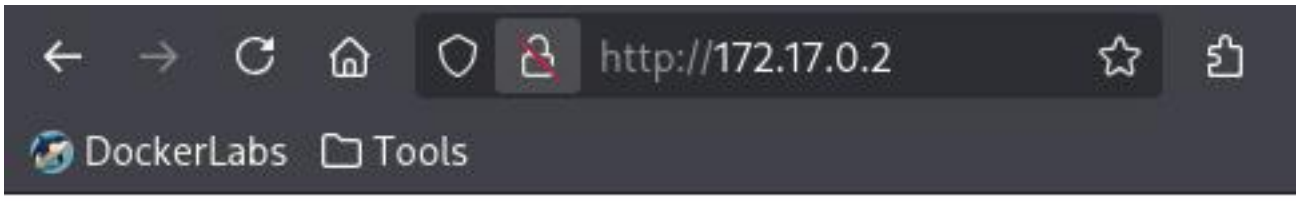
Como siempre usamos ***nmap*** para descubrir posibles vías de entrada:

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- --open -sSVC -n -nP 172.17.0.2
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-20 02:53 -0500
Nmap scan report for 172.17.0.2
Host is up (0.0000060s latency).
Not shown: 65532 closed tcp ports (reset)
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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.20 seconds
```

Tenemos abiertos los puertos 80 (HTTP), 3000 también para HTTP, aunque parece que no va a servir de mucho, y el 5000 para SSH.

Hacemos lo de costumbre e inspeccionamos lo primero la página web:



Bienvenido a Mi Sitio

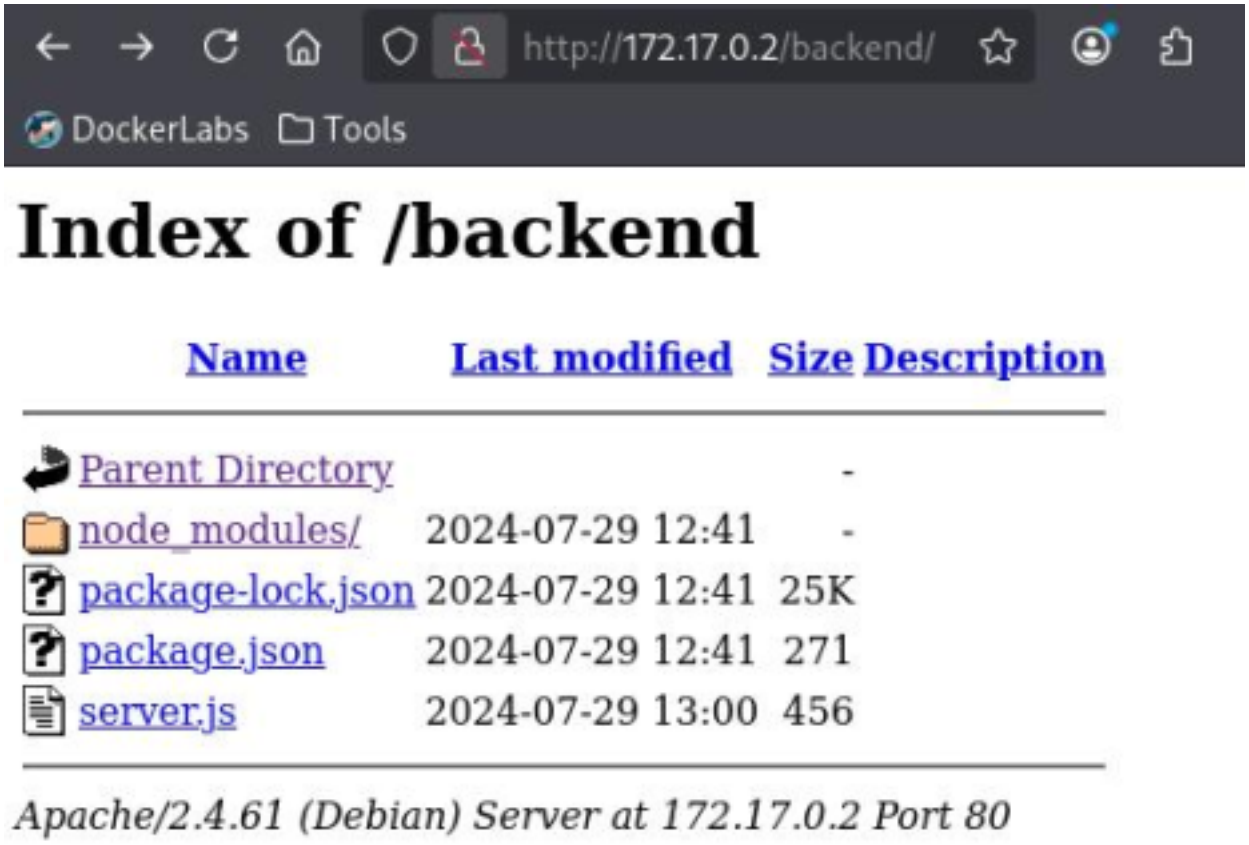
Boton en fase beta

Nada aprovechable.

Buscamos algún directorio interesante, utilizando ***gobuster***:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://172.17.0.2/ -w /usr/share/wordlists/dirb/common.txt -x php,txt,html
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              php,txt,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta.php             (Status: 403) [Size: 275]
/.htaccess.php        (Status: 403) [Size: 275]
/.hta.html            (Status: 403) [Size: 275]
/.hta                 (Status: 403) [Size: 275]
/.htpasswd.php        (Status: 403) [Size: 275]
/.htpasswd            (Status: 403) [Size: 275]
/.htaccess.html       (Status: 403) [Size: 275]
/.htaccess.txt        (Status: 403) [Size: 275]
/.hta.txt             (Status: 403) [Size: 275]
/.htpasswd.txt        (Status: 403) [Size: 275]
/.htaccess            (Status: 403) [Size: 275]
/.htpasswd.html       (Status: 403) [Size: 275]
/backend              (Status: 301) [Size: 310] [--> http://172.17.0.2/backend/]
/index.html           (Status: 200) [Size: 234]
/index.html           (Status: 200) [Size: 234]
/javascript           (Status: 301) [Size: 313] [--> http://172.17.0.2/javascript/]
/server-status        (Status: 403) [Size: 275]
Progress: 18452 / 18452 (100.00%)
===============================================================
Finished
===============================================================
```

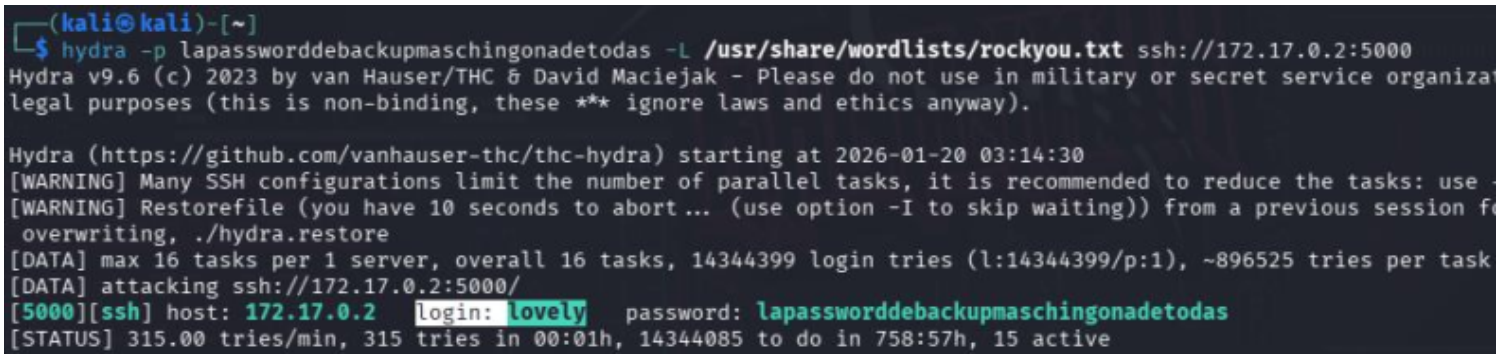
Miramos en "/backend":



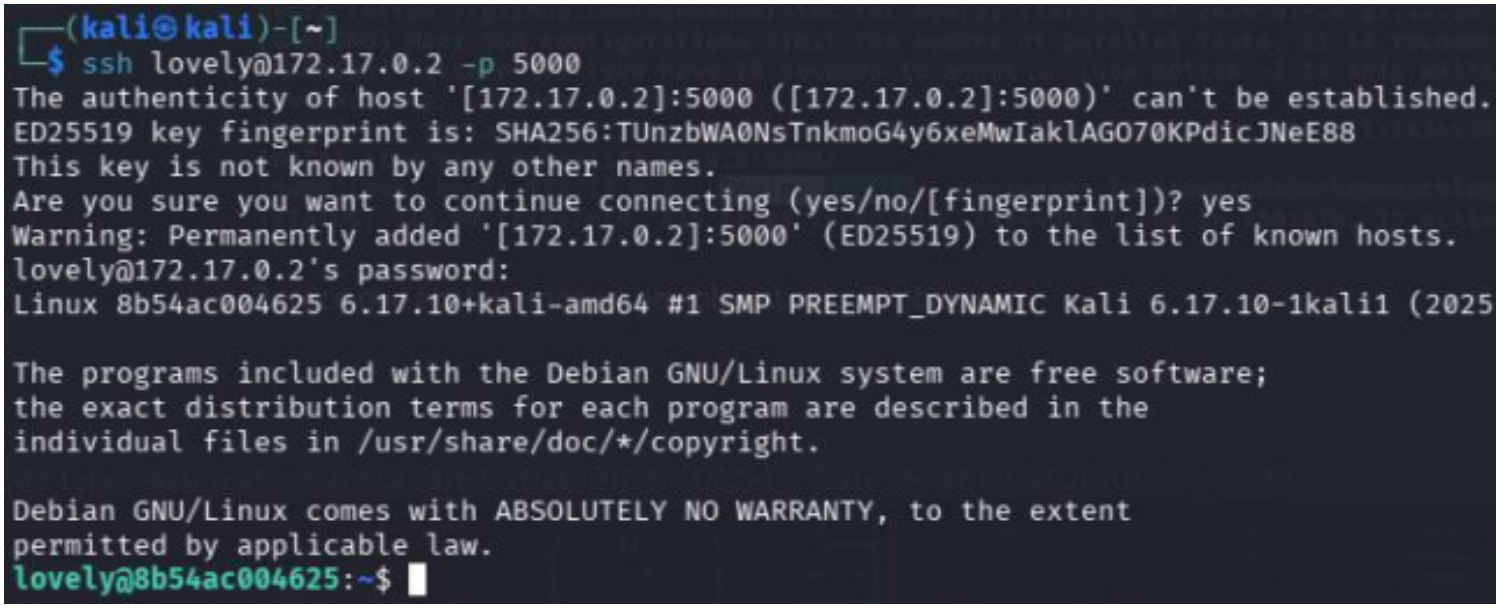
Y le echamos un vistazo a ese archivo de javascript (server.js):

```
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}`);
});
```

Esto promete, jajaja. Vamos a ver si conseguimos conectarnos por SSH con ayuda de ***Hydra***:



Parece que ahí tenemos un posible usuario. Probamos:

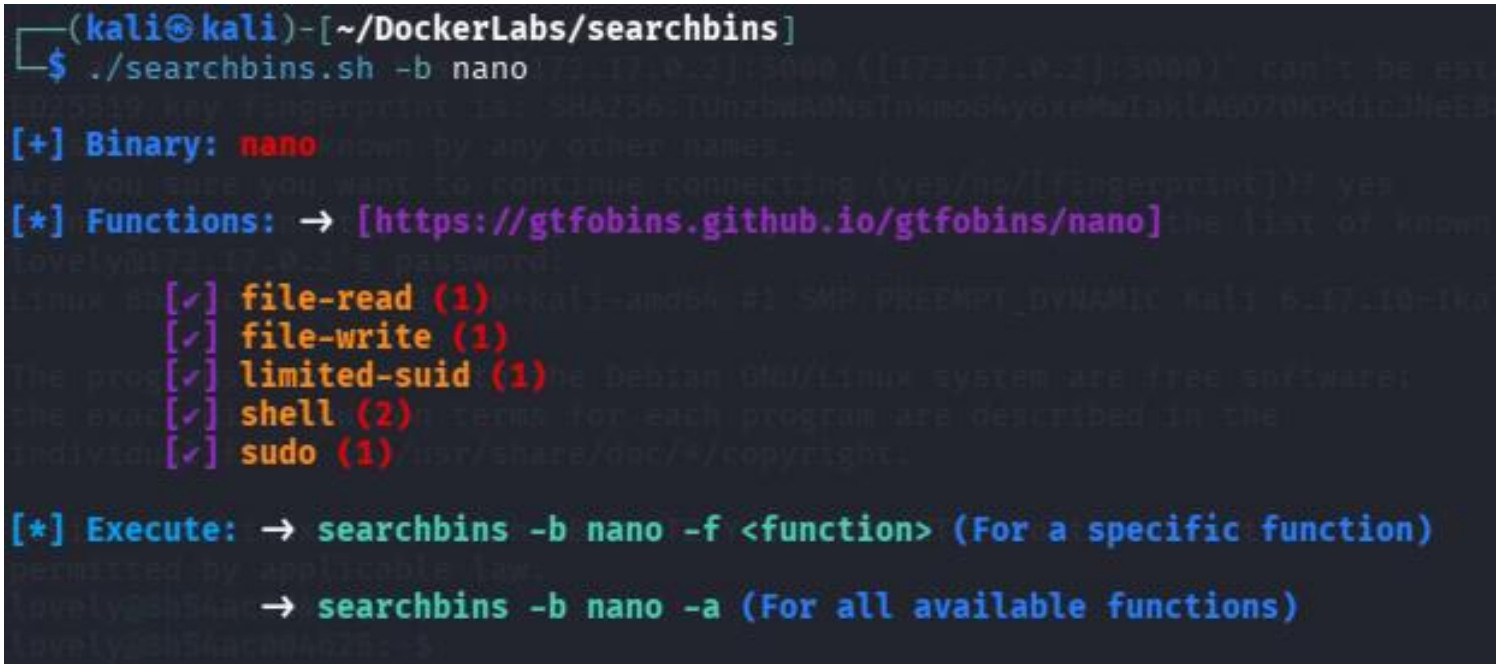


Estamos dentro. Vamos a ver si el usuario 'lovely' puede ejecutar algún binario con permisos root:

Pues sí, parece que 'nano'.

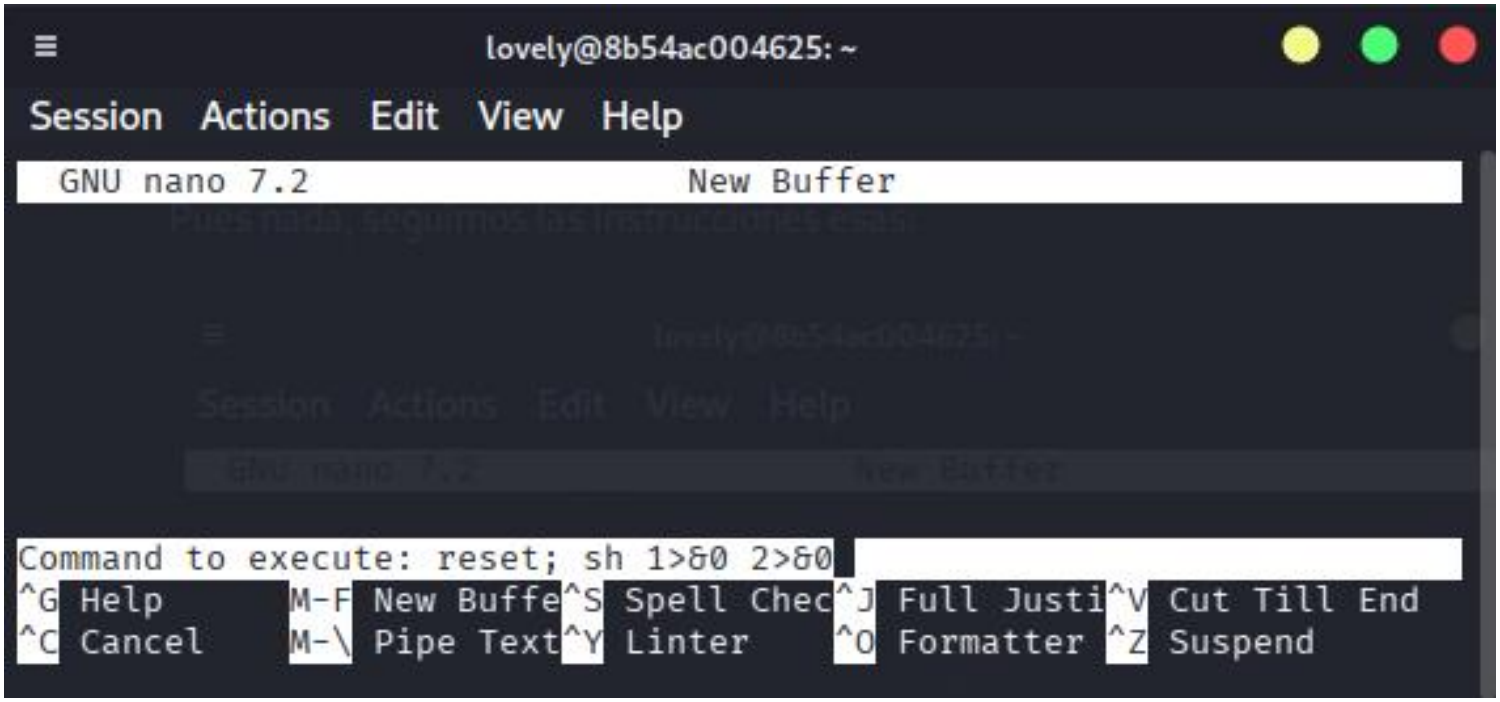
Hoy para variar, no vamos a utilizar la página de GTFObins (bueno, indirectamente sí).
Se trata de un script muy cómodo llamado 'searchbins.sh'. Tenéis en el canal de Mario un video explicándolo muy bien.
Lo podéis bajar de GitHub (https://github.com/r1vs3c/searchbins).

El funcionamiento es muy sencillo. Sólo tenéis que meterle uno de los binarios que os haya aparecido al hacer 'sudo -l', mirando permisos SUID, etc.,.
En este caso, 'nano':



Y ahí nos dice que hay que indicar el método por el que obtuvimos el binario, en nuestro caso haciendo 'sudo -l':

Pues nada, seguimos las instrucciones esas:




```
root@8b54ac004625:/home/lovely# whoami
root
root@8b54ac004625:/home/lovely#
```

Y listo, somos root!!