tenable network security

Performing PCI DSS and OWASP Web Application Audits with Nessus

January 22, 2014

(Revision 8)

Ron Gula – Chief Executive Officer, Chief Technology Officer

Table of Contents

Overview

Tenable Network Security offers solutions to perform vulnerability scanning, passive network monitoring, configuration auditing, real-time log collection, and analysis of enterprise applications and networks. This paper focuses on performing web application audits specific to the following standards using Tenable's Nessus vulnerability scanner:

- The Open Web Application Security Project (OWASP) Top 10 for 2013
- Payment Card Industry Data Security Standard (PCI DSS)

This paper reflects standards described by version 3.0 of the Payment Card Industry Data Security Standard (PCI DSS) requirements, with specific attention given to demonstrating PCI DSS 6.5 and 6.6 compliance requirements. While Tenable focuses on performing web application vulnerability assessments to demonstrate compliance with PCI DSS 6.6, running a web application firewall or performing a source code audit may also help to fulfill the compliance requirement.

In relation to the PCI DSS standard, this paper focuses on how Nessus can be used to simulate an Internet-based scan from an Authorized Scanning Vendor (ASV). PCI does not allow organizations to self-certify but requires an external vulnerability scan from an ASV, such as Tenable Network Security. Several other ASVs make use of Tenable's Nessus vulnerability scanner. This paper demonstrates how to perform internal testing to be better prepared for certification testing. Technical details on using Nessus for total PCI coverage into configuration audits, antivirus testing, and patch testing, as well as Tenable's enterprise network monitoring and logging solutions, are covered in the "Real-Time PCI Compliance Monitoring" paper referenced below.

Tenable's Research team continuously updates Nessus' logic to perform web-based audits. Updates come from research performed by Tenable, feedback from customers such as Qualified Security Assessors (QSAs) performing PCI assessments, certified ASVs that use Nessus to perform PCI DSS scanning, and from regulatory requirements beyond PCI DSS such as the U.S. government's DISA STIG standards. As such, Nessus may have more advanced web-based audits available than what is described in this paper.

As of December 2013, searching for the term "CGI Generic" in the list of available Nessus plugins lists the following 49 Nessus checks:

ID	Name	Family
11139	CGI Generic SQL Injection	CGI abuses
33817	CGI Generic Tests Load Estimation (all tests)	CGI abuses
39465	CGI Generic Command Execution	CGI abuses
39466	CGI Generic Cross-Site Scripting (quick test)	CGI abuses : XSS
39467	CGI Generic Path Traversal	CGI abuses
39468	CGI Generic Header Injection	CGI abuses
39469	CGI Generic Remote File Inclusion	CGI abuses
39470	CGI Generic Tests Timeout	CGI abuses
40406	CGI Generic Tests HTTP Errors	CGI abuses
42054	CGI Generic SSI Injection	CGI abuses
42055	CGI Generic Format String	CGI abuses
42056	CGI Generic Local File Inclusion	CGI abuses
42423	CGI Generic SSI Injection (HTTP headers)	CGI abuses
42424	CGI Generic SQL Injection (blind)	CGI abuses
42425	CGI Generic Cross-Site Scripting (persistent)	CGI abuses : XSS
42426	CGI Generic SQL Injection (HTTP Cookies)	CGI abuses
42427	CGI Generic SQL Injection (HTTP Headers)	CGI abuses
42479	CGI Generic SQL Injection (2nd pass)	CGI abuses
42872	CGI Generic Local File Inclusion (2nd pass)	CGI abuses
43160	CGI Generic SQL Injection (blind, time based)	CGI abuses
44134	CGI Generic Unseen Parameters Discovery	CGI abuses
44136	CGI Generic Cookie Injection Scripting	CGI abuses
44967	CGI Generic Command Execution (time-based)	CGI abuses
46193	CGI Generic Cross-Site Scripting (HTTP Headers)	CGI abuses : XSS
46194	CGI Generic Path Traversal (write test)	CGI abuses
46195	CGI Generic Path Traversal (extended test)	CGI abuses
46196	CGI Generic XML Injection	CGI abuses
47830	CGI Generic Injectable Parameter	CGI abuses
47831	CGI Generic Cross-Site Scripting (comprehensive test)	CGI abuses : XSS
47832	CGI Generic On Site Request Forgery (OSRF)	CGI abuses
47834	CGI Generic Open Redirection	CGI abuses
48926	CGI Generic 2nd Order SQL Injection Detection (potential)	CGI abuses
48927	CGI Generic SQL Injection Detection (potential, 2nd order, 2nd pass)	CGI abuses
49067	CGI Generic HTML Injections (quick test)	CGI abuses : XSS
50413	CGI Generic Padding Oracle	CGI abuses
50418	CGI Generic Fragile Parameters Detection (potential)	CGI abuses
50494	CGI Generic Path Traversal (quick test)	CGI abuses
51528	CGI Generic Command Execution (time-based, intrusive)	CGI abuses
51529	CGI Generic Cross-Site Scripting (persistent, 2nd pass)	CGI abuses : XSS
51972	CGI Generic Cross-Site Scripting (Parameters Names)	CGI abuses : XSS
51973	CGI Generic SQL Injection (Parameters Names)	CGI abuses
52483	CGI Generic Cross-Site Scripting (persistent, 3rd Pass)	CGI abuses : XSS
55903	CGI Generic Cross-Site Scripting (extended patterns)	CGI abuses : XSS
55904	CGI Generic Script Injection (quick test)	CGI abuses : XSS
56242	CGI Generic Tests Load Estimation (full tests)	CGI abuses
56243	CGI Generic Tests Load Estimation (quick tests, text injection)	CGI abuses
56244	CGI Generic Tests Load Estimation (quick tests, HTML injection)	CGI abuses
56245	CGI Generic XPath Injection (2nd pass)	CGI abuses
56818	CGI Generic Cross-Site Request Forgery Detection (potential)	CGI abuses

In addition to these generic checks, Nessus includes thousands of specific vulnerability checks for known security issues in web servers, web applications, web APIs, and web management interfaces.

The following resources, available on the Tenable website under the "Whitepapers" section, provide more information describing how Tenable can help to perform broader compliance analysis:

- Real-Time FISMA Compliance Monitoring
- Real-Time Massachusetts Data Security Law Monitoring
- Real-Time PCI Compliance Monitoring
- Web Application Scanning with Nessus

Each of the covered standards is introduced followed by a brief description of how Nessus web-based audits can be used to help achieve compliance with the standard. Nessus scanning techniques can be accomplished with Nessus as a standalone scanner as well as when being managed by Tenable's SecurityCenter. In addition, each paper has a chapter covering unique web-based auditing technologies from Tenable including passive network analysis, configuration auditing, database activity monitoring, log analysis, and file integrity checking.

OWASP

Tenable Network Security has specifically added technology and checks to the Nessus vulnerability scanner to make it easier to find risks identified by this project. For a quick configuration guide to set up an OWASP Top 10 scan policy in Nessus, consult Appendix 1.

OWASP first published web application audit guidelines in 2003, which were then updated in 2004, 2007, 2010, and again in 2013. OWASP guidelines are labeled as risks A1 through A10. A table describing the high-level changes and what is covered between the 2010 and 2013 releases is shown below:

OWASP Top 10 – 2010		OWASP Top 10 – 2013	
A1	Injection	A1	Injection
A2	Cross-Site Scripting (XSS)	A2	Broken Authentication and Session Management
A3	Broken Authentication and Session Management	A3	Cross-Site Scripting (XSS)
A4	Insecure Direct Object References	A4	Insecure Direct Object References
A5	Cross-Site Request Forgery (CSRF)	A5	Security Misconfiguration
A6	Security Misconfiguration	A6	Sensitive Data Exposure
A7	Insecure Cryptographic Storage	A7	Missing Function Level Access Control
A8	Failure to Restrict URL Access	A8	Cross-Site Request Forgery (CSRF)
A9	Insufficient Transport Layer Protection	A9	Using Components with Known Vulnerabilities
A10	Unvalidated Redirects and Forwards	A10	Unvalidated Redirects and Forwards

Each of the OWASP Top 10 risks identified in the 2013 recommendations are covered below. Each section includes a short discussion on how Nessus' web application tests or vulnerability scanning techniques can be used to identify OWASP risks.

2013 OWASP Top 10 – A1 Injection

When user input is interpreted by a web application, it may result in execution of code by a back-end process. Common examples of this include:

- SQL injection – user-controlled data results in arbitrary SQL statements being executed by a backend database.
- Command execution – user-controlled data is processed in such a way that users can cause arbitrary operating system commands or language framework code (e.g., PHP) to run.
- LDAP injection – user-controlled data is processed in an LDAP query, resulting in arbitrary commands being executed.

There are many other injection points. The basic concept is that user-submitted data is not cleanly processed and is fed directly into an interpreted set of executable code.

Nessus tests for many different types of injection attacks including:

Nessus A1 Injection Techniques	Description
Generic SQL Injections	Nessus includes several plugins to test for SQL injection issues through multiple techniques including: • Traditional SQL injection • SQL injection through HTTP cookies • SQL injection through HTTP headers • Blind SQL injection (logic) • Time-based blind SQL injection • 2nd order SQL injection
Specific SQL Injections	As of December 2013, Nessus included network and patch audits for more than 400 specific SQL injection vulnerabilities for applications such as Drupal, Joomla, and Bugzilla.
XPATH Injection	Nessus is able to identify XPATH injection security issues through blind SQL injection testing.
SSI Injection	Nessus includes two generic tests for traditional SSI injection as well as SSI injection through HTTP headers.
Command Execution	Nessus includes three generic tests for command execution. The first performs basic parameter pollution to look for command execution. The second test performs time-based attacks to detect command execution and the third test uses time-based detection in a more intrusive and thorough manner.

In addition to the tests that specifically deal with injection, Nessus' cross-site scripting (XSS) checks, covered in the next section, also make use of a variety of "injectable parameters". Nessus also performs a wide variety of checks for other types of injection including directory traversal attacks, format strings, file inclusions, and more.

2013 OWASP Top 10 – A2 Broken Authentication and Session Management

Web sites that have security issues may permit users to exploit a vulnerability that allows them to steal the credentials or impersonate another user on the web application. The OWASP project asks seven questions to determine if an application's authentication or session management is potentially vulnerable:

OWASP A2 Questions	Nessus Audit Technique
Are credentials always protected when stored using hashing or encryption?	Nessus checks that HTTP authentication occurs over TLS and reports accordingly. Cookies are displayed, including all their attributes. Session cookies are checked against disclosure (e.g., do they have “secure” or “HttpOnly” attributes? Are they transmitted over HTTPS?).
Can credentials be guessed or overwritten through weak account management functions?	Nessus users can leverage the Hydra brute force guessing tool to test for weak passwords. Nessus also includes several checks for common default or backdoor accounts in web applications. Tenable offers many different configuration audit policies for applications that use the authentication features of Apache or IIS to help test the configuration of these servers.
Are session IDs exposed in the URL?	Nessus does not currently implement logic to generically check for session IDs. Nessus does check for a variety of session ID vulnerabilities in known applications and also tests for session ID randomness.
Are session IDs vulnerable to session fixation attacks?	Nessus uses multiple methods to test for this issue including cookie injection and manipulation.
Do session IDs timeout and can users log out?	Tenable recommends that this test be performed manually, although Nessus scan preferences support the concept of a re-authentication delay that can be set arbitrarily low to see if a session can be forced to time out.
Are session IDs rotated after successful login?	Nessus has a plugin that tests session fixation as well as several other checks to enumerate all session IDs.
Are passwords, session IDs and other credentials sent only over TLS connections?	Nessus checks that HTTP authentication occurs over TLS and reports accordingly.

2013 OWASP Top 10 – A3 Cross-Site Scripting (XSS)

Cross-site scripting can occur when user-submitted data is rendered to other users in an unfiltered manner. This permits a malicious attacker to execute hostile or misleading code in the user’s web browser.[1]

OWASP defines three types of XSS issues: **stored**, **reflected**, and **DOM**.

Stored XSS results from submitting user data to a database or back-end process that stores the data before rendering it for other users. A typical example would be a web-based discussion group where a user’s answer or comment is displayed for all other users. This comment could have unescaped HTML or JavaScript code in it.

Reflected XSS attacks use a malicious link, rendered in email or on a web server, to send users to a vulnerable web server to exploit or attempt to exploit the browser. When the URL is processed, it immediately renders the HTML or JavaScript to the user who clicked on the link. The nature of the attack is based on the URL appearing to be trusted or subverting a web server that is trusted by the user.

DOM-based XSS attacks result from using content that modifies the Document Object Model (DOM) environment of a victim’s web browser. It is similar to a reflected XSS attack in that a malicious URL can be sent to a potential victim. However, the content is executed within the browser as compared to malicious rendering of content on a web server with a reflected attack.

The most severe form of XSS attacks results in the disclosure of a user’s session cookie or authenticated credentials, which leads to compromising the account. XSS attacks have also been used to implement a keylogger or conduct other activities that are not intended by the user.

[1] The “X” in XSS stands for “cross” and is used instead of CSS to differentiate it from a commonly used HTML initialism for “Cascading Style Sheet”.

Tenable has implemented multiple Nessus plugins to focus on the detection of most methods for XSS attacks. Nessus has seven plugins that use a variety of techniques to test for reflected and stored cross-site scripting via script parameters and headers.

Tenable has also implemented two Nessus plugins (#47830 – CGI Generic Injectable Parameter Weakness; #49067 – CGI Generic HTML Injections (quick test)) that are specifically designed to rapidly identify parameters for XSS testing.

2013 OWASP Top 10 – A4 Insecure Direct Object References

Insecure direct object references allow authorized users to change a parameter and simply access data regardless of authorization. For example, a poorly written web application may have a customer ID value that permits the user who is authorized to use that customer ID to change the customer ID to another value to gain access to a different user's account information. Guessing multiple IDs could allow an attacker to enumerate potentially sensitive data of every user of the application.

OWASP recommends code reviews to see if an application enforces indirect and direct references. OWASP also notes that automated scanners do not contain the logic to differentiate sensitive data on a typical complex web application. Despite that, Tenable feels there are several types of audits performed by Nessus that impact this OWASP risk:

- Most direct object references are the result of common weaknesses such as path traversal, SQL injections, local file injections and the dozens of other web applications tests performed by Nessus.
- The 2nd order non-blind SQL injection tests performed by Nessus can identify specific SQL tables.
- Scripts #44134 (CGI Generic Unseen Parameters Discovery) and #40773 (Web Application Potentially Sensitive Parameter Detection) will report potentially dangerous CGI parameters.

2013 OWASP Top 10 – A5 Security Misconfiguration

There are many types of vulnerabilities that can exist in the framework, operating system and web server application. A security misconfiguration that results in an exploitable vulnerability could be the result of missing patches or software configuration settings.

The OWASP project outlines five questions for performing an assessment of this risk category.

OWASP A5 Questions	Nessus Audit Technique
Do you have a process for keeping all your software up to date?	Nessus credentialed audits test for missing patches in the OS, web server, libraries such as PHP and SQL database. Nessus also has checks to see if a running service has been manually installed and not part of the software inventory that could indicate manually compiled web or database daemons.
Is everything unnecessary disabled, removed or not installed (e.g., ports, services, pages, accounts, privileges)?	Nessus vulnerability scans and credentialed audits identify all open ports. Manual inspection of these ports can identify unnecessary services and Nessus audit policies can be created to alert on unauthorized services as well. Nessus also identifies pages and directory browsing through web crawling and this can be manually inspected to identify new pages.
Are default account passwords changed or disabled?	Default accounts and privileges can be tested with Nessus automatically through dozens of plugins that test for known default credentials in common applications. In addition, Nessus scan policies can be created and configured to test for additional credentials.
Is your error handling set up to prevent stack traces and other overly informative error messages from leaking?	Nessus web application tests perform several different types of queries that will typically show up as errors in the system logs. Error configuration of the web server and underlying libraries is also something that can be audited with Nessus audit policy files. Several plugins test applications in such a way as to elicit error messages and use the information to detect vulnerabilities.

Are the security settings in your development frameworks (e.g., Struts, Spring, ASP.NET) and libraries understood and configured properly?	Nessus audit policies can be used to test the content of framework configuration files as well as to test the file integrity of the configuration files to ensure they have not changed.

2013 OWASP Top 10 – A6 Sensitive Data Exposure

Many web applications are prone to exposing sensitive data due to vulnerabilities or configuration issues, such as credit card numbers, Social Security numbers, other personally identifiable information (PII), and login credentials. Nessus performs audits for many different types of information disclosure, and numerous plugins test for weak or insecure cryptography.

OWASP A6 Questions	Nessus Audit Technique
Is any of this data stored in clear text long term, including backups of this data?	Nessus includes Windows File Contents Compliance Checks (#24760) to audit for Social Security numbers, credit card numbers, and other sensitive content.
Is any of this data transmitted in clear text, internally or externally? Internet traffic is especially dangerous.	Where Nessus has the ability to perform checks against data at rest, Tenable's Passive Vulnerability Scanner (PVS) performs checks against data in motion, with the ability to scan for sensitive data as it traverses between internal networks and from internal networks to the Internet.
Are any old / weak cryptographic algorithms used?	Nessus performs a variety of checks against many services to detect if they are still employing outdated or insecure ciphers and protocols.
Are weak crypto keys generated, or is proper key management or rotation missing?	Nessus contains dozens of plugins that check for key management and the presence (or ability to generate) weak cryptography keys on many platforms.
Are any browser security directives or headers missing when sensitive data is provided by / sent to the browser?	Nessus looks for a variety of web browsers and applications that do not properly set headers. In addition, when web application tests are performed, Nessus will attempt to manipulate and inject HTTP headers for privilege escalation and information disclosure.

2013 OWASP Top 10 – A7 Missing Function Level Access Control

Web applications are frequently vulnerable to access control weaknesses, as well as "business logic" issues. These types of problems can result in unauthorized users gaining access to privileged functions or information. For example, if an application does not check for access rights correctly, an anonymous user may be able to access a directory of all users of an application, resulting in a compromise of confidentiality.

OWASP A7 Questions	Nessus Audit Technique
Are server side authentication or authorization checks missing?	Nessus looks for a wide variety of authentication issues, including ways to bypass the process in many open source and commercial products.
Are server side checks done that solely rely on information provided by the attacker?	Nessus tests for many issues in open source and commercial products related to client-side validation being used instead of server-side.

2013 OWASP Top 10 – A8 Cross-Site Request Forgery (CSRF)

This web application weakness leverages image tags, XSS, and other techniques to trick an authenticated user to a privileged site into submitting a request that does something potentially damaging with the user's credentials. For example, consider a web application that automatically posts a message to Twitter but requires a user to authenticate to the application. If the URL method for posting the message was known ahead of time, an attacker could craft a URL with

their desired message and send it to the targeted user via XSS or embedded in an image tag. If the user clicks on the URL, their authenticated state with the application would process the URL and send the attacker's message to Twitter as the user. There have been many examples of using CSRF to reset passwords, purchase products, generate Google AdWord hits, and more.

There are multiple Nessus audits that are relevant to help ensure CSRF vulnerabilities are not exploited on your web application:

- Testing for XSS vulnerabilities with Nessus can ensure that these may not be used to perform CSRF attacks. Although not necessary to perform a CSRF attack, XSS vulnerabilities allow token-based CSRF defenses to be defeated.
- Nessus plugin #47832 performs "On Site Request Forgery Vulnerability" testing. This is a narrower form of CSRF attack testing.
- Over one hundred application and patch audit plugins test for the presence of CSRF or detect applications known to be vulnerable.

2013 OWASP Top 10 – A9 Using Components with Known Vulnerabilities

Many web applications, especially products and versions that have not been upgraded or patched, often contain known vulnerabilities across a wide variety of attack vectors. Nessus contains over 3,200 plugins that can fingerprint and detect known vulnerabilities in web applications. Plugins in the "CGI Abuses" and "CGI Abuses: XSS" families are written to enumerate vulnerabilities in web application products, both open source and commercial, that have been previously reported.

New vulnerabilities are discovered and published every day, and as a result, staying up-to-date is a must in order to perform optimally effective security scans. Every week, several dozens of plugins are added to the Nessus plugin feed, and many of these address recently discovered vulnerabilities in web applications and their underlying frameworks.

2013 OWASP Top 10 – A10 Unvalidated Redirects and Forwards

Web sites that use redirection or forwarding may be vulnerable to having their users inadvertently forwarded to hostile web sites.

Any type of query or URL that allows forwarding to another web site could be used by a hostile third party. Redirecting a user to a web site with malicious code that attacks the browser, steals session information or steals passwords are all possible scenarios. A user may be exposed to this attack through phishing and not realize that a site they already have authenticated or have access to is being used to redirect them to a malicious site.

Similarly, an attacker with knowledge of a web site's secure areas or functions could use phishing to attempt to get an authenticated user such as an administrator to do something unintended. Examples include creating an account, changing a password of a user to steal that account, and more.

Nessus check #47834 named "CGI Generic Redirection Vulnerability" tests web site forms with specially crafted parameters for redirection to a third party web site. In addition, Nessus contains five plugins that test for site redirection vulnerabilities that have been publicly disclosed.

PCI DSS Web Based Assessments

The Payment Card Industry Data Security Standard (PCI DSS) is a comprehensive set of requirements designed by the credit card industry to help merchants establish and maintain a baseline data security program in order to protect payment account information. A key component of complying with PCI DSS for all merchants is to have internet-facing sites pass a vulnerability scan that does not show any:

- SQL injection vulnerabilities
- Cross-site scripting (XSS) vulnerabilities
- Directory traversal vulnerabilities

- Cross-site request forgery (CSRF) vulnerabilities
- Out of date or insecure SSL encryption
- Any vulnerabilities with a CVSS score greater than 4.0
- Information leakage issues

This section of the paper reviews how such a scan can be performed using Nessus as a standalone scanner, Nessus when being managed by Tenable's SecurityCenter, or the Nessus Perimeter Service.

These PCI DSS scanning and minimum vulnerability requirements should not be confused with other PCI DSS requirements, such as the demonstration of a patch management program, having in depth web application security or performing a penetration test. Although Nessus is a useful tool used by many ASVs and is the cornerstone of Tenable's PCI Scanning Service ASV solution, it is not a replacement for ASVs any more than any other security tool is.

Required ASV Scanning Components

The PCI Approved Scanning Vendors (ASV) Program Guide requires that all approved scanning vendors offer the following required components:

PCI ASV General Characteristics	Nessus Audit Technique
Be Non-disruptive	Tenable engineers all Nessus checks to have as little impact on the scanned systems and network as possible.
Perform Host Discovery	Nessus supports ICMP ping sweeps, multiple types of TCP pings, UDP pings as well as using Ethernet ARP pings for host discovery.
Perform Service Discovery	Nessus can fingerprint thousands of known services, regardless of the port they are running on. Tenable routinely receives submissions for new service fingerprints as well. Service discovery occurs on both TCP and UDP protocols.
Perform OS and Service Fingerprinting	As part of Nessus' service discovery, fingerprints are used to recognize banners. In many cases, Tenable's Research team will also write a custom plugin that specifically and reliably identifies a certain protocol. For operating system recognition, Nessus uses an advanced set of heuristics to combine the results of multiple TCP/IP fingerprints and queries to specific services such as NTP or NetBIOS for high accuracy.
Have Platform Independence	Nessus performs vulnerability scanning, patch auditing, and configuration auditing against a wide range of operating systems, network devices and embedded systems.
Be Accurate	Tenable's user base provides a larger set of feedback than any other vulnerability scanner vendor. Tenable performs extensive code review, testing, and tweaking of our checks in-house, but our ability to get real-world feedback from customers in order to fix reported issues is unmatched by any other vendor.
Account for Load balancers	Nessus attempts to identify any vulnerability on any port, regardless if it is load balanced or not. Nessus plugin #12224 also detects load balanced web servers and plugin #31422 detects reverse NAT/Intercepting proxies.
Perform a scan without interference from active protection systems	There are many different types of "interference" that can be performed by an active protection system. Nessus has two plugins that detect if a system is protected by a firewall (plugin #27576) or by a web application firewall (plugin #41058). Many of the active responses performed by an active protection system will appear to Nessus as if a firewall is interfering with the scan. An active protection system that silently drops packets will not be detected by Nessus. If the system does not drop packets at once, some scripts may issue a warning (e.g., #10919).

Temporary changes may need to be made by the scan customer to remove interference during a scan	If a Nessus scanner is placed on the outside of the network, it may be possible to determine if an active protection system is blocking a scan and save time during a costly PCI DSS assessment from an ASV.

PCI also states that an ASV should target a variety of technologies, such as routers, DNS servers, and web servers and refers to them as “scan components”. The following table lists each PCI scan component and how Nessus can be used to perform a vulnerability scan without credentials.

Please note that Nessus can be provided with credentials on most Unix, Linux, Windows, database and Cisco router systems and perform an in-depth patch audit of local and third party applications as well as standards-based configuration auditing. Such credentialed audits are out of scope for performing external ASV PCI vulnerability scans, but could be used to provide more accurate scanning results if there is a potential false positive in a vulnerability reported by your ASV.

PCI ASV Scan Components	Nessus Audit Technique
Firewalls & Routers	Tenable’s Research team has written a variety of Nessus plugins that detect firewall and router devices, as well as perform uncredentialed checks for vulnerabilities in these systems. Attention is also given to the detection of web application firewalls and network proxies.
Operating Systems	Nessus’ main focus is to audit operating systems. Tenable’s Research team focuses on exposed vulnerabilities of the operating system such as file sharing and also identifies vulnerabilities in applications that may be bundled. Nessus specifically checks for out of date or unsupported operating systems, and will use this information to fail a PCI assessment scan.
Database Servers	Nessus includes many different plugins to recognize the various components of Oracle, MySQL, MS SQL, Sybase, and other databases. Once recognized, Nessus can determine a wide variety of database vulnerabilities. If a database service is found when performing an external PCI scan, the scan will fail PCI DSS compliance.
Web Servers	Nessus identifies many different vulnerabilities in web servers such as Apache and IIS. It also identifies vulnerabilities associated with embedded devices that run web interfaces. Nessus can also be configured to report vulnerabilities that may have been fixed but have had their banners “back ported” to reflect an older, vulnerable version string. If your infrastructure contains applications such as Apache and PHP, you can leverage Nessus’ credentialed checks to test the actual patch level of the system and not rely on a banner check. Nessus scans for any directory browsing which, if found, results in an automatic failure of an ASV vulnerability scan for PCI DSS compliance.
Application Server	Nessus attempts to recognize common application servers that host technology such as JBoss and WebSphere and to identify any vulnerabilities associated with them.
Common Web Scripts	Through web crawling, Nessus identifies as many of the common web scripts as possible that are in use on the web application server; common CGIs are tested even if they are not seen by the spider (i.e., not directly reachable from the start page). Nessus has vulnerability checks for many commonly known scripts and applications and also has advanced web application checks to look for common web security issues generically.

Built-In Accounts	Nessus includes several checks for common vendor default accounts, hidden accounts, and accounts commonly configured by administrators such as the “sa” database account. Nessus will also identify if cleartext authentication is available through Telnet, basic authentication, SNMP v1, FTP, rlogin, and common email protocols.
DNS Servers	Nessus identifies DNS servers, common vulnerabilities associated with them, and will automatically fail a server during an ASV scan if the system performs a zone transfer.
Mail Servers	Nessus includes many checks to identify email protocols such as SMTP, POP, IMAP, and also their secure variants. In addition, email applications such as Exchange, sendmail, Cyrus, Qmail, and others are accurately identified along with any vulnerabilities associated with them. Generic weaknesses and flaws in email services are also tested. Email services are analyzed in a similar manner to how Nessus performs scans of web servers for potentially “back ported” applications.
Web Applications	Nessus performs extensive types of web application tests. Specific to a vulnerability scan from an ASV, Nessus performs the following checks and will fail a server if any of these are found. Specific Nessus plugins are identified when useful. For a more in-depth list of how Nessus performs these checks, please review the previous OWASP chapter. • SQL injection (multiple plugins test this) • Cross-site scripting (multiple plugins test this) • Directory traversal vulnerabilities (e.g., #39467, #46194, #46195) • HTTP response splitting/header injection (#39468) • Common application error messages (many different plugins such as #39446, which look for default Tomcat errors) • Testing for common backup script files (#11411) • Include file source code disclosure (many different plugins such as #12245, which reports Java source code) • Insecure HTTP methods (many different plugins such as #12141 for dangerous method detection) • WebDAV (#11424) or FrontPage (#10077) extensions enabled • Default web server installations (#11422) • Testing for diagnostics pages (multiple plugins)
Other Applications	Nessus attempts to test other applications that run on top of an HTTP service with the same tests as a regular web server. Nessus can test RSS feeds, proxy servers, and other technologies. Many unique Nessus plugins are also available to detect and test for vulnerabilities in streaming media such as QuickTime.
Common Services	Nessus detects many different kinds of services that are commonly enabled on Windows, Linux, Solaris, and other types of operating systems and devices.
Wireless Access Points	Nessus plugin #11026 uses a variety of methods to detect wireless access points from the Internet. There are also many specific types of vulnerabilities that Nessus can detect from debug services, web administration pages, and command line protocols.
Backdoors	Although Nessus is not an antivirus solution, Tenable’s Research team has produced a variety of uncredentialed Nessus checks for common backdoors and high profile infections. It also checks for services that may host malicious content such as hostile JavaScript on a web server or streaming malicious executables. A Nessus/Nessus Perimeter Service PCI compliance scan will fail any system that undergoes an ASV scan if it is found to have a known backdoor active.
SSL/TLS	Nessus has extensive coverage for the correct use of encryption for authentication and protection of sensitive data. As required to be performed by an ASV, Nessus performs the

	following SSL/TLS audits: • SSL/TLS detection (multiple checks) • Supported algorithms and key strengths (#10863, #21643, #26928 and #35291) • Detect signature signing algorithms (#10863 and #31705) • SSL certificate validity and expiration (#15901, #42980 and #42981) • SSL certificate common name (#45410 and #45411) • Low entropy Debian keys (#32321) A Nessus/Nessus Perimeter Service PCI compliance scan will fail any scanned system that has non-compliant SSL encryption or certificates.
Remote Access	Nessus includes many different detection types of remote access software and their associated vulnerabilities. Nessus detects many types of security issues with various VPN protocols, VPN technologies, VNC, Microsoft Terminal Server (RDP), a variety of web-based administration suites (e.g., phpMyAdmin), SSH, and Telnet.
Point-of-Sale (POS) Software	At this time, Nessus does not include a specific plugin family for detection of Point-Of-Sale (POS) software or hardware. However, with credentials, Nessus can detect known malware that targets POS systems (e.g., "Dexter").

Vulnerability Reporting

Nessus includes a variety of items in its classification, organization, and reporting of vulnerabilities that make it ideal for PCI DSS testing:

- PCI DSS requires all vulnerability severities to make use of version 2 of the Common Vulnerability Scoring System (CVSS). Tenable has been scoring vulnerabilities detected by Nessus with this standard for several years.
- PCI DSS also requires use of US government standards such as Common Vulnerability Exposure (CVE). Nessus makes use of CVEs for reference wherever possible.
- Any detected vulnerability with a CVSS score of 4 or higher results in an automatic failure of a scanned system.
- Tenable maps all vulnerabilities with a CVSS score of 10 into a "Critical" severity, 7 through 9.9 into a "High" severity, 4 through 6.9 into a "Medium" severity, and lower than 4 into a "Low" severity.

Performing the PCI DSS Assessment

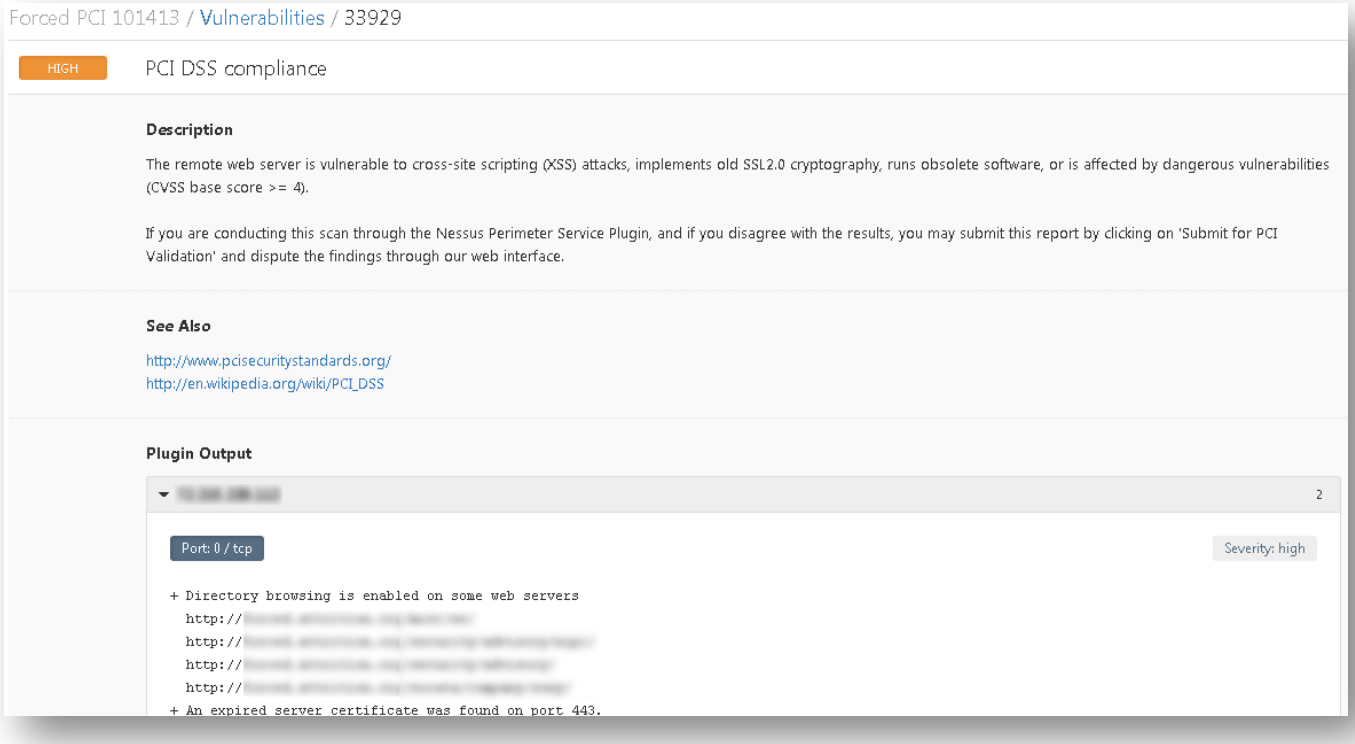
Since most Nessus or SecurityCenter users are not PCI Approved Scanning Vendors, but are likely performing a vulnerability scan to *prepare* for an official assessment, Tenable has designed some flexibility into how Nessus can be configured to perform the scan.

PCI DSS requires many tests to be performed. If you are performing these tests frequently or on a large scale, you may want to only perform a portion of these tests and analyze the results to get a quick picture of your organization's posture.

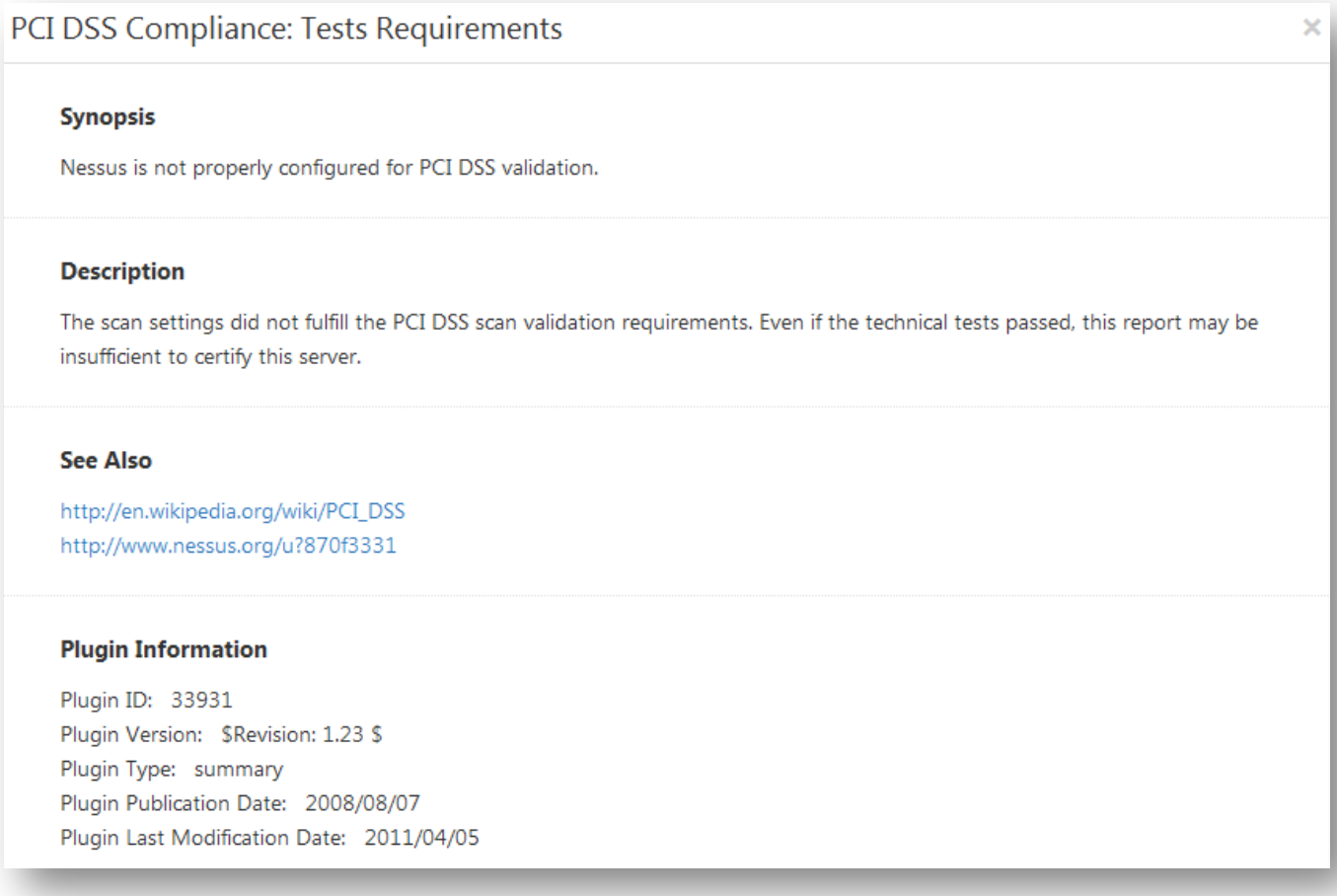
Nessus includes three plugins to assist you in providing a quick overview of your scan results:

- #33929 PCI DSS Compliance
- #33930 PCI DSS Compliance: Passed
- #33931 PCI DSS Compliance: Tests Requirements

When a PCI DSS scan is configured on Nessus, it will automatically collect the results of the scan and report anything that makes a scanned host non-compliant with PCI DSS. This will be reported by plugin #33929. An example is shown below from a host that has been scanned for PCI DSS compliance using Nessus:



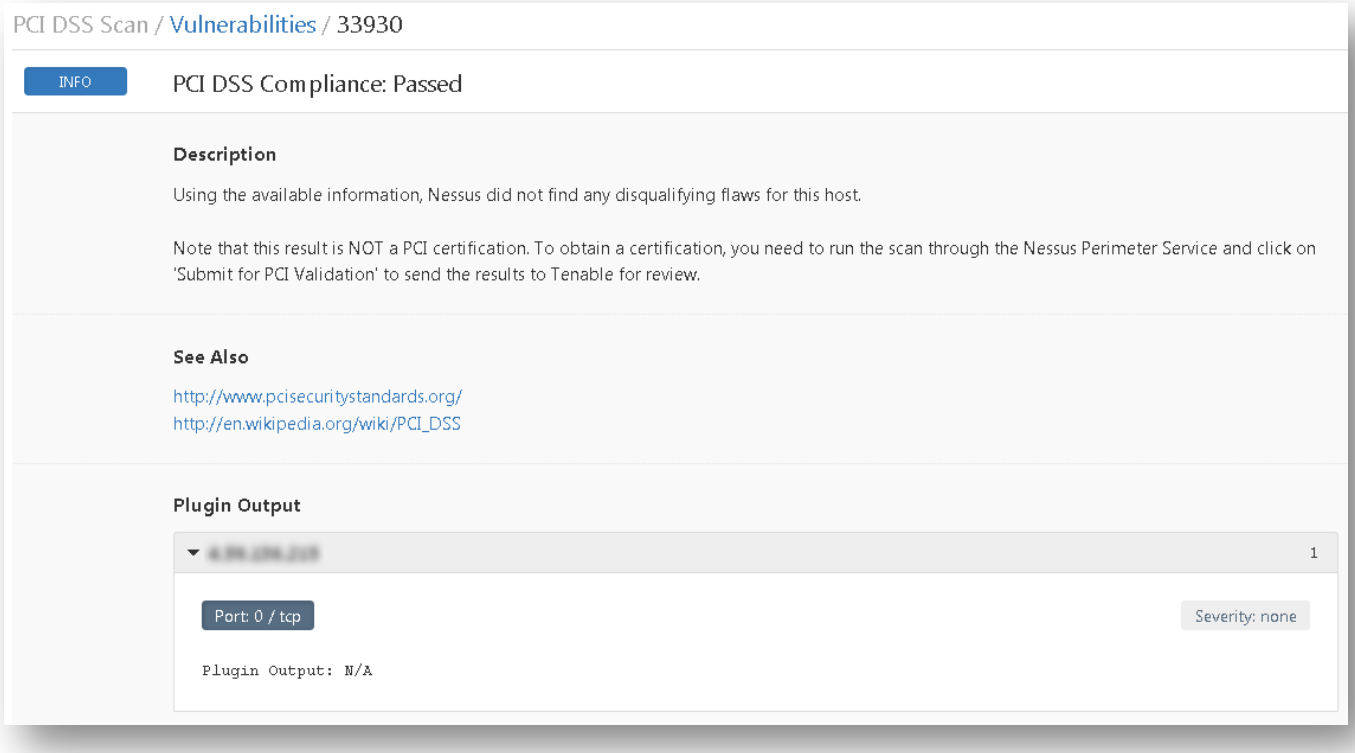
Notice that a summary listing of all non-compliant PCI information records is found in the section titled “Plugin Output” of plugin #33929. For flexibility, you may wish to disable some of Nessus’ checks or settings to increase the speed in which a PCI DSS assessment may be executed. For example, you may want to reduce the number of ports scanned in order to quickly scan port 80 for several dozen web servers. Such a scan would not produce the same sort of test as an actual PCI DSS assessment, but it may find results that still make the systems non-compliant with PCI DSS. Because of this, plugin #33931 inspects the settings of the Nessus scan policy and evaluates them for PCI DSS compliance. An example output is shown below:



This plugin will consider port scan ranges, types of checks run, and many other parameters of the scan policy to ensure that a proper scan is being run to evaluate PCI DSS compliance.

When performing incomplete PCI DSS assessments, the important concept is to realize that if non-compliant results are found, then an organization is deemed non-compliant. If no issues are identified, the system may be compliant, but since a full assessment has not been completed, such a statement would not be accurate.

To drive this point home, plugin #33930 looks at the output of both of the previous plugins. If no PCI DSS compliance issues were found and a valid test was performed, the plugin will report that the server passes both internal and external PCI DSS scanning requirements and is ready for PCI DSS compliance testing from an ASV. An example screen shot is shown below:



PCI DSS 6.5 & 6.6

PCI DSS section 6 is part of the standard's effort to require vendors to maintain a vulnerability management program. Section 6.5 describes specific requirements pertaining to common web application issues and draws content directly from the OWASP project. Section 6.6 describes specific requirements relating to the operation of a web application firewall or using a scanner such as Nessus to perform web application assessments.

OWASP 2013 Top 10 Mapping for PCI DSS 6.5 Requirements

PCI DSS version 3.0 makes specific references to using current industry best practices for vulnerability management such as those referenced in the OWASP Guide. Previously, the PCI recommendations drew heavily from OWASP content published in 2007 and 2010. Since OWASP content from 2013 is available, we provide a mapping instead of a discussion on each PCI DSS 6.5 requirement. In the next section, we will map PCI DSS requirements 6.5.1 through 6.5.9 to specific OWASP 2013 Top 10 risks.

PCI 6.5 Requirements	OWASP 2013 Top 10 Mapping
6.5.1 Injection flaws, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.	2013 OWASP Top 10 – A1 Injection
6.5.2 Buffer overflows	This used to reference A5 of the 2004 OWASP Top 10 but was removed in 2007. Nessus still performs tests for these types of issues.
6.5.3 Insecure cryptographic storage	This used to reference A7 of the 2010 OWASP Top 10 but was removed in 2013. Nessus still performs tests for these types of issues.

6.5.4 Insecure communications	This used to reference A9 of the 2010 OWASP Top 10 but was removed in 2013. Nessus still performs tests for these types of issues.
6.5.5 Improper error handling	This test was dropped from OWASP 2010 and previously pointed to A6 of the 2007 OWASP Top 10. Nessus still performs tests for this sort of issue through several scripts.
6.5.6 All "high risk" vulnerabilities identified in the vulnerability identification process	While not a direct component of the OWASP Top 10, Tenable maps all vulnerabilities with a CVSS score equal to or greater than 7 into a "High" ("Critical" for a CVSS score of 10) severity and Nessus includes these scores in reports.
6.5.7 Cross-site scripting (XSS)	2013 OWASP Top 10 – A3 Cross-Site Scripting
6.5.8 Improper access control	2013 OWASP Top 10 – A4 Insecure Direct Object References, A6 – Sensitive Data Exposure
6.5.9 Cross-site request forgery (CSRF)	2013 OWASP Top 10 – A8 Cross-Site Request Forgery

Performing PCI DSS 6.6 Web Application Vulnerability Assessments with Nessus

PCI DSS version 3.0 references public-facing web applications in section 6.6, which states:

> *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 an automated technical solution that detects and prevents web-based attacks (for example, a web-application firewall) in front of public-facing web applications, to continually check all traffic.*

Section 6.6 does not go into as much detail as section 6.5 with respect to which types of vulnerabilities must be tested. However, it does require multiple items including the following:

Web Application Knowledge

Web application assessments must be performed by a qualified professional. This person must not be a member of the development team and can also be a third party. The person must be proficient in web application security issues as well as the scanning technology that is used.

Generic web application tests report data on the scan itself: network or protocol errors, timeouts, etc. It is possible to check the coverage, the reliability, and the completeness of the tests.

Nessus users who perform web application vulnerability assessments based on PCI DSS 6.6 must be able to demonstrate their proficiency in using Nessus to perform PCI assessments to a PCI QSA.

Areas where a user may be able to gain knowledge of how to perform Nessus web application audits include:

- Performing tests on pre-production web applications with Nessus
- Attempting web application audits on test web applications such as "Damn Vulnerable Linux" or "Mutilidae"
- Attending advanced Nessus training available from Tenable
- Keeping abreast of new Nessus web-based audit techniques by tracking the Tenable blog, Discussion Forums, and Twitter feeds

Manual Verification of Results

A Nessus web application scan of a sensitive site must consider the generated logs and source code of the tested site. This can help verify web application vulnerabilities and also provide feedback in tuning Nessus to perform a more accurate vulnerability scan.

For example, manual source code analysis may help identify hidden forms, applications, and variables that might not be possible to learn in an automated fashion through crawling of the web site.

Additional Web Application Security Monitoring Technologies

Tenable Network Security offers a variety of solutions that either solve or can demonstrate many of the specific security requirements specified by the PCI DSS. For a detailed listing of how Tenable's approach can help demonstrate PCI DSS requirements 1 through 12, please download the "Real Time PCI Compliance Monitoring" paper available from the Whitepapers section of the Tenable website.

In addition to assisting with PCI DSS compliance, there are many unique technologies offered by Tenable that help detect web application insecurities and actual compromises.

Passive Web Site Discovery and Auditing

Performing a full scan of the network to discover every web server is often impractical. However, the larger the network, the more likely it is that a new web server, or perhaps even a new web site (more than one web site may be placed on a single web server) may be placed online.

Tenable's Passive Vulnerability Scanner (PVS) monitors all network traffic and identifies all active web servers and web sites regardless of which port is being used. The PVS allows for continuous discovery of new web servers and web sites and these can be fed back into an active vulnerability assessment process with Nessus.

The PVS will also identify many vulnerabilities associated with these discovered web sites including items such as:

- Expired SSL certificates
- SQL databases exposed to the Internet
- Web applications that display errors from making SQL queries
- Vulnerabilities with underlying web libraries such as PHP
- Web sites that host JavaScript on third-party servers
- Web authentication forms that are not protected by SSL

Real-time Log, Process, and File Integrity Monitoring

Tenable's Log Correlation Engine (LCE) gathers logs and systems events from web servers and operating systems to perform log normalization and facilitate log search.

In addition to log analysis, agents from the LCE can be used to gather system command logs (through process accounting) performed by administrators and the actual web server and databases processes. The LCE agents can also monitor directories vital to the web application for the creation, removal or modification of files. Finally, all of these events can be used as sources of correlation and anomaly detection.

Examples of web security event issues found by the LCE include:

- Detecting modified HTML, PHP, Java, and other types of web application files
- Creating an audit trail of all commands run by administrators and potential web site attackers
- Alerting when your web server processes execute other commands or programs that have not occurred in the past

- Detecting “spikes” in web error logs that indicate web application probing
- Detecting when a remote IP address has caused web errors on more than one of your web servers

In addition, when logs from non-web sources are brought into the LCE, the following types of alerts and correlations can be performed:

- Intrusion detection logs can be used to perform correlation with known vulnerabilities on the web server.
- If the web server is compromised and begins to attack other systems, the LCE can use intrusion detection events to alert on this.
- NetFlow and network session data can be used to watch web site traffic to alert on long web sessions that indicate compromises.
- Firewall, proxy, load balancing, and web application firewall logs can be viewed alongside web logs for greater understanding of security events.

Web Application Configuration Auditing

Tenable’s Research team has developed hundreds of audit policies for Cisco routers, Unix and Linux systems, Windows systems, databases such as Oracle, web servers such as IIS, and web application libraries such as PHP. When considering a “web application”, many organizations attempt a holistic approach and include all devices from the perimeter routers to the back-end database servers.

Using Nessus to perform automated configuration audits of the web application infrastructure allows for continuous and rapid testing to look for changes that may have been introduced by administrators, developers or software upgrades.

Policies can be custom built for any type of web application. Many policies are available for several compliance standards and vendor guides including:

- **CIS Compliance Audit Policies** – CIS certified configuration audit policies for Windows, Solaris, Red Hat, FreeBSD and many other operating systems.
- **Sensitive Content Audit Policies** – Audit policies that look for credit cards, Social Security numbers, and many other types of sensitive data.
- **Configuration Audit Policies** – Audit policies based on CERT, DISA STIG, NSA, GLBA, and HIPAA standards.
- **Microsoft Windows Audit Policies** – Audit policies based on standard Microsoft security templates.
- **Network and Virtualization Infrastructure** – Audits for network equipment (routers, switches, firewalls) and virtualization infrastructure (VMware, Hyper-V, XenServer).
- **Antivirus Audit Policies** – Audit policies designed to allow users to determine if an antivirus package is installed and set to a working state.
- **Virus Detection Audit Policies** – Audit policies that scan for known trojans and rootkits.
- **PCI Audit Policies** – Audit policies to test AIX, HP-UX, Linux, Solaris, and Windows systems for minimum required PCI configuration settings.
- **Tenable Application Audit Policies** – Audit policies that examine hosts to determine if Tenable software applications exist and notifies of the presence and state of these packages.
- **Database Audit Policies** – Audit policies designed to allow users to audit their database configuration using the Nessus Database Compliance Check plugin.

- **Control System Audits** – Audit policies for a wide variety of Control Systems and SCADA applications from Digital Bond.

- **SCAP-based Audit Policies** – Audit policies that perform NIST FDCC/USGCB and DISA STIG SCAP configuration audits. These audit files test for the required settings specified by the DISA STIG SCAP and NIST FDCC/USGCB programs.

Database Activity Monitoring

Gathering logs for the actual SQL queries between web applications and the backend databases is often difficult. If gathering of transaction logs was not built into the application from the start, there may not be an audit trail of SQL queries that can be used for diagnostics, trending or forensic investigation of security attacks.

Tenable's Passive Vulnerability Scanner sniffs many protocols, including SQL queries from MySQL, Oracle, and MS SQL. These logs are sent to the Log Correlation Engine where they can be viewed, normalized, and searched. The Log Correlation Engine can produce several useful reports and types of analysis based on the SQL logs. These include:

- Highlighting all SQL logins and login failures

- Visually displaying all queries, insertions, and other database commands over time

- Identifying spikes in SQL activity based on historic behavior

- Alerting on new types of SQL logs and errors that have not been seen before

- Searching SQL queries to look for common types of SQL injection

If SQL logs are used with other logs from the web server and security infrastructure, the Log Correlation Engine will have access to a tremendous amount of information that is useful for monitoring of web applications.

Appendix 1: Configuring an OWASP Top 10 Scan with Nessus

To configure a Nessus application scan that will test for the OWASP Top 10 issues, follow the steps below:

1. Create a new policy. (Policies -> New Policy)
2. The “Policy Wizard” will appear and provide an option to select “Web Application Tests”. In addition to the information required to do extensive testing of your custom application, this policy will also scan for known web server and published application vulnerabilities.
3. Under “Step 1”, set up a scan as you normally would.
4. Under “Step 2”, select the scan type: either “Less complex (shorter scan time)” or “More in depth (longer scan time)”. Next, set the base path of the web application so that Nessus can mirror it to facilitate testing.
5. Under “Step 3”, select an authentication method from the drop-down menu and provide the required information in the fields below. Note that without authenticating, Nessus cannot thoroughly scan the application.

Additionally, you can find detailed information on the preferences in the Nessus 5.2 HTML5 User Guide:

About Tenable Network Security

Tenable Network Security is relied upon by more than 20,000 organizations, including the entire U.S. Department of Defense and many of the world's largest companies and governments, to stay ahead of emerging vulnerabilities, threats and compliance-related risks. Its Nessus and SecurityCenter solutions continue to set the standard to identify vulnerabilities, prevent attacks and comply with a multitude of regulatory requirements. For more information, please visit www.tenable.com.

GLOBAL HEADQUARTERS

Tenable Network Security
7021 Columbia Gateway Drive
Suite 500
Columbia, MD 21046
410.872.0555
www.tenable.com