



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Leadership Essentials for Managers (Cybersecurity Leadership 512)"
at <http://www.giac.org/registration/gslc>

Auditing using Vulnerability tools to identify today's threats Business Performance

GIAC (GSLC) Gold Certification

Author: Carlos Vazquez, donq423@gmail.com

Advisor: Richard Carbone

Accepted: November 12, 2014

Abstract

The majority of IT staff is likely to say that systems are safe from intrusion and many may be at a loss to offer evidence to back that up. A common characteristic of many high profile security breaches is that both IT and business leadership believe that their systems are secure from intrusion when, in fact, they are not. This is often called a “false sense of security”.

Building a robust vulnerability management program removes such subjectivity from security assessments and gives an organization's leadership quantitative insight into the effectiveness of security controls. This research project will demonstrate the importance of a good robust program that combines scanning technology with management practices designed to prioritize and remediate high-risk organization's vulnerabilities before they are exploited by an attacker. A business lacking a vulnerability management program cannot identify high-risk vulnerabilities in either its systems or the effectiveness of the security controls it has implemented, increasing the risk of losing valuable proprietary information and its business reputation.

1. Introduction

A properly implemented vulnerability management program represents a key element in an organization's information security program by providing a business oriented approach to risk mitigation. This program provides a way to assess the potential business impact and probability of threats and risks to an organization's information infrastructure before those events occur.

Scanning facilitates the securing of the network but it can also be used for malicious purposes by identifying threats and risks to its systems. The use of these tools can help to identify and fix these weaknesses before an intruder uses them against an organization.

Vulnerability scanning generally refers to the scanning of systems that are connected to the Internet but can also refer to system audits on internal networks that are not connected to the Internet in order to assess the threat of rogue software or internal threat by malicious employees in an enterprise.

Scanner tools can also help to validate the inventory of devices on the corporate network. An inventory includes the device type, operating system version and patch level, hardware configurations and other applicable system information. This data is useful in security management and tracking (Rouse, n.d.).

A vulnerability scanner has capabilities that include the following:

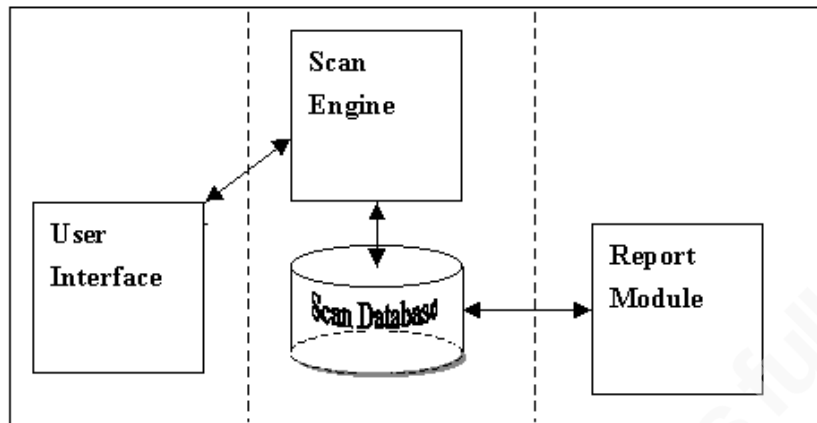
- Keeping an up-to-date database of vulnerabilities;
- Detection of genuine vulnerabilities without an excessive number of false positives;
- Ability to conduct multiple scans at the same time;
- Ability to perform trend analyses and create clear reports of the results;
- Provide recommendations for effective countermeasures to eliminate discovered vulnerabilities.

If security holes are detected by the scanner, a vulnerability disclosure may be required. The person or organization that discovers the vulnerability, or a responsible industry body such as the Computer Emergency Readiness Team (CERT), may make the disclosure, sometimes after alerting the vendor and allowing them a certain amount of time to remedy or mitigate the problem (Rouse, n.d.).

2. Components of vulnerability scanners

There are four different components that constitute a vulnerability scanner:

- Engine Scanner - performs security checks according to its installed plug-ins, identifying system information, and vulnerabilities (Snyder, n.d.). It can scan more than one host at a time and compare the results against known vulnerabilities.
- Database - stores vulnerability information, scan results, and other data used by the scanner. The number of available plug-ins and the updating frequency of plug-ins will be different depending on the corresponding vendor. Each plug-in may have not only the test case itself, but also a vulnerability description, a Common Vulnerabilities and Exposures (CVE) identifier (Common Vulnerabilities and Exposures, 2014). Some scanners with an "auto-update" feature can download and install the latest set of plug-ins to the database automatically.
- Report Module - provides different levels of scan result reporting such as comprehensive technical reports with suggested remedies for system administrators, summary reports for security managers, and high-level graph and trend reports for corporate executives' leadership.
- User Interface - allows the administrator to operate the scanner. It may be either a Graphical User Interface (GUI), or just a command line interface. Figure 1 shows the basic components of vulnerability scanners.



Components of Scanner

Figure 1: Components of vulnerability scanner

When enterprise networks are widely distributed, Distributed Network Scanners are used. They are composed of remote scanning agents, a plug-in update mechanism for those agents, and a centralized management point (SANS, 2002). These scanners are capable of assessing vulnerabilities across multiple physically dispersed networks from one centralized management location, which can then distribute updates to scanning agents, change settings in all scan engines, and schedule testing activities across the entire enterprise. Scan results are then collected from all remote scanning agents to be placed into the central database for analysis and reporting.

Most organizations today use traditional active scanning to discovery vulnerabilities, which requires a remote scan of each network attached device. However, this approach to vulnerability assessment is often limited by:

- Limitation of access – in some cases the IT staff will require authorization from management to scan systems that are critical to the organization because the scanner tool may impact their availability and productivity.
- Distribution of assets – other assets and services are difficult to identify or access due to location, such as cloud or mobile devices. When a scan is performed over WAN, the length of time can be affected by incoming and outgoing traffic going over different links.

- Information overload - vulnerability scanners drown IT security teams in data and are notorious for producing an extensive and boring table of vulnerabilities with no network context, risk prioritization, or actionable fixes.
- Not actionable - scanner reports prioritize vulnerabilities based on asset importance and established vulnerability severity ranking, typically based on the Common Vulnerability Scoring System (CVSS) scoring tables. This methodology does not consider network context and can lead administrators to fix non-threatening vulnerabilities and ignore the critical ones (Cohen, 2014).

2.1. Types of vulnerability scanners

There are two main types of vulnerability scanners. The first is the Network-based scanner. This type of scanner attempts to look for vulnerabilities from the outside-in. It is launched from a remote system like a laptop or desktop with no type of user or administrator access on the network. Network-based scanners look for exploitable remote vulnerabilities such as open ports, buffer overflows, and other exploits. There are many such scanners today. For example, Retina which is a commercial network-based scanner works using a client-server architecture. The Common Hacking Attack Methods (CHAM) is a Retina tool which is quite effective and valuable to scan and identify zero-day as well as known vulnerabilities on the network (Mehta, 2007).

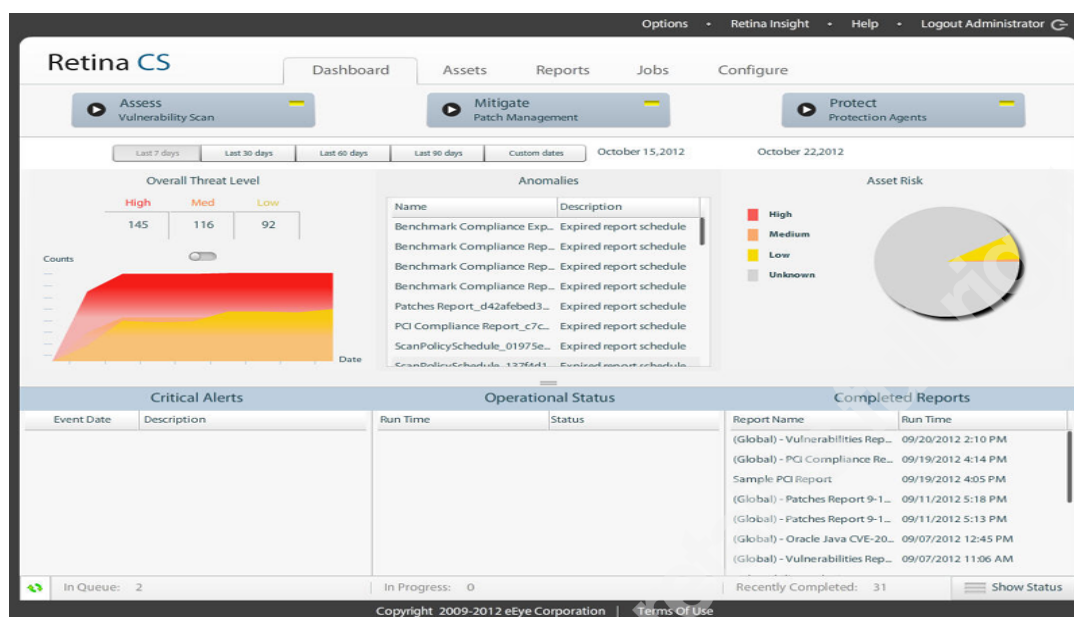


Figure 2: Retina Scanner GUI Dashboard

Nessus is one of the most popular and capable vulnerability network-based scanners, particularly for UNIX systems. It is constantly updated, currently with more than 46,000 plugins. Key features include remote and local (authenticated) security checks, client/server architecture with a web-based interface, and an embedded scripting language for writing new plugins or understanding the existing ones (Nessus, n.d.). The open-source version of Nessus was forked by a group of users who still develop it under the name of OpenVAS.

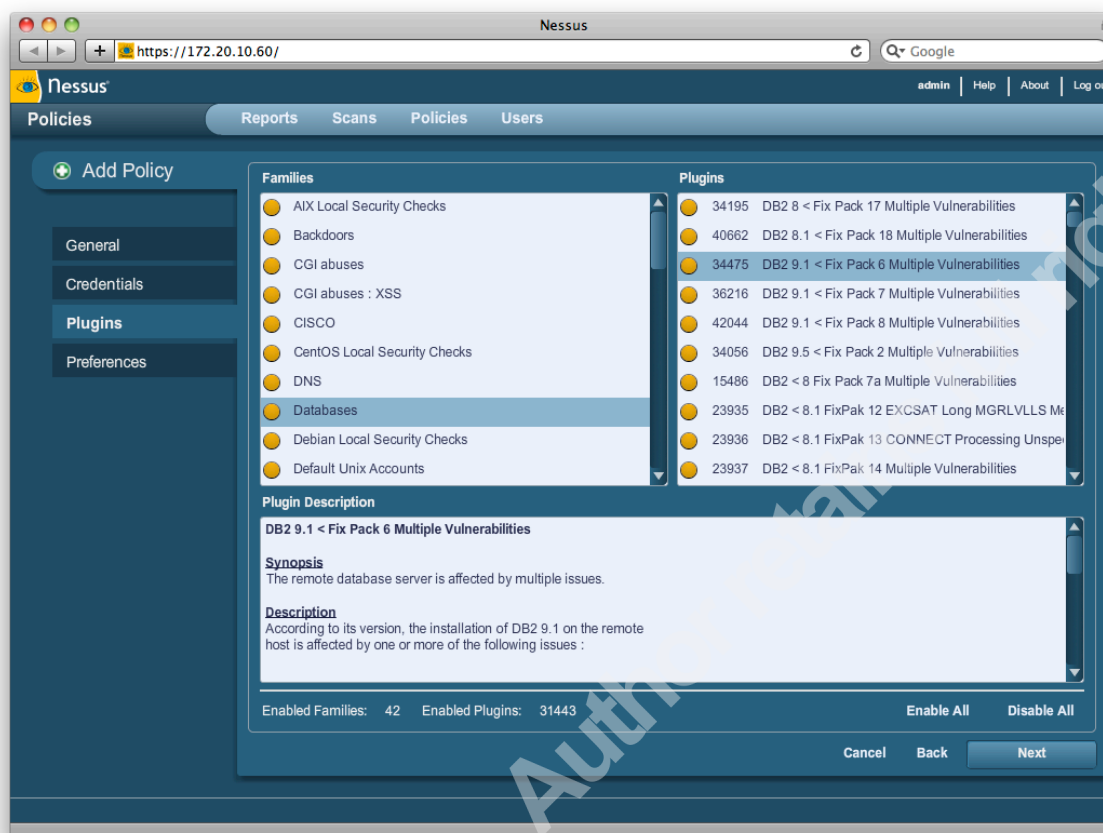


Figure 3: Nessus Plug-in Dashboard

SAINT (Security Administrator's Integrated Network Tool) is another example of a network-based scanner. Like Nessus, it used to be free and open source but is now a commercial product (Saint, n.d.). It is available in a variety of configurations, including software for installation on a UNIX or Linux computer, SaintBox (an appliance), and WebSaint which is a remote scanning service that allows the organization to initiate scanning over the web and then log back in to view results. Generally, SAINT is considered to be a workhorse vulnerability assessment tool, highly scalable and true to its mature vulnerability assessment roots, while presenting an easy-to-use and configurable user environment (Stephenson, 2006).

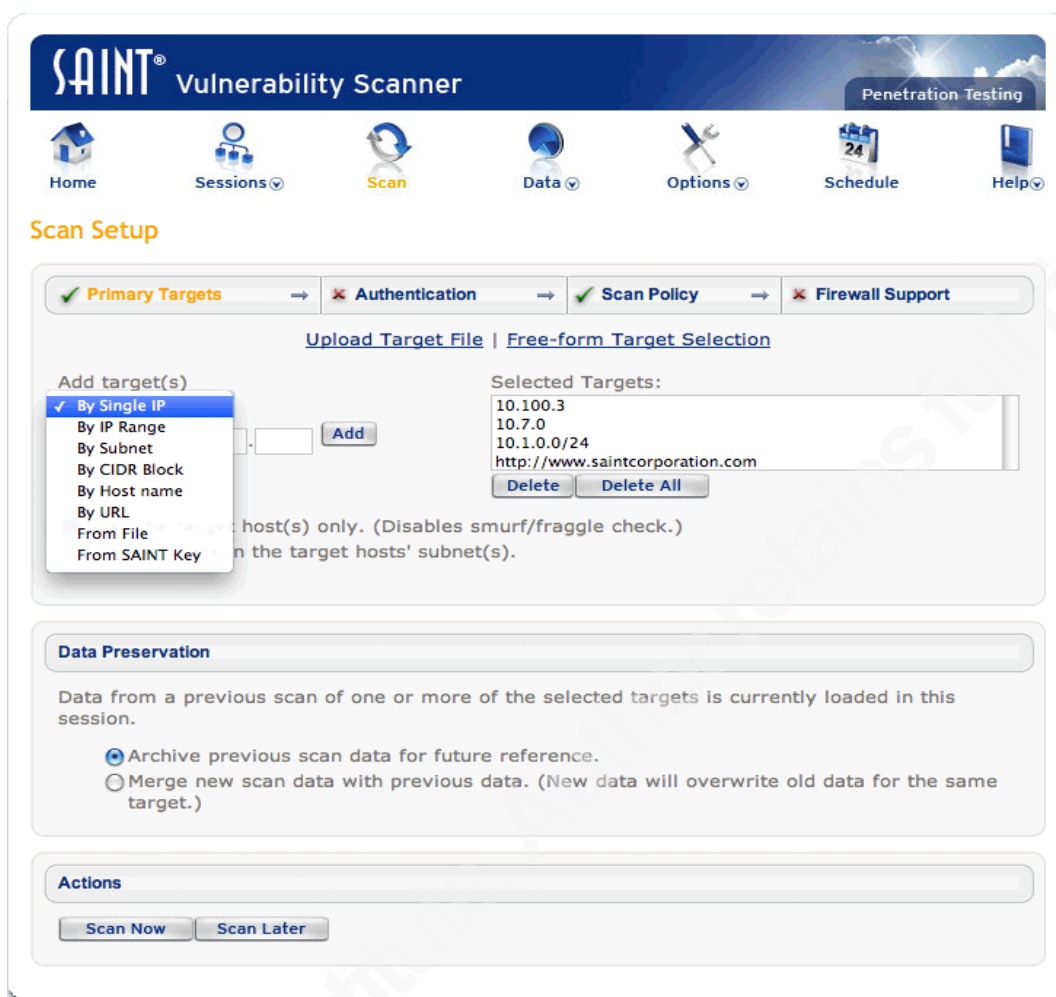


Figure 4: SAINT initial configuration

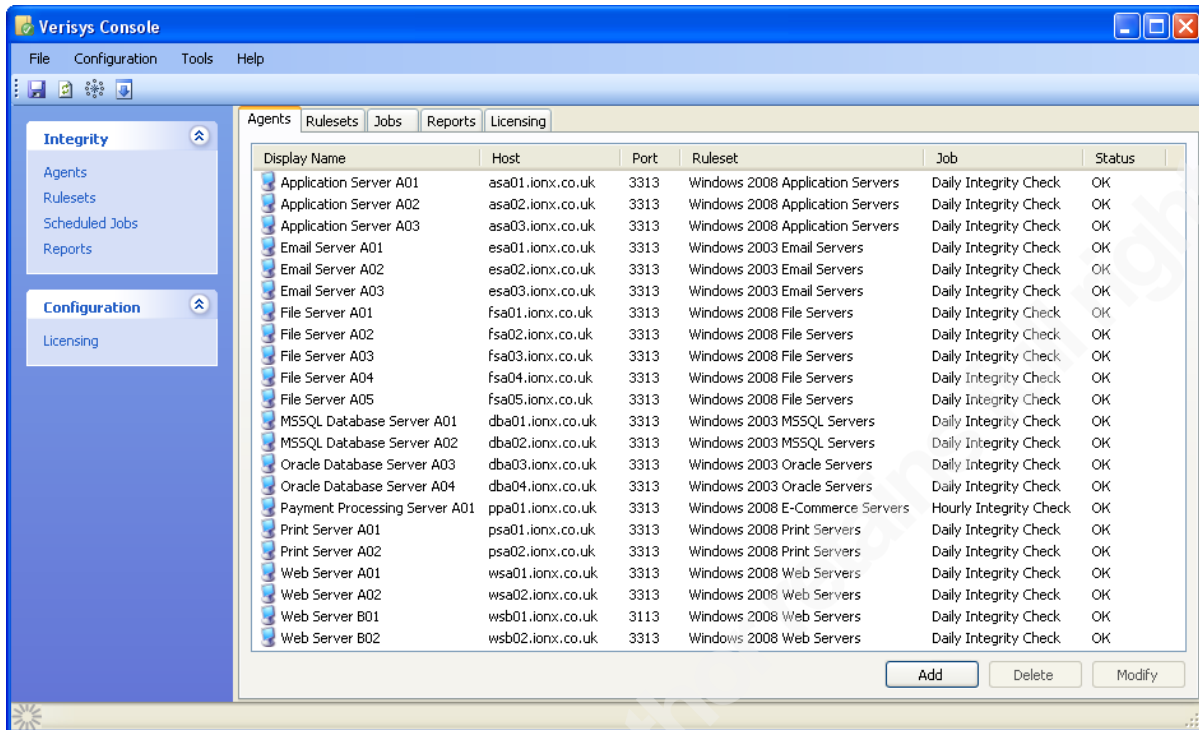
The second type of scanner is the host-based scanner. This type of scanner looks at the host from the inside-out. Host-based scanners generally require an agent to be installed on the server. The agent then reports back to the administrator any vulnerability it finds. Host-based scanners search for problems such as weak file permissions, poor password policy, lack of security auditing, and so on. For example, Microsoft Baseline Security Analyzer (MBSA) is an easy-to-use tool designed for the IT professional that helps small and medium-sized businesses determine their security state in accordance with Microsoft security recommendations and offers specific remediation guidance. Built on the Windows Update Agent and Microsoft Update infrastructure, MBSA ensures consistency with other Microsoft management products including Microsoft Update (MU), Windows Server Update Services (WSUS), Systems Management

Server (SMS) and Microsoft Operations Manager (MOM). MBSA on average scans over 3 million computers each week (MBSA, n.d.).



Figure 5: Microsoft Baseline Security Analyzer scan results

Verisys is another example of host-based scanner. It's a commercial file integrity monitoring solution for Windows based systems. It consists of both client and server components which are the Verisys Console and Agent respectively. Agents are deployed to servers or workstations to be monitored while the console provides a GUI to manage remote agents (Verisys, 2014).



The screenshot shows the Verisys Console application window. On the left is a navigation pane with 'Integrity' selected, containing links for Agents, Rulesets, Scheduled Jobs, and Reports. Below it is a 'Configuration' section with a link for Licensing. The main area displays a table of agents under the 'Agents' tab. The table has columns for Display Name, Host, Port, Ruleset, Job, and Status. All agents listed have a status of 'OK'.

Display Name	Host	Port	Ruleset	Job	Status
Application Server A01	asa01.ionx.co.uk	3313	Windows 2008 Application Servers	Daily Integrity Check	OK
Application Server A02	asa02.ionx.co.uk	3313	Windows 2008 Application Servers	Daily Integrity Check	OK
Application Server A03	asa03.ionx.co.uk	3313	Windows 2008 Application Servers	Daily Integrity Check	OK
Email Server A01	esa01.ionx.co.uk	3313	Windows 2003 Email Servers	Daily Integrity Check	OK
Email Server A02	esa02.ionx.co.uk	3313	Windows 2003 Email Servers	Daily Integrity Check	OK
Email Server A03	esa03.ionx.co.uk	3313	Windows 2003 Email Servers	Daily Integrity Check	OK
File Server A01	fsa01.ionx.co.uk	3313	Windows 2008 File Servers	Daily Integrity Check	OK
File Server A02	fsa02.ionx.co.uk	3313	Windows 2008 File Servers	Daily Integrity Check	OK
File Server A03	fsa03.ionx.co.uk	3313	Windows 2008 File Servers	Daily Integrity Check	OK
File Server A04	fsa04.ionx.co.uk	3313	Windows 2008 File Servers	Daily Integrity Check	OK
File Server A05	fsa05.ionx.co.uk	3313	Windows 2008 File Servers	Daily Integrity Check	OK
MSSQL Database Server A01	dba01.ionx.co.uk	3313	Windows 2003 MSSQL Servers	Daily Integrity Check	OK
MSSQL Database Server A02	dba02.ionx.co.uk	3313	Windows 2003 MSSQL Servers	Daily Integrity Check	OK
Oracle Database Server A03	dba03.ionx.co.uk	3313	Windows 2003 Oracle Servers	Daily Integrity Check	OK
Oracle Database Server A04	dba04.ionx.co.uk	3313	Windows 2003 Oracle Servers	Daily Integrity Check	OK
Payment Processing Server A01	ppa01.ionx.co.uk	3313	Windows 2008 E-Commerce Servers	Hourly Integrity Check	OK
Print Server A01	psa01.ionx.co.uk	3313	Windows 2008 Print Servers	Daily Integrity Check	OK
Print Server A02	psa02.ionx.co.uk	3313	Windows 2008 Print Servers	Daily Integrity Check	OK
Web Server A01	wsa01.ionx.co.uk	3313	Windows 2008 Web Servers	Daily Integrity Check	OK
Web Server A02	wsa02.ionx.co.uk	3313	Windows 2008 Web Servers	Daily Integrity Check	OK
Web Server B01	wsb01.ionx.co.uk	3113	Windows 2008 Web Servers	Daily Integrity Check	OK
Web Server B02	wsb02.ionx.co.uk	3313	Windows 2008 Web Servers	Daily Integrity Check	OK

Figure 6: Verisys Console Report

It is also a good practice to utilize both a network and host based scanner when testing critical systems. Network-based scanners have many options for dangerous tests, such as Denial of Service (DoS). On the other hand, host-based scanners usually require an agent be added on the system being tested which could introduce a problem on the target system if the scanner is incorrectly configured or if the agent conflicts with an application or service on the target system. It's always best to test the host-based scanner on non-production systems prior to deployment on a live production environment.

3. What is a vulnerability?

ISO 27005 defines vulnerability as “A weakness of an asset or group of assets that can be exploited by one or more threats where an asset is anything that can has value to the organization, its business operations and their continuity, including information resources that support the organization's mission” (Vulnerability, 2014).

Vulnerabilities can be separated into two major categories. The ones related to errors made by programmers in writing the code for the software and the ones related to misconfiguration of the application software that makes systems less secure.

In the hands of an intruder, vulnerability scanners become a means of finding victims and determining those victims' weak points. This is like an undercover intelligence operative who infiltrates the opposition's supposedly secure location and gathers information that can be used to launch a full scale attack (Vacca, 2009).

It is important to mention that the first scanners were designed as hacking applications, but this is a case in which the hacker's weapons have been appropriated and used to defend against them.

3.1. The importance of a good Vulnerability assessment program

A vulnerability assessment program is essential for any business as there are many tasks that need to be set. In many businesses, when it comes to network security, some organizations stop at patch management and antivirus software. A lack of knowledge could be a problem when checking in for configurations, known issues in third-party applications, as well as potentially troublesome hardware that in their default configuration could be harmful to the network's security (Carabott, 2011). These processes are what constitute a vulnerability assessment.

The following is a list of general configuration problems that might be found in an operating system installation:

- Needless open shares;
- Operating Systems not patched regularly;
- Proprietary software not updated frequently;
- Unused user accounts;
- Needless open ports;
- Rogue devices;
- Unsafe script configurations;
- Servers permitting use of dangerous protocols;

- Wrong permissions on important system files;
- Running of unnecessary, potentially dangerous services.

Beyond these misconfigurations, when running a vulnerability assessment on the network, the IT staff may find several security issues with a large range of software and hardware including:

- Default passwords on devices;
- Needless services running on some devices;
- Running web services that contain known vulnerabilities;
- Unsafe applications such as peer-to-peer applications;
- Third-party applications that are a vulnerability to known exploits.

Some vulnerability scanners will also look for signs of known malware based on the computer's behavior rather than actually scanning the files for known malware signatures (Carabott, 2011). In some cases, this approach can help uncover issues that an antivirus might miss, especially if that malware were being protected by a rootkit.

It is important to note that each of the aforementioned issues can put the network's security at risk even if it is fully patched. Networks are dynamic by nature; they change continually (Clark, 1990). A vulnerability assessment should be set to run continuously and notify the administrator every time change is detected to maximize network security protection.

The following are the four main parts that constitute the vulnerability assessment cycle (Katsicas, 2009):

- Detect - conduct a vulnerability assessment and report findings to management;
- Correct - advise the corrective actions that should be taken to resolve the findings and to maintain the continuity of business operations;
- Prevent - set the preventive actions that should be followed to avoid any future threats against existing vulnerabilities;

- Assess Risks - present to management the risk-based assessment report that includes the possible business impact from the identified assessment results. Figure 7 illustrate the vulnerability assessment cycle process.



Figure 7: Vulnerability Assessment Cycle (Katsicas, 2009)

4. Examples of real data security breaches

It is difficult to estimate the cost of each cyber security breach which can range greatly. Larger cyber security incidents in the past have resulted in many of dollars millions of breached data that include compromised personal, intellectual, and proprietary information.

The majority of security breaches will result in additional costs for impacted organizations (Phneah, 2012). These costs range from breach containment, crisis management, investigations and forensics, customer compensation, damaged system replacements, lawsuits and other penalties. Some notable examples of data security breaches include the following:

Business Name: *Heartland Payment Systems*

Date: March 2008

Impact: 134 million credit cards exposed through SQL injection to install spyware on Heartland's data systems

A federal grand jury indicted Albert Gonzalez and two unnamed Russian accomplices in 2009. Gonzalez, a Cuban-American, was alleged to have masterminded the international

operation that stole credit and debit cards. In March 2010, he was sentenced to 20 years in federal prison. The vulnerability to SQL injection was well understood and security analysts had warned retailers about it for several years (Taylor, 2012). Yet, the continuing vulnerability of many Web-facing applications made SQL injection the most common form of attack against Web sites at the time.

Business Name: *Google/other Silicon Valley companies*

Date: Mid-2009

Impact: Stolen intellectual property

In an act of industrial espionage, the Chinese government launched a massive and unprecedented attack against Google, Yahoo, and dozens of other Silicon Valley companies. The Chinese hackers exploited a weakness in an old version of Internet Explorer to gain access to Google's internal network. It was first announced that China was trying to gather information on Chinese human rights activists. It's not known exactly what data was stolen from the American companies, but Google admitted that some of its intellectual property had been stolen and that it would soon cease operations in China (Taylor, 2012). For users, the urgent message is that those who haven't recently updated their web browser should do so immediately.

Business Name: **Drupal.Org**

Date: 2013

Impact: Exposure of password and personal information

The popular open-source content management system Drupal reset the passwords of users on its Drupal.org website following a data security breach of its servers. Drupal is the platform behind hundreds of thousands of blogs and websites.

The attackers targeted vulnerabilities in the third-party software installed on the Drupal.org server infrastructure, the company said. Exposed data included usernames, email addresses, country information and hashed passwords (Westervelt, 2013). The breach was announced in May and possibly could have impacted nearly a million account holders. The company said it updated its security measures and hardened its Apache Web servers following the breach.

Business Name: Multiple Companies**Date:** 2014**Impact:** Heartbleed Vulnerability; Exposure of Password

The existence of the bug was made public by security firm Codenomicon in April, although it operated undetected for almost two years. Heartbleed affected about 17% of the Internet's secure web servers making passwords vulnerable to theft—information that was normally protected by SSL/TLS encryption.

A massive number of companies were affected including Amazon, Pinterest, Reddit, Tumblr, Airbnb, Wordpress, and Wattpad. Users on each site were advised to change their passwords, while companies were advised to patch their copy of OpenSSL to fix the problem. Operating systems like Android 4.1.1 were also discovered to be vulnerable. The industry mobilized one of its biggest responses ever to a data breach by creating the Core Infrastructure Initiative, a multi-million dollar project to fund critical elements of the web's infrastructure. Backed by companies like Amazon, Dell, Facebook, Google, and Microsoft the funding will help lead developers on various projects and pay for security audits and software development (McCartney, 2014).

5. Running an effective Vulnerability Management Program

Running recurrent vulnerability scans is no longer sufficiently effective for an organization to maintain an effective security posture. To ensure it can proactively detect and respond effectively to security threats, an organization needs to implement a long-term vulnerability management program which is integrated with other key security practices. This allows vulnerabilities to be detected early so that other processes, such as patch management, protect the organization from a potential breach. Any new vulnerability introduced as the result of newly discovered software bugs, new devices added to the network, or changes to systems will go undetected until the next scan (Tenable, n.d.). If there are long gaps between scans, then those affected systems are at risk until those vulnerabilities are identified and addressed.

In addition, having random vulnerability scans can result in large number of vulnerabilities to be addressed after each scan. In some cases the total volume of vulnerabilities discovered as a result of a scan can discourage the IT staff performing the remediation action.

Organizations should employ the following steps to improve the vulnerability management processes (Perradeau, 2009):

- Get executive management support for identifying and remediating vulnerabilities within the organization's tolerance for risk;
- Obtain a complete inventory of all assets and their vulnerabilities;
- Rank remediation efforts according to business risks;
- Remediate vulnerabilities by delivering planned work projects to IT management;
- Continually update asset discovery, vulnerability testing, and remediation processes;
- Use automated patch management and vulnerability discovery.

A good start is by establishing a solid vulnerability management policy. The policy creation and management for an organizations starts at the top and requires executive oversight to ensure systematic implementation.

Here are some key points to consider: Policies establish the nature of controls used to ensure security, such as standard configurations for all security devices and applications including antivirus, firewall, and intrusion detection system (IDS). IT security staff should produce a template with a short list of configurations and features so that policy makers can understand their options for security controls. Policies and controls apply to servers, network services, applications, and endpoints. Policy makers need to determine the business vulnerability impact of each asset. Prioritization weighs the business risks and importance of each asset, which affects the importance and achievement order of vulnerability remediation.

An effective Vulnerability Management policy should also require system owners to address vulnerabilities that are identified on systems they own. This policy should have clearly defined timetables for how long a system owner has to address any such vulnerability. For example, the policy may require vulnerabilities which pose a very high technical risk to be addressed within a five day window, a vulnerability which poses a high technical risk be addressed in a two week window, and a vulnerability which poses a medium technical risk be addressed within a month. Unless these policies are created before a security assessment is performed, IT staff could be fighting with system owners, because they are not addressing identified vulnerabilities in a reasonable amount of time (McCully, 2013).

To reach systems properly, the auditor should obtain a complete list of all network segments used throughout the organization, such as corporate wired and wireless networks, production networks, backup or administration networks, transit networks, testing networks, and remote offices. Each of these networks must be identified and documented.

The networks also should be included in an architecture diagram that shows network interconnections as well as perimeter security devices, such as routers, firewalls, and intrusion detection systems (Romanosky, 2006). This diagram will allow management to understand how vulnerabilities found in one network may impact the security of assets in another network.

Once a network inventory is obtained, all IT assets connected to each network segment should be scanned or monitored periodically for vulnerabilities. These assets include devices such as application servers, security devices, networking devices, and printers.

The scanner tools should be scheduled to run daily, monthly, or quarterly based on the needs, risks, or capability of the organization. Monitoring refers to software agents installed on IT assets that report host configuration information. It also refers to network devices that continuously listen to network traffic and report, or optionally block, malicious traffic that may exploit a vulnerability. These devices also are useful for identifying rogue or previously unknown IT assets. They are considered a preventive security control because of their ability to block attacks before they cause loss.

Organizations should validate the results from the vulnerability monitoring and scanning process. Although the complexity and accuracy of vulnerability scanning and monitoring devices are generally good, they always have limitations. Errors can occur in the form of false positives or false negatives. A false positive is a vulnerability that has been reported but does not exist, because the detection mechanism was in error. A false negative occurs when vulnerability exists, but the detection system failed to identify it (Romanosky, 2006).

The main purpose of the Risk Assessment Plan is to determine the likelihood that some event will occur. Once vulnerability data have been acquired, the organization must be able to determine the actual risk they cause. While a full risk management project is generally not necessary, a basic risk assessment is. Given the large number of vulnerabilities discovered with each scan, it is likely an organization will be performing a large number of risk assessments.

Therefore, organizations must have a well-defined procedure to measure risks that can be applied quickly and accurately. Note that the presence of any vulnerability does not always end in remediation efforts, and the organization may choose to accept the risk posed by the vulnerability. For example, when existing security controls sufficiently reduce the likelihood of a successful attack or when a potentially targeted asset is of little or no value (Romanosky, 2006). In these cases, the risk acceptance should be documented and approved to avoid reassessing the same finding later.

The organization should prioritize the remediation of vulnerabilities according to the criticality of the vulnerable asset, the likelihood or frequency that an attack will occur, and the effort required to implement the fix. Therefore, auditors will compare the actual risk posed to the organization with the cost to implement the fix and prioritize the risk based on its cost-effectiveness. The organization also may want to examine the causes of past security incidents and prioritize accordingly. The findings that result from the analysis will determine the nature and quantity of the work that will follow.

Risk acceptance will also be considered in cases in which an organization's decision is not to address a specific vulnerability. When there is certainty or a strong doubt that a remediation measure would break functionality and result in down time, management may decide that the risk presented by the vulnerability is preferable to the likelihood of technical difficulties arising from the fixing of that vulnerability. Therefore, it would not be significant to spend effort and funds addressing such a security hole.

The success of vulnerability management will be achieved with a continuous assessment effort. The best vulnerability management program should be proactive and look for weaknesses and configuration errors before they are exploited by real threats (Vitale, 2012). Figure 8 illustrates an example of a vulnerability process model that can be applicable to any organization.

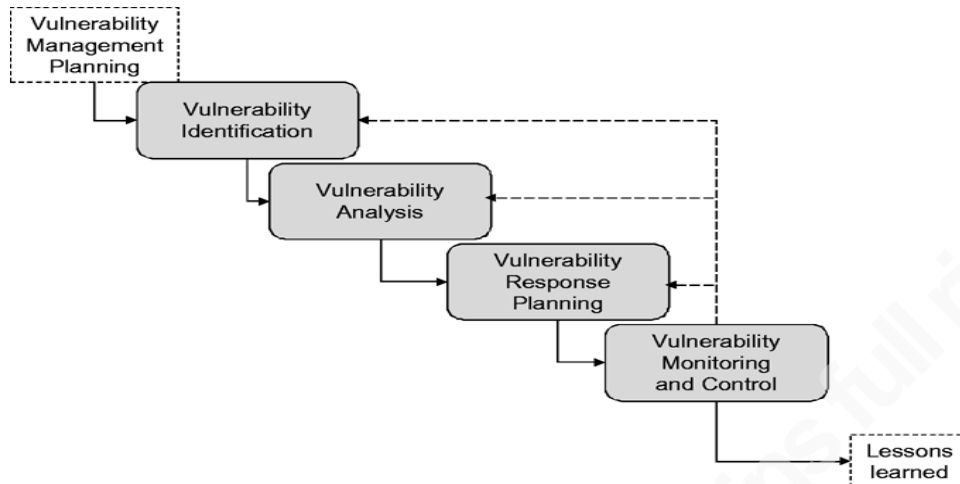


Figure 8: Example of Vulnerability Process

Vulnerability management systems test network resources for the presence of known vulnerabilities (Ogren, 2004). This produces a record per scan of the number of vulnerabilities discovered, their severity, and the asset classes affected.

It is also a good business practice to add vulnerability management and configuration management processes to close vulnerabilities before exploits arrive.

It always good to conduct at least once a year audits by a third party to assess the organization's security policy implementation and measure the security team's performance.

Vulnerability management delivers the fundamental metrics that management needs to evaluate the organization's network security program. Best practices use their metrics to assess the success or failure of various efforts to improve performance (Ogren, 2004).

6. Conclusion

The benefits of vulnerability management program are countless, but unless IT security staff goes through the process, management and leadership won't have the plan in place to protect the business. It is a continual process that monitors the organization's effectiveness to identify, classify, remediate, and mitigate vulnerabilities. Proactively managing system and network vulnerabilities will reduce or eliminate the potential for attacks.

Vulnerability management helps to prevent security exploits rather than trying to learn from the aftermath. Understanding and applying the major steps involved in the vulnerability management process will help to secure organizations. It provides management with detailed instructions to implement the procedures and processes, timelines, organizational and individual responsibilities, and actions in response to vulnerability exploitation. Critical systems and infrastructure components are identified for prioritization to ensure current security issues within the environment are identified and evaluated using the proper vulnerability tools to facilitate a risk management approach. Such checks along with proper implementation planning allow the network to run free of problems and any breach. Daily checks also ensure that the security of the system is up-to-date and continually resistant to any security issue.

Organizations that chose to adopt an ongoing vulnerability management program will prove to be much more resilient and much less likely to understand their network risks than those organizations that chose for an easier path. Equally important is ensuring the vulnerability management process is integrated tightly with other processes that complement and enhance each other's capabilities. In particular, is critical to any organization, the ability to detect new assets on the network and to quickly scan for vulnerabilities and threats. Continuous vulnerability management is fast becoming a necessity for all organizations to ensure the security of their systems.

7. References

- Carabott, E. (2011). *Why You Need to Run a Vulnerability Assessment*. Retrieved July 15, 2014 from <http://www.gfi.com/blog/vulnerability-assessment/>
- Clark, David. (1990). *The Changing Nature of Computer Networks*. Retrieved July 20, 2014 from <http://groups.csail.mit.edu/ana/Publications/DDC-Changing-Nature-of-Computer-Networks-June1990.pdf/>
- Cohen, G. (2014). *Best Practices for Vulnerability Management*. Retrieved July 20, 2014 from <http://insights.wired.com/profiles/blogs/best-practices-for-vulnerabilitymanagement#axzz3C9tVdpt5>
- Common Vulnerabilities and Exposures. (2014). In *Wikipedia, the free encyclopedia*. Retrieved July 20, 2014 from http://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures
- Katsicas, Sokratis K. (2009). "35". In Vacca, John. *Computer and Information Security Handbook*. Morgan Kaufmann Publications. Elsevier Inc. p. 605. ISBN 978-0-12-374354-1.
- Nessus. (n.d). *Nessus*. Retrieved July 10, 2014 from <http://sectools.org/tool/nessus/>
- McCartney, Jennifer. (2014). *MBSA*. Retrieved August 14, 2014 from <http://www.techradar.com/news/software/security-software/the-top-10-data-breaches-of-the-past-12-months-1248890>
- MBSA; (n.d). *MBSA*. Retrieved July 10, 2014 from <http://sectools.org/tool/mbsa/>
- McCully, G. (2013). *Components of an Effective Vulnerability Management Program*. Retrieved July 20, 2014 from <http://blog.securestate.com/components-of-an-effective-vulnerability-management-program/>
- Ogren, E. (2004). *Dynamic Best Practices of Vulnerability Management*. Retrieved July 20, 2014 from <https://www.qualys.com/docs/yankee-whitepaper.pdf>
- Perradeau, Eric. (2009). *Q&A: Vulnerability management*. Interview with Mirko Zorz. Retrieved 28 July 2013 from <http://www.net-security.org/article.php?id=1282>
- Phneah, E. (2012). *Losses from security breaches becoming significant for firms*. Retrieved August 02, 2014 from <http://www.zdnet.com/losses-from-security-breaches-becoming-significant-for-firms-2062305073/>
- Rita, Mehta. (2007). *Top 10 Vulnerability Scanners That Can Help You Assess Vulnerability of Your Network*. Retrieved August 10, 2014 from <http://www.geekhowtos.com/top-10-vulnerability-scanners>
- Romanosky, S. (2006). *Managing and Auditing IT Vulnerabilities*. Retrieved July 20, 2014 from http://www.theiia.org/bookstore/downloads/freetomembers/0_1021.dl_gtag6.pdf
- Rouse, M; (n.d.). *Vulnerability Scanner*. Retrieved July 10, 2014 from

Carlos Vazquez, donq423@gmail.com

- <http://searchsoftwarequality.techtarget.com/definition/vulnerability-scanner>
- Saint. (n.d). *Saint*. Retrieved July 10, 2014 from <http://sectools.org/tool/saint/>
- SANS Institute InfoSec Reading Room. (2002). *Distributed scan model for Enterprise-Wide Network Vulnerability Assessment*. Retrieved August 10, 2014 from <http://www.sans.org/reading-room/whitepapers/auditing/distributed-scan-model-enterprise-wide-network-vulnerability-assessment-74>
- Snyder, Joel. (n.d.). *Testing and comparing vulnerability analysis tools*. Retrieved July 15, 2014 from <http://searchsecurity.techtarget.com/Testing-and-comparing-vulnerability-analysis-tools>
- Stephenson, Peter. (2006). *SAINT Scanner*. Retrieved July 10, 2014 from <http://www.scmagazine.com/saint-scanner/review/3/>
- Taylor, Armerding. (2012). *The 15 worst data security breaches of the 21st Century*. Retrieved July 25, 2014 from <http://www.csoonline.com/article/2130877/data-protection/the-15-worst-data-security-breaches-of-the-21st-century.html>
- Tenable Network Security. (n.d.). *Implementing an Effective Vulnerability Management Program*. Retrieved August 12, 2014 from http://static.tenable.com/whitepapers/Vulnerability_Management_Program.pdf
- Vacca, J.R. (2009). *Managing Information Security*. (2nded.). Waltham, MA: Syngress.
- Verisys. (2014). In *Wikipedia, the free encyclopedia*. Retrieved August 12, 2014 from <http://en.wikipedia.org/wiki/Verisys>
- Vitale, D. (2012). *Create and implement a vulnerability management program*. Retrieved August 14, 2014 from <http://dougvitale.wordpress.com/2012/07/11/create-and-implement-a-vulnerability-management-program/>
- Vulnerability. (2014). In *Wikipedia, the free encyclopedia*. Retrieved July 20, 2014 from http://en.wikipedia.org/wiki/Vulnerability_%28computing%29
- Westervelt, R. (2013). *Top 10 Security Breaches Of 2013*. Retrieved July 22, 2014 from <http://www.crn.com/slide-shows/security/240165003/top-10-security-breaches-of-2013.htm/pgno/0/4>