Appendix F - Additional Resources for Web Vulnerabilities

Summary

The following is a list of resources on mitigating, remediating, and preventing vulnerabilities, for system owners that receive a vulnerability reports via their CVD process. After validation is performed on the report, these resources assist in determining an appropriate remediation response.

Vulnerabilities present in current operational systems must be remediated after a proper validation and risk assessment are performed. Custom development should integrate secure coding practices, and risk analysis should be performed before the custom web application is placed on the internet in operation.

Overview

Vulnerability mitigation consists of two distinct processes.

First, during development and deployment, web applications should be developed with secure coding practices after threat modeling, risk analysis, and secure design are is performed. The application should be deployed following best system administration practices. Testing tools can be used identify vulnerabilities before the application is deployed.

Second, after deployment, vulnerabilities discovered in active production web applications will need to be remediated based on severity, impact, threat, and other risk factors. Testing tools can again be used to identify vulnerabilities and help monitor deployed applications.

Below are resources that may be useful for secure design and development, testing and vulnerability identification, understanding common web application vulnerabilities, and vulnerability mitigation and remediation.

Testing and Validation of Web Vulnerabilities

To perform testing and validation of reported web vulnerabilities, we recommend the use of a Windows virtual machine (VM) running a recent version of Windows with Firefox, Chrome, Burp Suite, and OWASP ZAP installed, along with any dependencies. Linux and/or Mac OS systems are recommended for additional tool support, but are not necessary.

To the extent possible, validation is best performed from an open *public internet*-like network. Enterprise web proxy or other filtering or inspection systems can negatively affect the behavior of validation tools. In many cases, it’s possible to configure the tools to be proxy-aware, however some proxy or inspection systems can prevent tools from functioning properly. Firewall rules must allow necessary traffic, usually (but not always) outbound connections to HTTP/S ports TCP 80 and 443. FTP and SMTP services have also been needed in validation.

Client-based Testing Tools

Tool	Notes	Link(s)
Virtual Machine Validation Workstation	Due to the small chance that a report will contain malicious a proof-of-concept or Trojan horse, recovery and containment may be easier if analysts use virtual machines for validation workstations.	https://www.linux-kvm.org/page/Main_Page https://www.vmware.com/ https://www.virtualbox.org/ (or any other virtualization tool)

Web Browsers	Analysts should have access to Internet Explorer, Firefox, Chrome, and Edge with any necessary enterprise certificate authorities installed.	Internet Explorer https://www.microsoft.com/en-us/download/internet-explorer.aspx Firefox https://www.mozilla.org/en-US/firefox/ Chrome https://www.google.com/chrome/ Edge https://www.microsoft.com/en-us/windows/microsoft-edge Safari https://www.apple.com/safari/
Burp Suite	Requires Java JRE.	https://portswigger.net/burp/
OWASP ZAP	Requires Java JRE.	https://github.com/zaproxy/zaproxy/wiki/Downloads
Firefox Add-ons	There are several Firefox add-ons which can aid in validating reports.	Web Developer Add-on https://addons.mozilla.org/en-US/firefox/addon/web-developer/ Firebug https://addons.mozilla.org/en-US/firefox/addon/firebug/ Proxy Switcher https://addons.mozilla.org/en-US/firefox/addon/proxy-switcher/ Tamper Data https://addons.mozilla.org/en-US/firefox/addon/tamper-data/ RESTClient https://addons.mozilla.org/en-US/firefox/addon/restclient/ Cookies Manager+ https://addons.mozilla.org/en-US/firefox/addon/cookies-manager-plus/
cURL	Command-line interaction with HTTP/HTTPS services.	https://curl.haxx.se/
sqlmap	Requires Python 2.7.	http://sqlmap.org/
Metasploit Framework		https://www.rapid7.com/products/metasploit/download/
testssl.sh	Requires Linux platform.	https://testssl.sh/

Web-based Testing Tools

Tools may be used during development as well as after deployment. Since standards and best practices change over time, it is a good idea to periodically re-test deployed applications.

Tool	Notes	Link(s)
Securityheaders.io	Website-based testing tool that checks a deployed website for HTTP header security best practices and provides a list of recommended standards and best practices.	https://securityheaders.io/

OWASP Security Headers Project	Explanations and implementation guidance for HTTP security headers.	https://www.owasp.org/index.php/OWASP_Secure_Headers_Project
SSL Labs Server Test	Website-based testing tool that checks if deployed website uses proper SSL/TLS certificates and configuration. The tool provides a grade for the website and appropriate recommendations if any tests are failed.	https://www.ssllabs.com/ssltest/
Badssl.com	Website-based testing tool for ensuring clients (not servers) are properly configured for using SSL/TLS. May not be necessary depending on the service.	https://badssl.com/
OWASP WAP	Tool for detecting vulnerabilities in PHP web applications.	https://www.owasp.org/index.php/OWASP_WAP-Web_Application_Protection

Vulnerability Remediation Resources

Once the web application is deployed, the application, server, or other components may need updates, patches, or other changes to address vulnerabilities as they are discovered. In some cases, the system owner or their developer will need to develop the mitigation or remediation.

Resource	Notes	Link(s)
How to Win Friends and Remediate Vulnerabilities	Whitepaper from SANS Institute with advice on setting up a remediation capability.	https://www.sans.org/reading-room/whitepapers/application/win-friends-remediate-vulnerabilities-34530
Guide to Enterprise Patch Management Technologies	NIST report on managing patches for vulnerability remediation	http://dx.doi.org/10.6028/NIST.SP.800-40r3
SQL Injection Prevention	Advice on avoiding and fixing SQL injection vulnerabilities from OWASP	https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
Cross-Site Scripting	Advice on preventing and fixing cross-site scripting (XSS) vulnerabilities from Google.	https://www.google.com/about/appsecurity/learning/xss/#PreventingXSS
Common Weakness Enumeration (CWE)	CWE provides a taxonomy of different types of vulnerabilities. Many CWE entries provide brief advice on potential mitigations.	https://cwe.mitre.org/
CWE/SANS Top 25 Most Dangerous Software Errors	List of dangerous vulnerabilities, the *Insecure Interaction Between Components* section contains web application vulnerabilities.	https://www.sans.org/top25-software-errors/
OWASP Top Ten Project	A list of the ten most critical web application security risks.	https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
Web Application Security Consortium (WASC)	Collection of web application security resources.	http://www.webappsec.org/

Development Resources

A threat model of the service or web application being provided should be developed earlier in the process to help make architecture decisions that will keep the application more secure. Below are resources for creating and using a threat model for development or deployment, as well as secure coding development principles.

Resource	Notes	Link(s)
Threat Modeling Process by OWASP	"A structured approach to application threat modeling that enables you to identify, quantify, and address the security risks associated with an application."	https://owasp.org/www-community/Threat_Modeling_Process
***Threat Modeling* book by Adam Shostack**	A book of material on how to properly perform threat modeling for a number of scenarios. The author also offers training courses.	https://threatmodelingbook.com

Open Web Application Security Project (OWASP) Secure Coding Guide	A short guide for secure coding principles specifically tailored for web applications.	https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide
CERT Secure Coding Standards	Secure coding standards should be followed to avoid vulnerabilities as much as possible. CERT provides coding standards for common web application programming languages like Java and Perl. Note that the standards were developed for general usage, and not all rules may apply to web applications.	https://www.securecoding.cert.org/
The Basics of Web Application Security	Summary of important web application secure development practices.	https://martinfowler.com/articles/web-security-basics.html
Basic Security Practices for Web Applications	Microsoft web application security advice.	https://msdn.microsoft.com/en-us/library/zdh19h94.aspx