TASK 1 - NETWORK VULNERABlLlTY ASSESSMENTS

lNTRODUCTlON TO PROJECT:

The project titled “Network Vulnerability Assessment” was created with the aim of finding out what kind of vulnerabilities the company faces, as a way to not only resolve them but act as a proactive means towards avoiding future recurrences of vulnerabilities.

As the security team, it is imperative to protect the company’s systems from unauthorized personnel however with the continuous innovation of technology, staying on top of new software’s and mitigation measures to safeguard our systems is imperative and hereby acts as the motivation and rationale behind the project.

A company system or object is considered to be vulnerable when it has a weakness in its system that results in it being susceptible to attacks such as denial of service (DoS) or incidences of unauthorized access by third parties. lf not attended to immediately through active defence mechanisms and preventative methods, these vulnerabilities can pave the way for cybercriminals to exploit them and result in the company system being jeopardized and the Confidentiality, lntegrity and Availability (ClA triad) not being maintained. Given that ensuring systems maintain the ClA triad, vulnerability assessments are necessary.

ln addition, A Company should maintain its operational integrity by complying to Cybersecurity regulations and industry compliance standards. The systems, software’s and methodology adopted must be in compliance with the NlST, PCl and GDPR. As a result, scanning for vulnerabilities and areas in need of patching is necessary to be in accordance with the standards we are required to implement.

The following project aims to discover, protect and prevent future and current vulnerabilities by first discovering them using the company lP address and then identifying methods to address them and safeguard them for future purposes. Through a use of well trusted Vulnerability Scanner: Nessus and the use of Nmap, the project has adapted various sources to ensure that all vulnerabilities have been discovered. By doing so, A Company is able to maintain sensitive data, ensure only authorized personnel have access to data they require, ensure that company is complying with industry standards and maintaining integrity throughout the organization.

NETWORK VULNERABILlTY TESTNG:

Vulnerability testing systematically evaluates, reviews and analyses an organizations network infrastructure by finding vulnerabilities and loopholes that may jeopardize the company's security or be a method used for cyberattacks. The strength of a company's network security is determined by vulnerability testing and hereby will determine the ability of a company to maintain business continuity, protection of sensitive data, compliance and network privacy. Without network vulnerability testing, it is impossible for a company to manage its vulnerabilities, due to the fact that it cannot begin its management process without identifying what areas require to be managed more strategically.

ASSESSMENT METHODOLODY

The project was conducted through the use of one Vulnerability scanner, and manually through the command Nmap on Kali Linux. The rationale behind using various sources and methods was to ensure a comprehensive and wide range of vulnerabilities to be detected. lt was also to ensure a lack of overreliance on one source, but to implement various sources to increase accuracy in results and findings.

TOOLS:

The primary goal of Network Reconnaissance was performed using the following tools:

Nessus: Nessus is a vulnerability scanner powered by Tenable that seeks to help identify potential vulnerabilities within a system, out of compliance settings and misconfigurations that may be used by exploits for malicious purposes.

Nmap: Nmap known as “Network Mapper” is a tool that can be used on Linux as an open-source tool for Network discovery, security auditing, discovering hosts and operating systems. Nmap allows network admins to find which devices are running on their network, discover open ports and services, and detect vulnerabilities

The tools were used for overall network reconnaissance and vulnerability scanning. Nmap was used to understand the network architecture of the company as well as understanding attack surfaces on the network including open ports and vulnerabilities. Nessus was used for vulnerabilities within the company network such as software's that are not in compliance with industry standards, potential attacks and CVE vulnerabilities according to NlST.

COMPLIANCE AND REGULATORY STANDARDS:

The assessment and project adhered to specific industry compliance standards to ensure operational integrity of the company.

ISO 27000: The ISO 27000 series is a number of best practices to help organizations improve their information security. This standard is implemented when dealing with data breaches and serves to act as a guideline in defences against these breaches for effective security.

NIST CSF 8001-171: Under this compliance, it is imperative to identify and address vulnerabilities. According to the NIST guideline, it helps with vulnerability management, as well as the security and privacy controls for organizations. It serves as a framework in the testing period.

VULNERABILITY CLASSIFICATIONS

The results that were outputted by the vulnerability scanner: Nessus were categorized according to the National Vulnerability Database Common Vulnerability Scoring system (CVSS) through five score metrics: Critical, High, Medium, Low or Informational.

Critical: These are vulnerabilities with a CVSS score of 9.0 to 10.0, that indicate they can be easily exploited by an attacker and system can be compromised.

High: Vulnerabilities with a CVSS score of 7.0 to 8.9, that indicate local users can gain privileges that can allow unauthenticated remote users to view resources or cause a denial of service.

Medium: Vulnerabilities with a CVSS score of 4.0 to 6.9, that indicate flaws that may be difficult for third parties to exploit but are cause for concern as they can still lead to compromise.

Low: Vulnerabilities with CVSS score of 0.1 to 3.9, that indicate vulnerabilities that if exploited may cause either no adverse effect or minimal adverse consequences.

ASSESSMENT FINDINGS

Through the use of two sources, Nessus identified a total of sixteen Vulnerabilities with one being” High” and three scored a CVSS of “Medium”.

On the other hand, vulnerability scanning on Nmap revealed two vulnerabilities that were both categorized as “Medium”.

Below are the vulnerabilities found that are non-informational and found from the various sources. Evidence of the collated vulnerabilities can be referenced to at the end of the document.

1. CVE-2016-2183

Name: SSL Medium Strength Cipher Suites Supported (SWEET32)

Severity: High

CVSS Score: 7.5

Detail: The Sweet32 attack is a vulnerability that can occur through the use of some SSL Cyphers that are weak in design and offer less projection against attacks. The attack makes use of these older versions of the SSL Cyphers used in common protocols such as TLS and OpenVPN, in order for remote users to obtain plaintext data.

2. CWE 327

Name: TLS Version 1.0 Protocol Detection

Severity: Medium

CVSS Score: 6.5

Detail: The current system accepts the use of the TLS 1.0. This version relies on the SHA-1 hash of messages exchanged which is not secure. This vulnerability allows an attacker to execute a downgrade attack on the handshake, compromising security far more than contemporary standards deem acceptable.

3. CVE-2019-20372

Name: nginx < 1.17.7 lnformation Disclosure

Severity: Medium

CVSS Score: 5.3

Detail: These files contain crucial server settings, including listening ports and server names. The current nginx within the system of 1.7.7 allows HTTP request smuggling which allows an attacker to read unauthorized web pages, hereby compromising the security of the system.

Plugin #51192

Name: SSL Certificate Cannot be trusted

Severity: Medium

CVSS Score: 6.5

Detail: This vulnerability occurs when the certificate is signed by an unknown authority hereby meaning it is impossible to verify its integrity. The current system is hereby using an SSL certificate that cannot be trusted. This is a vulnerability because without a trusted SSL certificate, it can lead to man in the middle exploits given its difficult to authenticate and verify the web server with the use of an untrusted certificate.

Cross domain and client access policies Severity: Likely vulnerable (Medium)

CVSS Score: 6.5

Detail: The vulnerability found within the system is due to overly permissive configurations that can pave the way for web clients and third parties to commit Cross-Site Forgery attacks and unauthorized access by third parties to sensitive data. This hereby means that the current system can be exploited and may result in the confidentiality of the system being compromised.

4. CVE-2005-3299

Name: phpMyAdmin 2.6.4

Severity: Medium

CVSS Score: 5.0

Detail: The current system has a PHP file inclusion vulnerability which is a web vulnerability and security flaw that allows unauthorized users to access files, provide download functionality and look for information. This vulnerability allowing remote attacks access compromises the CIA of the organizations system.

MITIGATION STRATEGIES

1. CVE-2016-2183

Reconfigure the affected application in order to ensure other parts of the system are not compromised.

Disable and deprecate the current cipher suites in the TLS or SSL configuration.

Disable all 3DES Ciphers

Use of stronger encryption algorithms such as AES for stronger and trusted protection from remote user attacks.

CWE 327

Remove all TLS 1.0 protocol dependencies within the software.

Update system protocols use to TLS 1.2 and TLS 1.3

2. CVE-2019-20372

Upgrade to nginx version 1.17.7 or later versions

Plugin #51192

Renew SSL certificate to check whether it will update to a trusted version.

Purchase new SSL certificate

Cross domain and client access policies

Review permissions set to various web clients

Provide permissions using the Principle of Least privilege to web clients in order to maintain confidentiality and eliminate risk of Cross- Site forgery attacks.

lmplement Token Synchronization, that is effective in mitigating CSRF attacks because it ensures that requests can only be made from a valid user session. This means that even if an attacker can generate a request that looks like it comes from the user, they will not have the correct token and the request will be rejected.

3. CVE-2005-3299

lmplement Whitelisting. This is a list of trusted email addresses, lP addresses, domain names or applications or even executable files, while denying all others. By having this and only allowing trusted sources, it eliminates the risk of third parties accessing files they are not authorized to access.

Use of databases rather than servers. lnstead of saving files or information that can be compromised and have sensitive information on a web server, saving them on a database is more secure. This allows for ClA to be maintained.

Restrict execution permissions for upload directories as well as upload file sizes.

Run dynamic application security tests to determine if your code is vulnerable to file inclusion exploits.

Sanitize user-supplied inputs, including GET/POST and URL parameters, cookie values, and HTTP header values. Apply validation on the server side, not on the client side.

By implementing these mitigation and remediation methods, it is possible to maintain security within the organizations system. Furthermore, by identifying the various risks the organization is vulnerable to, it has allowed us to stay ahead by making the necessary patches to our system and areas that need to be reconfigured entirely.

CONCLUSlON

The project conducted by the security team at CDF was an overall success as it allowed us to identify, evaluate and protect our systems from vulnerabilities we are susceptible to as an organization. Through the use of Nessus and Nmap, six main vulnerabilities were found to exist with one being categorized as “High” while the rest maintained an overall scoring of “Medium”.

SUMMARY OF FlNDlNGS:

Vulnerabilities identified with a CVSS High score: SSL Medium Strength Cipher Suites Supported (SWEET32).

Vulnerabilities identified with a CVSS Medium score: TLS 1.0 version, nginx information disclosure, SSL certificate not trusted, PHP file inclusion and Cross Site forgery attacks.

KEY RECOMMENDATlONS:

The information below is a summary of the found mitigation and preventative methods that will be implemented to addressing each of the six vulnerabilities found within the systems;

Patching and Updates of system: This includes updating of untrusted SSL certificates and update to newer versions of protocols such as the TLS 1.2 in order to ensure that system has up to date security measures and does not pave way for man in the middle potential attacks that are found within older versions of software.

lmplementation of safer security measures: This includes implementing safer habits that are more secure to the system such as use of Whitelisting that ensures only trusted and authorized sources have access rights. Furthermore, this includes reviewing of current permissions set to web servers and reconfiguring them to increase security within our systems. Finally, use of databases rather than web servers for data storage to avoid information being compromised.

Network protection and safeguarding: This incudes proactive safeguarding measures to ensure CA of the organization systems including implementation of stronger firewalls as well as permissions to web clients being provided using the Principle of Least Privilege agenda.

Regular monitoring: This includes consistent vulnerability assessments to ensure that not only is system equipped to handle newer cyberattack methods but to safeguard from current vulnerabilities and current exploits. Furthermore, this also includes regular monitoring through logs in order for faster detection of potential attacks or unusual activity for a more proactive approach.

The security team has made a commitment to ensuring the safety of the organizations systems, one of the ways being Vulnerability Management. Through the recommendations highlighted above, there is a commitment to security of the company and protection of our Confidentiality, lntegrity and Availability from unauthorized third parties.

TASK 2 PENETRATlON TESTlNG ON WEB APPLlCATlONS

lNTRODUCTlON TO PROJECT:

The project titled “Penetration Testing on Web Applications” was created Vulnerabilities and Penetration Testing (VAPT) are both security services that focus on identifying vulnerabilities in the network, server and system infrastructure. Both the services serves a different purpose and are carried out to achieve different but complimentary goals. Vulnerability Assessment focuses on internal organizational security, while Penetration Testing focuses on external real-world risk. Vulnerability Assessment (VA) is a rapid automated review of network devices, servers and systems to identify key vulnerabilities and configuration issues that an attacker may be able to take advantage off. lts generally conducted within the network on internal devices and due to its low footprint can be carried out as often as every day. Penetration Testing (PT or PenTest) is an in-depth expert-driven activity focused on identifying various possible routes an attacker could use to break into the network. ln-addition with the vulnerabilities it also identifies the potential damage and further internal compromise an attacker could carry out once they are past the perimeter. Web Application Penetration Testing is security testing methods for security holes or vulnerabilities in web applications and corporate websites. Due to these vulnerabilities, websites are left open for exploitation.

ln this Project, Penetration Testing is conducted on Open source wed application and the web application name is Altoro Mutual (testfire.net). lt is a banking web application and it is provided by HCL company A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

ASSESSMENT OVERVlEW:

Our Task to evaluate the security posture of its Company infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the OWASP Testing Guide, and customized testing frameworks.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.

- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.

- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.

- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

DIFFERENT TYPES OF WEB APPLICATION PENETRATION TESTING TOOLS:

The project was conducted through the use of one Penetration Tester on Kali Linux. The rationale behind using various sources and methods was to ensure a comprehensive and wide range of vulnerabilities to be detected. It was also to ensure a lack of overreliance on one source, but to implement various sources to increase accuracy in results and findings.

TOOLS:

- NMAP:

NMAP is short for Network Mapper. It is an open-source tool that helps you map a network by scanning ports, discovering operating systems, and creating an inventory of devices and the services running on them. It sends differently structured packets for different transport layer protocols which return with IP addresses and other information. You can use this information for Metasploit currently includes nearly 1677 exploits along with almost 500 payloads that include

- Command shell payloads
- Dynamic payloads
- Meterpreter payloads
- Static payloads

The framework also includes listeners, encoders, post-exploitation code, and whatnot.

- Host discovery
- OS fingerprinting
- Service discovery

- Security auditing

You can use the tool for a large network with thousands of devices and ports.

- Wireshark:

Wireshark is another famous open-source tool that you can use for protocol analysis. It allows you to monitor network activities at a microscopic level. It is a growing platform with thousands of developers contributing from across the world.

With Wireshark you can perform

- Live capture and offline analysis
- Inspection of hundreds of different protocols
- Browse captured data via GUI
- Decrypt protocols
- Read live data from Ethernet, and a number of other mediums
- Export output to XML, PostScript, CSV, or plain text

Wireshark is the industry standard for protocol analysis in many different sectors. If you know what you are doing, it is a great tool to use.

- Metasploit:

Metasploit is a Ruby-based open-source framework, used by both ethical hackers and malicious actors to probe systematic vulnerabilities on networks and servers. The Metasploit framework also contains portions of fuzzing, anti-forensic, and evasion tools. It is easy to install and can work on a wide range of platforms regardless of the languages they run on. The popularity and the wide availability of Metasploit among professional hackers make it an important tool for Penetration Testers as well.

- Burp Suite:

Burp Suite is a set of penetration testing tools by Portswigger Web Security. It is used by ethical hackers, pen-testers, and security engineers. It is like a one-stop-shop for bug bounty hunters and security researchers. Let us take a look at a few tools included in Burp Suite.

- Spider: It is a web crawler. You can use it to map the target application. It lets you create an inventory of all the endpoints, monitor their functionalities, and look for vulnerabilities.

- Proxy: As explained earlier, a proxy sits between the browser and the internet to monitor, and modify the requests and responses in transit.

- Intruder: It runs a set of values through an input point and lets you analyze the output for success, failure and content length. These aside the suite includes Repeater, Sequencer, Decoder, Extender, and some other add-on tools.

FINDING SEVERITY RATINGS:

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Critical

9.0-10.0 Exploitation is straightforward and usually results

in system-level compromise. It is advised to form a plan of action and patch immediately.

High

7.0-8.9 Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.

Moderate

4.0-6.9 Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after

high-priority issues have been resolved.

Low

0.1-3.9 Vulnerabilities are non-exploitable but would reduce an organization’s attack surface. It is advised to form a plan of action and patch during the next maintenance window.

lnformational

N/A No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

RlSK FACTORS

Risk is measured by two factors: Likelihood and lmpact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

lmpact

lmpact measures the potential vulnerability’s effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope Exclusions

Our Project did not perform any of the following attacks during testing:

- Denial of Service (DoS)

- Phishing/Social Engineering

TESTlNG SUMMARY

The Web Application assessment evaluated TestFire Web Application security posture. From an internal perspective, the pen tester performed manual vulnerability assessment against the website link provided by testfire to evaluate the overall patching health of the website. The pen tester found many vulnerabilities on the web app and exploited it.

The pen tester used some manual techniques to exploit basic level of vulnerabilities and was successfully able to get admin access to the web app using various exploitation methods. The security of the web app is completely compromised by basic level of exploitation which are mentioned below along with the evidence. Since it was a time bound assessment so most the vulnerabilities were reported by the pen tester in the given time. However, there are more serious exploits still present in the system but due to the time limit only some of them are reported.

KEY STRENGTHS AND WEAKNESSES:

The following identifies the key strengths identified during the assessment: The following identifies the key weaknesses identified during the assessment:

- Default credentials were used in web app
- Broken Authentication.
- lmproper input validation on various input parameters
- lnternal server error being displayed.
- lnsecure direct object reference allowed.
- lnjection allowed on various input parameters.
- No mechanism to stop brute force of login information
- Displaying user information on web page.
- GitHub source code link of web app displayed on screen.

VULNERABlLlTY SUMMARY & REPORT CARD

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Web Application Penetration Test Findings

Critical

High

Moderate

Low

Informational

FindingSeverity Recommendation

Web Application Penetration Test

1: Apache Tomcat insecure default administrative password (High) Change the default

password

2: Insecure Direct Object reference (High) Validate and filter any sort of input even if it's not malicious.

3: SQL Injection Vulnerability allowing login bypass. (Critical) Use parametrized query instead of string concatenation within the query.

4: Login Brute on username and password. (High) Restrict multiple requests from a same source.

5: Improper input Validation. (High) Proper validation and filter of input to

be performed.

6: Reflected XSS. (Moderate) Implement proper filtration of various characters during input.

7. Displaying user info on web app (Low) Properly check for any user info displayed on screen.

8. Displaying internal server error. (Informational) Properly check error log and error

messages displayed on screen.

TECHNICAL FINDINGS

Web Application Penetration Test Findings

Finding 001: Apache Tomcat Insecure Default Administrative Password (High)

Description: Test fire allows the use of default admin password being used on the login page due to which any unauthenticated user knowing the default password available on the internet can gain access to the admin account and have admin privileges.

Risk: Likelihood: High – This attack is effective on web app and have major consequences to it.

Impact: Very High – This attack gives admin privilege to a user who can make any changes on the web application.

System: All

Tools Used: Manually

References: https://www.acunetix.com/vulnerabilities/web/apache- geronimo- default-administrative-credentials/

CONCLUSION:

Computer security is a vast topic that is becoming more important because the world is becoming highly interconnected, with networks being used to carry out critical transactions. Cyber-crime continues to diverge down different paths with each New Year that passes and so does the security of the information. The latest and disruptive technologies, along with the new cyber tools and threats that come to light each day, are challenging organizations with not only how they secure their infrastructure, but how they require new platforms and intelligence to do so. There is no perfect solution for cybercrimes but we should try our level best to minimize them in order to have a safe and secure future in cyber space.