{EPITECH}

# BOOTSTRAP

BURP

# BOOTSTRAP

## Part 0 - Using OpenVPN


For most challenges that you will need to connect to a Virtual Machine (VM) on TryHackMe.

From this point you have two options:
- Using the AttackBox
- Using OpenVPN on your own machine.

The AttackBox is nice but comes with several issues. It is very time limited if you're not a premium subscriber, it takes a long time to load and it forces you to keep your web browser open at all time.

The OpenVPN way however, just makes you connected directly with your machine to TryHackMe network. This way, you can just use your machine with your favorite tools and payload without relying on something else.

You can choose whichever you prefer, this bootstrap will assume that you choose the OpenVPN way.

You can learn how to connect to the OpenVPN directly on the learning path on TryHackMe here

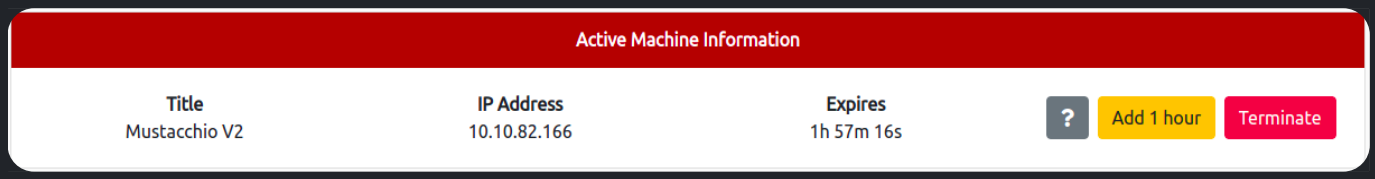
# Part 1 - Pentester 101

Because it probably is your first time pentesting, this bootstrap will explain the various toolbelt of a proper pentester.

We will be using this room : Mustacchio

## Starting the machine and /etc/hosts


Start your OpenVPN, then start the machine by clicking the green **Start Machine** button.




Once done, you will have a MACHINE_IP, you can now try to connect to this machine.
But before that, we will edit your /etc/hosts to simplify the whole process.

```
Terminal
~/B-SEC-200> sudo emacs -nw /etc/hosts
127.0.0.1 localhost
10.10.82.166 mustacchio.thm
```

It will allow you to use mustacchio.thm instead of the IP that will change every time you restart your THM machine. Sometimes, the server itself will not work properly until you set a proper hostname like that.

## Enumerating ports

First thing you should do is enumerating the ports. For that the perfect tool exists and has a name: **Nmap**


There is a lot to say about nmap, but I will let you find more interesting tutorial about it for free online : here

The basic way to use it is like this :

```
Terminal
~/B-SEC-200> nmap -vv mustacchio.thm -p-
Scanning mustacchio.thm (10.10.82.166) [65535 ports]
Discovered open port ????/tcp on 10.10.82.166
Discovered open port ????/tcp on 10.10.82.166
Discovered open port ????/tcp on 10.10.82.166
```

Scanning the machine this way will allow you to find the open ports, which can all be vectors for attacks.

What open ports do you find ?

## Enumerating a website

Once you find the open ports, you find one that is a website


First thing to do is manual enumeration, also called the “happy path”. You should use the web-site as a normal user first to find out potential issues.

This one however, does not have any special visible issues, so let’s get back to enumerating more, with one of my favorite tool: **Gobuster**

To use Gobuster you will need a common list of word like this one : here

```
Terminal
~/B-SEC-200> gobuster dir -u mustacchio.thm -w common.txt
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url: http://mustacchio.thm
[+] Method: GET
[+] Wordlist: common.txt
2021/12/01 15:56:02 Starting gobuster in directory enumeration mode
/?????? (Status: 301) [Size: 317] [-> http://mustacchio.thm/??????/]
/?????? (Status: 301) [Size: 317] [-> http://mustacchio.thm/??????/]
/?????? (Status: 301) [Size: 317] [-> http://mustacchio.thm/??????/]
2021/12/01 15:56:19 Finished
```

No fancy icon for this tool, but it will allow you to test many routes on a website, including routes that does not have a direct link to it.

Using this tool, can you find the hidden directory ?
Did you find anything interesting ? (Like a user for instance …)

## Password cracking

If all goes well for you, you have found a hash for the **admin** user.
A hash is a one-way function, so you can't reverse a hash. What you can do is re-hash a list of potential password to see if it produce the same hash !

For that you need two things:
- A good dictonary : rockyou.txt is a good start
- A password cracker tool: either John the Ripper or Hashcat, or both


Both are equally as good, john is sometimes more useful in cracking zip key file, ssh password file, etc… while hashcat can crack a bit faster and more different types of hash.

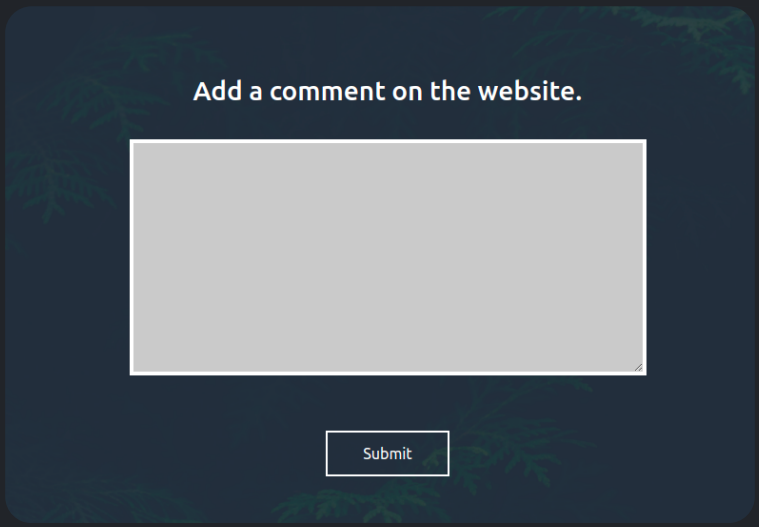
For this bootstrap I'll use hashcat.

```
Terminal
~/B-SEC-200> hashcat -a 0 -m ??? [hashes] rockyou.txt
1868e36a6d2b17d4c2745f1659433a54d4bc5f4b:??????????
Session..........: hashcat
Status...........: Cracked
Hash.Target......: 1868e36a6d2b17d4c2745f1659433a54d4bc5f4b
```

Once you find the password, you must find out where to use it.

## Common weaknesses and payload

Once logged in on the website, you'll find another point of entry. Each room will have different ways of hacking into the machine, can you identify this one ?




Have you ever heard of XML injection ?

Have you check the source code of the website ?

Once you have exploited this weakness in the website, what can you do with it ? Which file can you read that will be of use to you ?

Don’t forget that John is your best friend, if you ever need to by pass a passphrase !

## Privilege Escalation

If you made it up to there, congratulations, you are almost done ! You should have found the user.txt file that will give you half the points.
This proves that you’ve been able to access the machine instead of just using the web-service.

Now one thing common in boot2root and pentest engagements is going from user to the super admin, root. Every room has different ways to do privesc, some obvious, some less obvious and more realistic.

To enumerate yet again the potential weaknesses, we can automate this with a bunch of tools, my favorite one being **LinPeas**.

Since you’re not on your machine anymore, you have to use scp to copy this tool over ssh or host a simple python server on your machine to fetch it from the compromised host.

Once launched, the script will provide various information about potential weaknesses in the system, potentially allowing a privilege escalation.

```
Terminal
~/B-SEC-200> -rwsr-xr-x 1 root root 17K Jun 12 15:48 ???????? (Unknown SUID binary)
```

For example in this one you will find an unknown SUID binary, which means a custom way to privesc. For more common payload, you can use gtfobins to help you find common privesc.

Downloading the binary can be a good idea for this part.
There is a non exhaustive list of tools that can be useful:
- ghidra
- strace
- gdb

## Conclusion

With the privesc done, you should now have access to the root flag to validate the whole room, congratulations !

The only way to get better at hacking is by practicing, so work on the **Burp** challenges, read a lot about techniques and tools and train on the easier room on TryHackMe.

v 2.0