles.com/1594941
[iv] http://searchsecurity.techtarget.com/definition/parameter-tampering
[v] https://www.owasp.org/index.php/Injection_Flaws
[vi] http://EzineArticles.com/7329972
[vii] https://www.owasp.org/index.php/Top_10_2013-Top_10
[viii] https://www.owasp.org/index.php/Top_10_2013-Top_10
[ix] http://EzineArticles.com/7329972
[x] https://www.owasp.org/index.php/Top_10_2013-Top_10
[xi] https://www.owasp.org/index.php/Top_10_2013-Top_10
[xii] https://www.owasp.org/index.php/Top_10_2013-Top_10
[xiii] http://EzineArticles.com/7329972
[xiv] https://www.owasp.org/index.php/Top_10_2013-Top_10
[xv] http://EzineArticles.com/7329972
[xvi] http://resources.infosecinstitute.com/file-upload-vulnerabilities/