Demonstrating Compliance with Nessus Web Application Scans

Performing OWASP and PCI DSS Audits of Web Based Applications

September 27, 2010
(Revision 4)

Ron Gula
Chief Technology Officer

Michel Arboi
Senior Research Engineer

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 Tenable's Nessus vulnerability scanner performing web application audits specific to the following standards:

- OWASP *Top 10*
- PCI DSS

This paper reflects standards described by version 1.2 of the Payment Card Industry Data Security Standard (PCI DSS) requirements, with specific attention given to demonstrating PCI 6.5 and 6.6 compliance requirements. While Tenable focuses on performing web application tests to demonstrate compliance with PCI 6.5, running a web application firewall or performing a source code audit may also 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; rather, they require an external vulnerability scan from an ASV. A majority of these ASVs make use of the Nessus scanner from Tenable. This paper demonstrates how to perform internal testing to be better prepared for certification testing. Nessus' usage for total PCI coverage into configuration audits, antivirus testing and patch testing, as well as Tenable's enterprise network monitoring and logging solutions, are not covered in this paper. They 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 audits, 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 August 2010, searching for the term "CGI Generic" in the list of available Nessus ProfessionalFeed plugins lists the following 32 Nessus checks:

ID	Name	Family
11139	CGI Generic SQL Injection Vulnerability	CGI abuses
39465	CGI Generic Command Execution Vulnerability	CGI abuses
39466	CGI Generic Cross-Site Scripting Vulnerability (quick test)	CGI abuses : XSS
39467	CGI Generic Path Traversal Vulnerability	CGI abuses
39468	CGI Generic Header Injection Vulnerability	CGI abuses
39469	CGI Generic Remote File Inclusion Vulnerability	CGI abuses
39470	CGI Generic Tests Timeout	CGI abuses
40406	CGI Generic Tests HTTP Errors	CGI abuses
42054	CGI Generic SSI Injection Vulnerability	CGI abuses
42055	CGI Generic Format String Vulnerability	CGI abuses
42056	CGI Generic Local File Inclusion Vulnerability	CGI abuses
42423	CGI Generic SSI Injection Vulnerability (HTTP headers)	CGI abuses
42424	CGI Generic SQL Injection (blind)	CGI abuses
42425	CGI Generic Cross-Site Scripting Vulnerability (persistent)	CGI abuses : XSS
42426	CGI Generic SQL Injection Vulnerability (HTTP Cookies)	CGI abuses
42427	CGI Generic SQL Injection Vulnerability (HTTP Headers)	CGI abuses
42479	CGI Generic SQL Injection Vulnerability (2nd pass)	CGI abuses
42872	CGI Generic Local File Inclusion Vulnerability (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 Vulnerability (time-based)	CGI abuses
46193	CGI Generic Cross-Site Scripting Vulnerability (HTTP Headers)	CGI abuses : XSS
46194	CGI Generic Path Traversal Vulnerability (write test)	CGI abuses
46195	CGI Generic Path Traversal Vulnerability (extended test)	CGI abuses
46196	CGI Generic XML Injection	CGI abuses
47830	CGI Generic Injectable Parameter Weakness	CGI abuses
47831	CGI Generic Cross-Site Scripting Vulnerability (extended test)	CGI abuses : XSS
47832	CGI Generic On Site Request Forgery Vulnerability	CGI abuses
47834	CGI Generic Redirection Vulnerability	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

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 provide more information describing how Tenable can help to perform broader compliance analysis:

- Tenable Network Security website : http://www.tenable.com/
- 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 are 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 well as when being managed by Tenable's SecurityCenter. In addition, there is 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 is a proud sponsor of the Open Web Application Security Project (OWASP) and has specifically added technology and checks to the Nessus vulnerability scanner to make it easier to find risks identified by this project.

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

OWASP Top 10 – 2007		OWASP Top 10 – 2010	
A2	Injection Flaws	A1	Injection
A1	Cross-Site Scripting (XSS)	A2	Cross-Site Scripting (XSS)
A7	Broken Authentication and Session Management	A3	Broken Authentication and Session Management
A4	Insecure Direct Object Reference	A4	Insecure Direct Object References
A5	Cross-Site Request Forgery (CSRF)	A5	Cross-Site Request Forgery (CSRF)
	Insecure Configuration Management	A6	Security Misconfiguration
A8	Insecure Cryptographic Storage	A7	Insecure Cryptographic Storage
A10	Failure to Restrict URL Access	A8	Failure to Restrict URL Access
A9	Insecure Communications	A9	Insufficient Transport Layer Protection
	not in Top 10 – 2007	A10	Unvalidated Redirects and Forwards
A3	Malicious File Execution		*dropped from Top 10 – 2010*

A6	Information Leakage and Improper Error Handling		*dropped from Top 10 - 2010*

Each of the OWASP Top 10 risks identified in both the 2010 and 2007 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.

2010 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 system commands to run
- LDAP injection – user-controlled data is proceeded 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 August 2010, 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 two generic tests for command execution. The first performs basic parameter pollution to look for command execution and the second one performs time-based attacks to detect command execution.

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".

2010 OWASP Top 10 – A2 Cross-Site Scripting (XSS)

Cross-site scripting results from having user-submitted data rendered to other users in an unfiltered manner, which can result in executing 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 utilize 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 that results in having the account taken over. XSS attacks have also been used to implement a keylogger or conduct other activities that are not intended by the user.

Tenable has implemented multiple Nessus plugins to focus on the detection of most methods for reflected XSS attacks. Script #42425, "CGI Generic Cross-Site Scripting Vulnerability (persistent)", will also identify stored XSS issues.

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.

2010 OWASP Top 10 – A3 Broken Authentication and Session Management

Web sites that have security issues may permit users to exploit a vulnerability that allows them to impersonate, steal the credentials or impersonate another user on the web

[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".

application. The OWASP project asks seven questions to determine if an application's authentication or session management is potentially vulnerable:

OWASP A3 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. For applications that use the authentication features of Apache or IIS, Tenable offers many different configuration audit policies 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.

2010 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. An authorized attacker may change their 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.

2010 OWASP Top 10 – A5 Cross-Site Request Forgery (CSRF)

This web application weakness leverages image tags, XSS and other techniques to trick an authenticated user to a sensitive 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. 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.
- Five specific tests detect CSRF in known web applications.

2010 OWASP Top 10 – A6 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 A6 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 likely 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.
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 configuration files as well as to test the file integrity of the configuration files to ensure they have not changed.

2010 OWASP Top 10 – A7 Insecure Cryptographic Storage

Encryption is used to store and secure sensitive data. Web applications should be designed so that, even if they are compromised, the attacker can only access limited data. For example, a database that stores passwords "unsalted" but encrypted might be vulnerable to a short-term brute force attack.

Nessus audit policies can be used to search for sensitive data in applications that store data in flat files. Tenable offers audit policies that test for the presence of credit card numbers, customer data and many other types of potentially sensitive information. What is considered insecure or sensitive should be determined by the auditor, but it is extremely useful to have Nessus independently scan a server's file system that is part of a web application.

Additionally, if a database has tables that contain sensitive data, Nessus audit policies can be written to test the database's configuration. Databases such as MySQL, Oracle and MS SQL all contain many different options for storing data securely and controlling access to various tables. Nessus audit policies that test settings specific to your web application database backend security needs can be developed to automate independent verification of your data security settings.

2010 OWASP Top 10 – A8 Failure to Restrict URL Access

Web applications can be complex, have multiple "front ends" to backend systems, have different types of authentication for users and administrators and even have different web application logic for different types of functions. As such, there are frequent cases where a sensitive form or web application that performs maintenance, data analysis, data retrieval or data backup gets inadvertently exposed with no security. OWASP refers to this issue as a failure to restrict access to a URL.

Restricting access can be in the form of preventing direct access, such as with a local firewall, web proxy or router. It can also mean that a sensitive web site needs to have more authentication than what is currently implemented.

Nessus performs several types of audits to facilitate analysis of this OWASP risk:

- Nessus enumerates any page or directory that requires HTTP authentication such as Basic, Digest or NTLM.
- Nessus plugin #44134 named "CGI Generic Unseen Parameters Discovery" has found "hidden" administration pages for our customers.

For complete analysis, a manual inspection must be made as automated scanning cannot differentiate what is an important and exploitable unprotected interface versus what may be harmless to leave exposed.

2010 OWASP Top 10 – A9 Insufficient Transport Layer Protection

Transport layer encryption for sensitive web traffic carrying authentication and private data must be used. This OWASP risk identifies many different types of threats from network traffic monitoring that can be defeated with the proper amount of encryption.

There are many exploit scenarios for poor encryption including:

- Sniffing sensitive network data that is not encrypted such as passwords and credit card data. This may occur between the user and the application or between the web server and database server.
- Secure web sites with unsigned certificates create users that are accustomed to "clicking through" warnings about potential spoofing. When a real phishing attack is launched, the layer of SSL security is partially eroded allowing for the harvesting of passwords and launching browser-based attacks.

The OWASP project outlines five questions that should be asked to help identify if you are at risk. Nessus detects several known bugs in SSL implementations, including weak private keys generated by Debian based Linux.

OWASP A9 Questions	Nessus Audit Technique
SSL protects all authentication related traffic.	Nessus plugin #34850 named "Web Server Uses Basic Authentication Without HTTPS" checks for authenticating web pages being served on non-SSL protected pages.
SSL is used for all resources on all private pages and services.	Nessus helps enumerate hosted content and can be used for manual analysis to identify what is being hosted on non-SSL web sites.

Only strong algorithms are supported.	Nessus includes multiple checks for "weak" SSL ciphers. Nessus credentialed audit policies can test for SSL security registry settings for IIS and can also test SSL security configurations of Apache servers.
All session cookies have their "secure" flag set so the browser never transmits them in the clear.	Nessus includes two related checks for this vulnerability in existing application. First, Nessus check #49218 will verify a cookie has the "secure" flag set. In addition, check #48432 named "Web Application Session Cookies Not Marked HttpOnly" identifies another relevant flag to better protect cookies.
The server certificate is legitimate and properly configured for that server.	Nessus includes multiple checks for SSL certificates including reporting on the SSL certificate information, if and when it has expired, if it matches the hostname and more.

2010 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 a new 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 accounts 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.

PCI DSS Web Based Audits

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

- SQL injection vulnerabilities
- Cross-Site Scripting vulnerabilities
- Directory Traversal vulnerabilities
- HTTP response splitting vulnerabilities
- Out of date or insecure SSL encryption
- Any vulnerabilities with a CVSS score greater than 4
- Information leakage issues

This section of the paper reviews how such a scan can be performed using Nessus or Nessus when being managed by SecurityCenter.

These PCI scanning and minimum vulnerability requirements should not be confused with other PCI 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, it is not a replacement for ASVs any more than any other security tool is.

Required ASV Scanning Components

PCI requires that all authorized 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 IDS/IPS	There are many different types of "interference" that can be performed by an IDS/IPS. 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 IDS/IPS will appear to Nessus as if a firewall is interfering with the scan. An IDS/IPS that silently drops packets will not be detected by Nessus. If the IPS 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 an auditor can place their Nessus scanner on the outside of the network, they may be able to determine if an IDS/IPS is blocking their scans and save time during a paid-for PCI DSS audit 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, 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 operating systems and will use this information to fail a PCI audit.
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 SQL 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. In relation to a PCI DSS audit, any directory browsing should result in an automatic failure of PCI DSS compliance. Nessus scans for this and reflects these results in the PCI report.
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 fail a server for PCI DSS compliance if the system performs a zone transfer that is covered by Nessus check #10595.
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 PCI DSS audit from an ASV, Nessus performs the

	following checks and will fail a server if any of them 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 RealAudio.
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 including Conficker. It also checks for services that may host malicious content such as hostile JavaScript on a web server or streaming malicious executables. Nessus' PCI report will fail any system scanned that has a known backdoor active on it.
SSL/TLS	Nessus has extensive coverage for the correct use of encryption

	for authentication and protection of sensitive data. Please read the previous OWASP chapter (section A9) for a more in-depth discussion. 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) Nessus' PCI report 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 Software	Nessus does not include a specific plugin family for detection of Point-Of-Sale (POS) software or hardware.

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 or 4 or higher results in an automatic failure of a scanned system.
- Tenable maps all vulnerabilities with a CVSS score equal to or greater than 7 into a "High" severity, 4 though 6.9 into a "Medium" severity and lower than 4 into a "Low" severity.

Performing the PCI DSS Audit

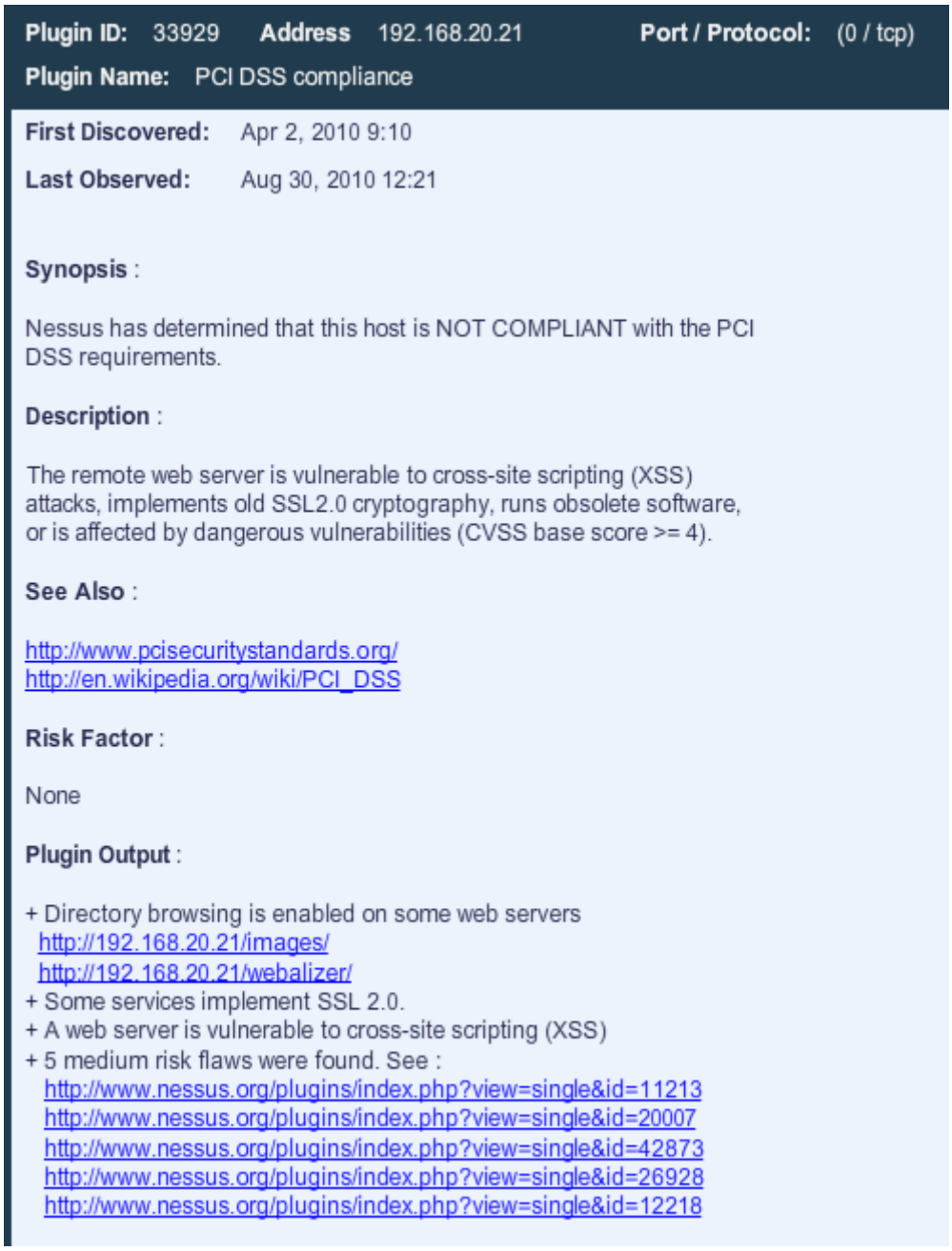
Since most Nessus ProfessionalFeed or SecurtiyCenter users are not authorized scanning vendors, but are likely performing a vulnerability scan to *prepare* for an official audit, Tenable has designed some flexibility into how Nessus can be configured to perform the scan.

PCI DSS requires many tests to be performed. If you are performing these tests often 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 audited for PCI DSS compliance through SecurityCenter:



Plugin ID: 33929 **Address** 192.168.20.21 **Port / Protocol:** (0 / tcp)

Plugin Name: PCI DSS compliance

First Discovered: Apr 2, 2010 9:10

Last Observed: Aug 30, 2010 12:21

Synopsis :

Nessus has determined that this host is NOT COMPLIANT with the PCI DSS requirements.

Description :

The remote web server is vulnerable to cross-site scripting (XSS) attacks, implements old SSL2.0 cryptography, runs obsolete software, or is affected by dangerous vulnerabilities (CVSS base score >= 4).

See Also :

http://www.pcisecuritystandards.org/
http://en.wikipedia.org/wiki/PCI_DSS

Risk Factor :

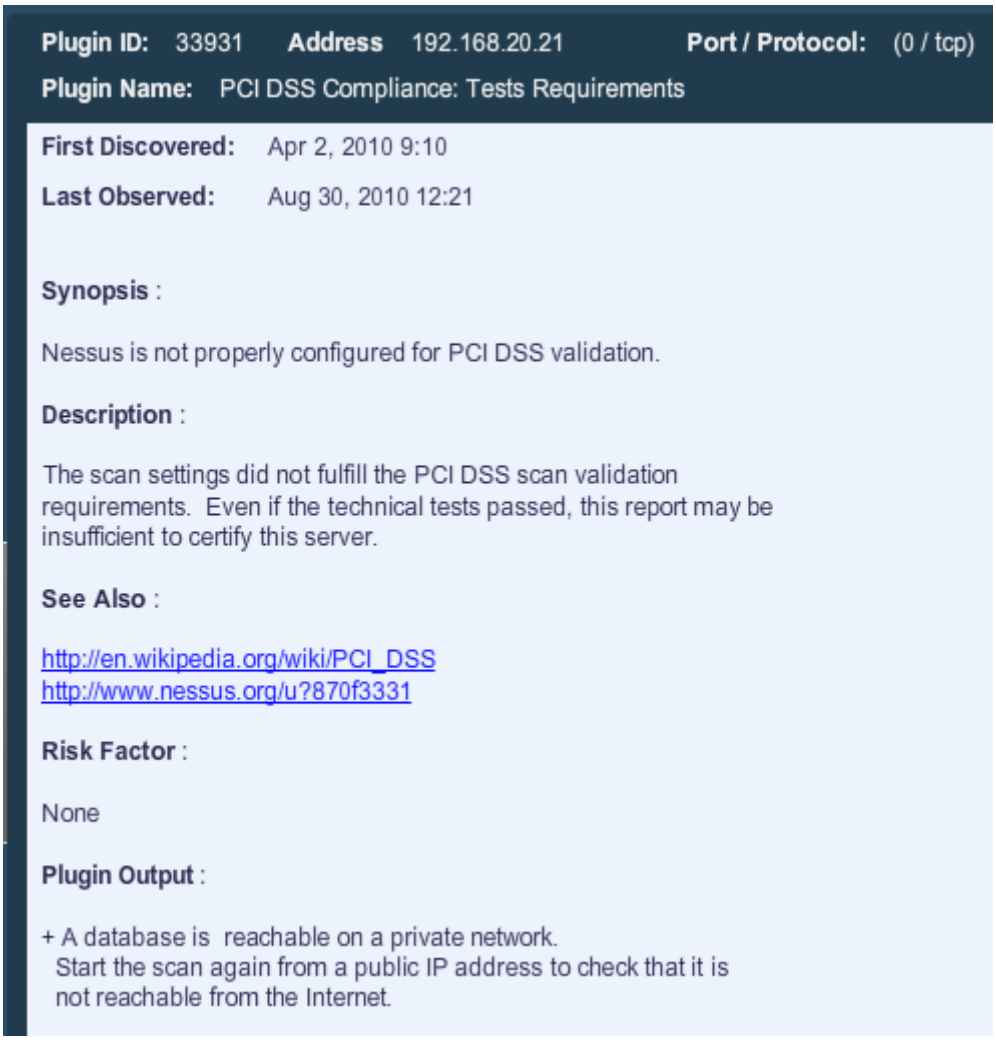
None

Plugin Output :

+ Directory browsing is enabled on some web servers
  http://192.168.20.21/images/
  http://192.168.20.21/webalizer/
+ Some services implement SSL 2.0.
+ A web server is vulnerable to cross-site scripting (XSS)
+ 5 medium risk flaws were found. See :
  http://www.nessus.org/plugins/index.php?view=single&id=11213
  http://www.nessus.org/plugins/index.php?view=single&id=20007
  http://www.nessus.org/plugins/index.php?view=single&id=42873
  http://www.nessus.org/plugins/index.php?view=single&id=26928
  http://www.nessus.org/plugins/index.php?view=single&id=12218

Notice that a summary of all non-compliant PCI information records the output of plugin #33929.

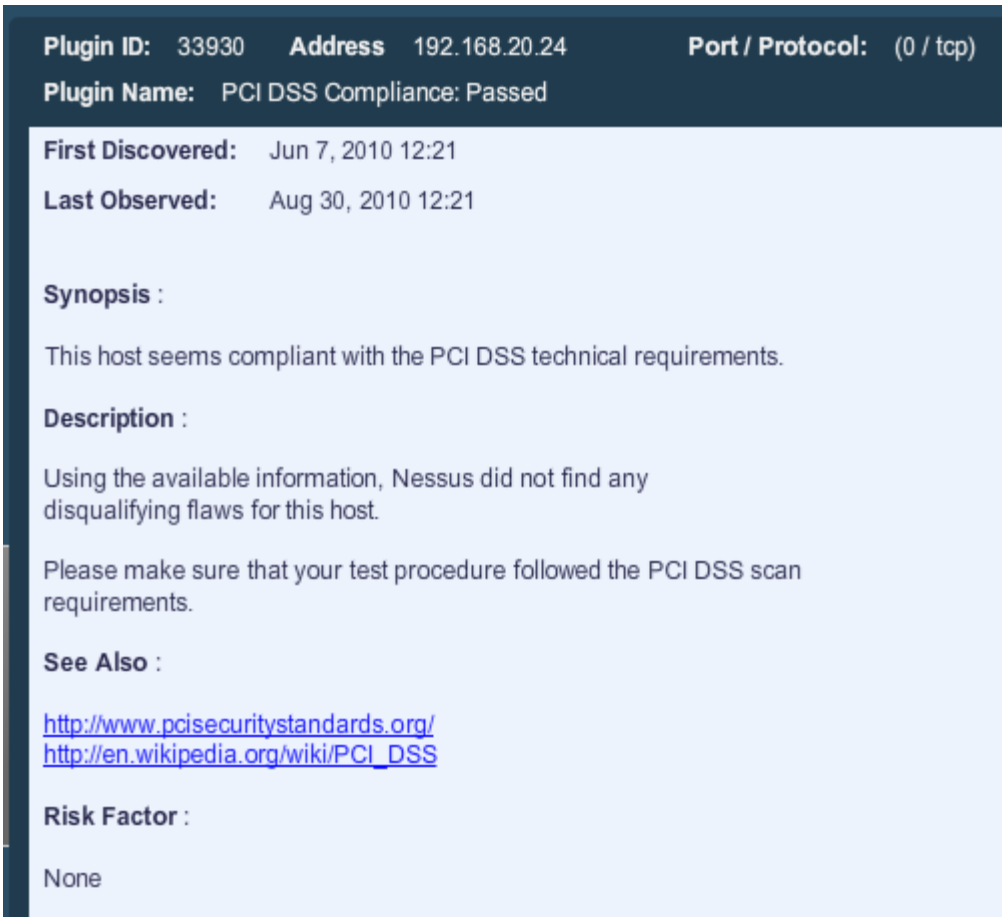
For flexibility, you may wish to disable some of Nessus' checks or settings to increase the speed in which a PCI DSS audit may be executed. For example, you may want to reduce the number of ports scanned in order to quickly scan port 80 for several dozen web servers. Such a scan would not produce the same sort of test as an actual PCI DSS audit, 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 audits, the important concept is to realize that if non-compliant results are found, then the system is deemed non-compliant. If no issues are identified, the system may still be compliant, but since a full audit has not been completed, such a statement would not be accurate.

To drive this point home, plugin #33931 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 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 2010 Top 10 Mapping for PCI DSS 6.5 Requirements

When PCI DSS version 1.2 was released, the latest 2010 OWASP Top 10 risks were not published. However, PCI DSS 1.2 makes specific references to using the latest OWASP content if and when it is available. Previously, the PCI recommendations drew heavily from OWASP content published in 2007. Since new OWASP content from 2010 is now 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 6.5.10 to specific OWASP 2010 Top 10 risks.

PCI 6.5 Requirements	OWASP 2010 Top 10 Mapping
6.5.1 Cross-site scripting (XSS)	2010 OWASP Top 10 – A2 Cross-site scripting

6.5.2 Injection flaws	2010 OWASP Top 10 – A1 Injection
6.5.3 Malicious file execution	This used to reference A3 of the 2007 OWASP Top 10 and is no longer required. Nessus still performs these types of tests as part of the local and remote file inclusion tests, command execution, SSI injection and many others.
6.5.4 Insecure direct object references	2010 OWASP Top 10 – A4 Insecure Direct Object References
6.5.5 Cross-site request forgery (CSRF)	2010 OWASP Top 10 – A5 Cross-site request forgery
6.5.6 Information leakage and improper handling	This test was dropped from OWASP 2010 and used to point to A6 of the 2007 OWASP Top 10. Nessus still performs tests for this sort of issue through several scripts.
6.5.7 Broken authentication and session management	2010 OWASP Top 10 – A3 Broken Authentication and Session Management
6.5.8 Insecure cryptographic storage	2010 OWASP Top 10 – A7 Insecure Cryptographic Storage
6.5.9 Insecure communications	2010 OWASP Top 10 – A9 Insufficient Transport Layer Protection
6.5.10 Failure to restrict URL access	2010 OWASP Top 10 – A8 Failure to Restrict URL Access

Performing PCI DSS 6.6 Web Application Vulnerability Assessments with Nessus

When PCI DSS 1.2 was released, it simply stated that organizations had to run a web application firewall or make use of:

> *Manual or automated vulnerability security assessment tools or methods that review and/or scan for application vulnerabilities can be used to satisfy this requirement.*

Since the initial release of PCI DSS 1.2, there has been an informational supplement for 6.6 that clarifies how application reviews (option 1) must be performed. It also clarified the use of web application firewalls (option 2), but this is out of scope for this document.

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 knowledgeable person. This person must not be a member of the development team and can also be a third party. The person must not only have knowledge of web application security issues, they must be proficient with 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.

When using Nessus to perform style web application vulnerability assessments based on PCI DSS 6.6, not only must the user be knowledgeable on how to perform these audits, they must be willing to demonstrate this knowledge to a PCI auditor.

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 web application scan of a sensitive site must consider the generated logs and source code of the tested site when using Nessus to perform the vulnerability audit. 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 solve many of the security and compliance requirements specified by PCI. For a detailed listing of how Tenable's Unified Security Monitoring approach can help demonstrate PCI DSS requirements 1 through 12, please contact Tenable to request a copy of our "Real Time PCI Compliance Monitoring" paper.

Beyond 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. It performs log normalization and facilitates 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:

- Detection of 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 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:

- Microsoft IIS best practices
- Apache best practices
- Center for Internet Security Windows servers
- Center for Internet Security Windows IIS web servers
- DISA STIG hardening of IIS web servers
- DISA STIG hardening of Windows servers
- DISA STIG hardening of Red Hat servers
- Oracle hardening based on DISA STIG requirements
- Oracle hardening based on Center for Internet Security requirements
- MySQL hardening based on Center for Internet Security requirements
- MS SQL hardening based on Center for Internet Security requirements
- Cisco router configuration settings on Center for Internet Security requirements

Database Activity Monitoring

Often, gathering logs for the actual SQL queries between web applications and the backend databases is 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 do 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, inserts 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.

About Tenable Network Security

Tenable, headquartered in Columbia, Md., USA, is the world leader in Unified Security Monitoring. Tenable provides agent-less solutions for continuous monitoring of vulnerabilities, configurations, data leakage, log analysis and compromise detection. For more information, please visit us at http://www.tenable.com/.

TENABLE Network Security, Inc.
7063 Columbia Gateway Drive
Suite 100
Columbia, MD 21046
TEL: 410-872-0555
http://www.tenable.com/