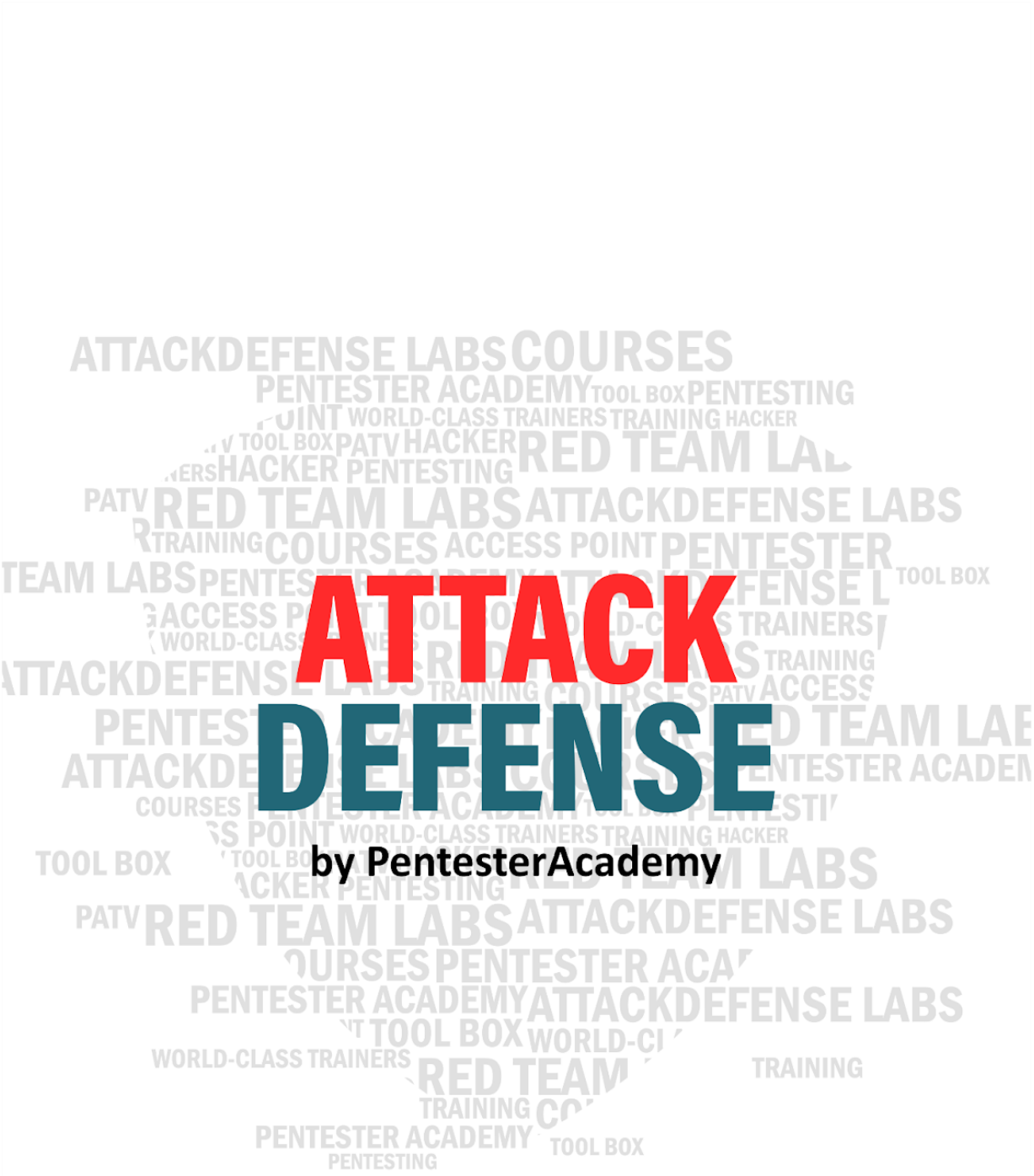
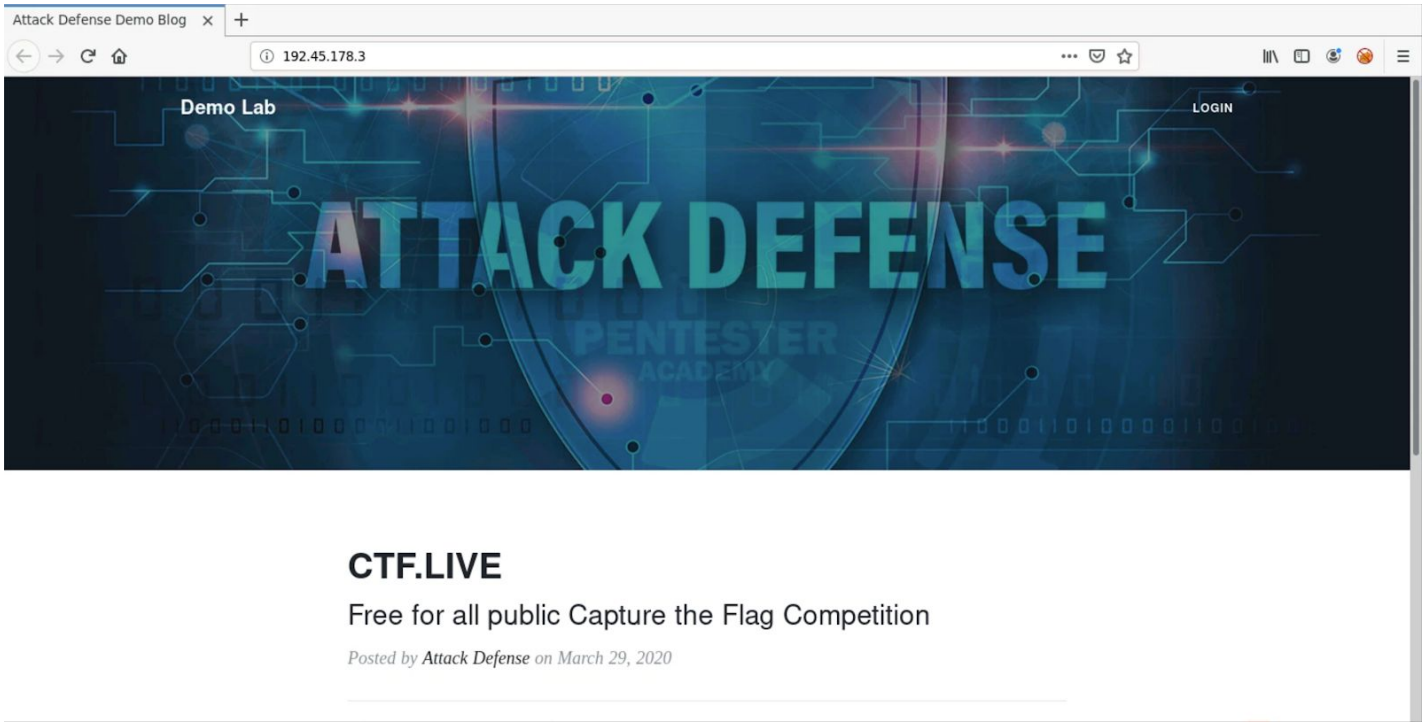

ATTACK
DEFENSE
by PentesterAcademy

Name	HTTP Method Enumeration
URL	https://attackdefense.com/challengedetails?cid=1802
Type	Webapp Pentesting Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

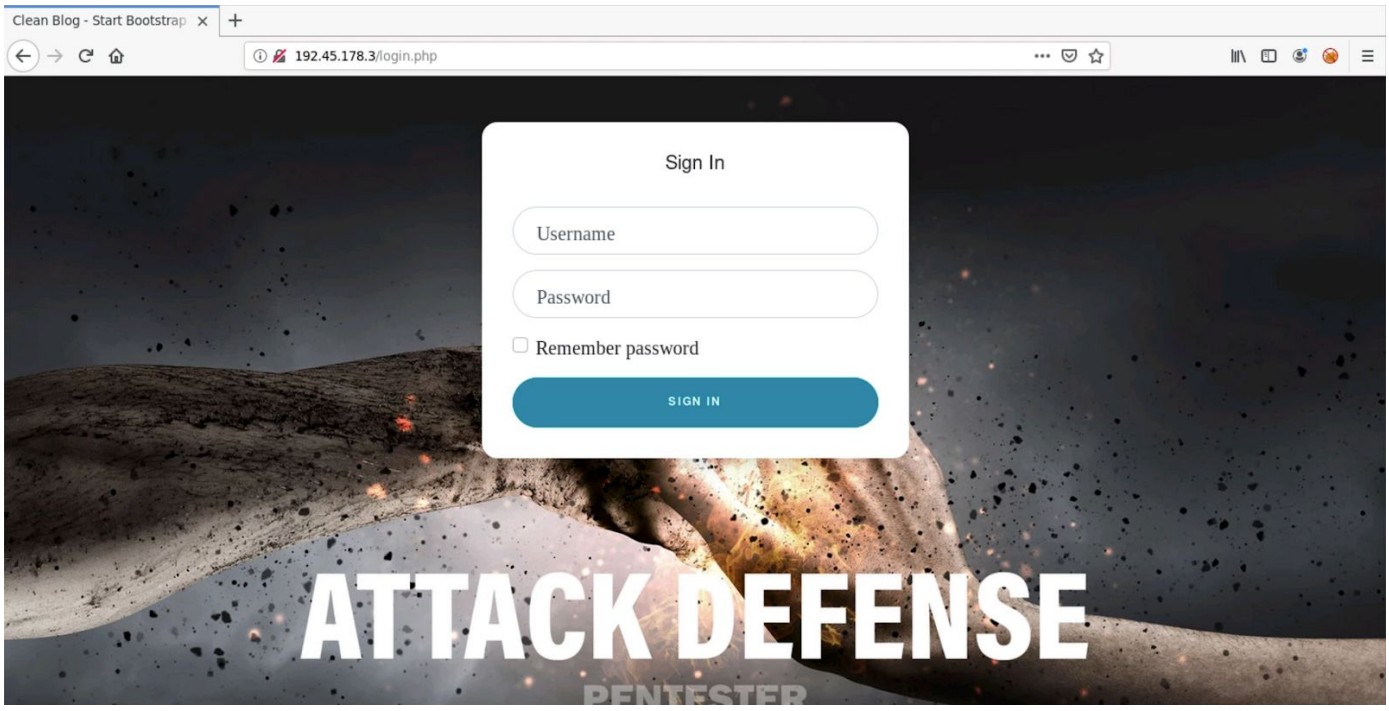
In this exercise, we will take a look at how to use burp suite and curl to enumerate the HTTP Methods supported by various web pages. Inspecting the web application.

Inspecting the web application.



There are two new links which can be followed from the home page. The login link on the navigation bar and the blog post. The login link redirects to "login.php" and the Blog link redirects to "post.php"

Step 1: Following Links: Click on the login Link.



The login page contains a form.

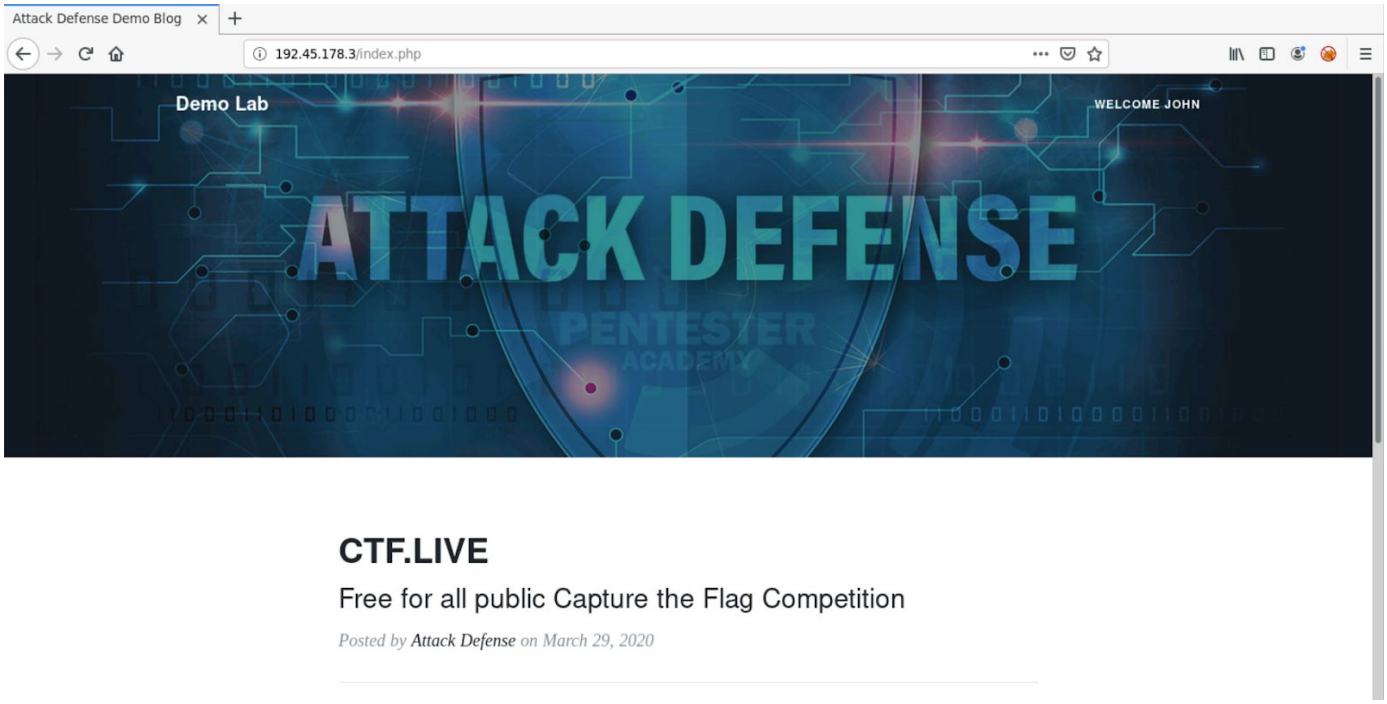
Step 2: Identify the endpoint which processes the form fields. Right click on the web page and click on the view source.

```
<h5 class="card-title text-center">Sign In</h5>
<form class="form-signin" action="/login.php" method="POST">
  <div class="form-label-group">
    <input type="text" id="inputEmail" name="name" class="form-control" placeholder="Username" required autofocus>
    <label for="inputEmail">Username</label>
  </div>

  <div class="form-label-group">
    <input type="password" id="inputPassword" name="password" class="form-control" placeholder="Password" required>
    <label for="inputPassword">Password</label>
  </div>

  <div class="custom-control custom-checkbox mb-3">
    <input type="checkbox" class="custom-control-input" id="customCheck1">
    <label class="custom-control-label" for="customCheck1">Remember password</label>
  </div>
  <button class="btn btn-lg btn-primary btn-block text-uppercase" type="submit">Sign in</button>

</form>
</div>
```

The parameters are passed in POST request to the same "login.php" page.

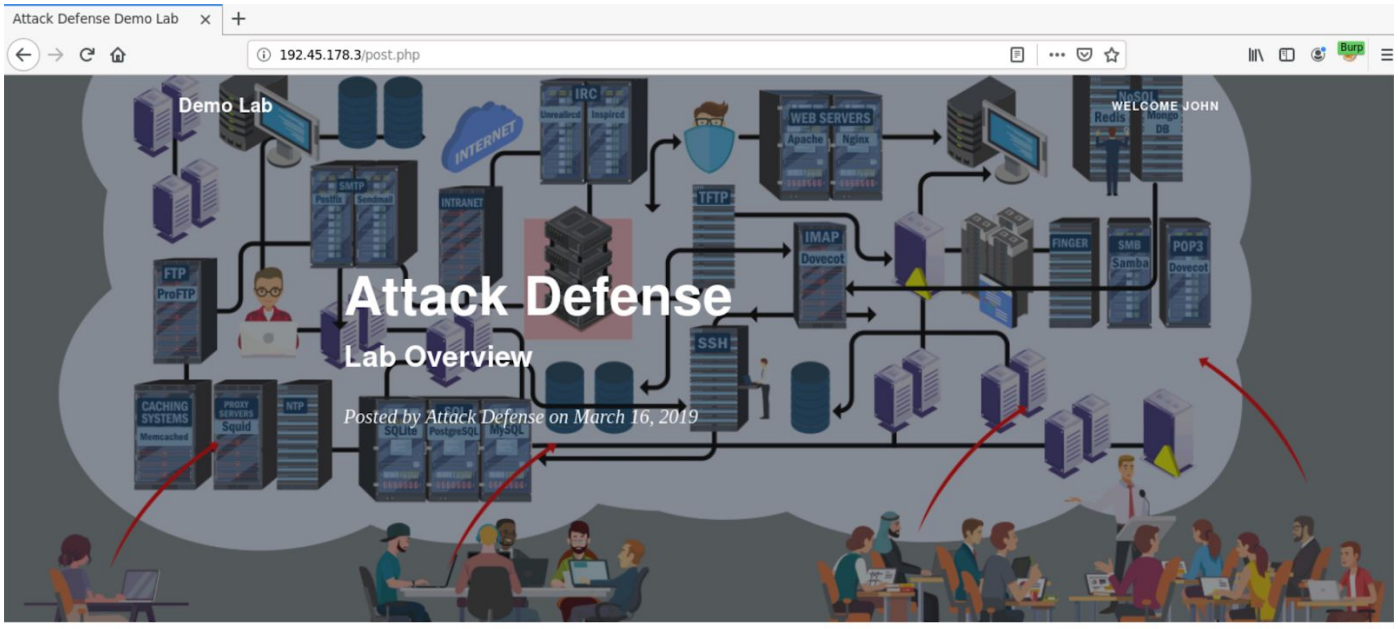
Step 3: Login to the web application with the provided credentials.

Username: john
Password: password



After login instead of the login link "Welcome John" message is displayed.

Step 4: Follow the remaining link. Click on the blog post.



1700+ Lab Exercises, Wide Coverage, New Labs Weekly!

Our labs host over 1700+ unique lab exercises on topics spanning from recon, exploitation, post-exploitation, data exfiltration, web applications, traffic analysis, CVEs, network components, infrastructure attacks, privilege escalation, forensics, firmware analysis, reversing, secure coding, IoT networks, Metasploit, Python for infosec and many others. New labs are added weekly!

Targets: Modern Components

Technology changes rapidly and so should your pentesting lab! We use modern network and stack components extensively in our labs, including but not limited to: Caching servers, SQL and NoSQL databases, distributed queues and databases, proxy servers, IMAP, POP3, SMTP servers, SMB servers and others. Our collection of labs use products like Nginx, Tornado, NodeJS, Gunicorn, MySQL, SQLite, PostgreSQL, Redis, MongoDB, ArangoDB, Couchbase, Apache Ignite, Kafka, Squid, Dovecot, Graylog, Samba and others

The Web pages which can be accessed by following the links are: index.php, login.php and post.php.

Using dirb to identify hidden directories.

Command: dirb http://192.45.178.3

```
root@attackdefense:~# dirb http://192.45.178.3

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue May  5 23:57:19 2020
URL_BASE: http://192.45.178.3/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.45.178.3/ ----
+ http://192.45.178.3/.git/HEAD (CODE:200|SIZE:23)
+ http://192.45.178.3/cgi-bin/ (CODE:403|SIZE:210)
==> DIRECTORY: http://192.45.178.3/css/
==> DIRECTORY: http://192.45.178.3/img/
+ http://192.45.178.3/index.php (CODE:200|SIZE:4407)
==> DIRECTORY: http://192.45.178.3/js/
+ http://192.45.178.3/LICENSE (CODE:200|SIZE:10273)
==> DIRECTORY: http://192.45.178.3/mail/
+ http://192.45.178.3/phpinfo.php (CODE:200|SIZE:74648)
+ http://192.45.178.3/server-status (CODE:403|SIZE:215)
==> DIRECTORY: http://192.45.178.3/uploads/
==> DIRECTORY: http://192.45.178.3/vendor/
```

The directories which are present on the server are css, img, js, mail, uploads and vendor.

Interacting with the home page with CURL.

Step 1: Sending GET request:

Command: curl -X GET 192.45.178.3

```
root@attackdefense:~# curl -X GET 192.45.178.3

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

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Attack Defense Demo Blog</title>

  <!-- Bootstrap core CSS -->
  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom fonts for this template -->
  <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
  <!-- <link href='css/lora.css' rel='stylesheet' type='text/css'> -->
  <!-- <link href='css/open-sans.css' rel='stylesheet' type='text/css'> -->

  <!-- Custom styles for this template -->
  <link href="css/clean-blog.min.css" rel="stylesheet">

</head>
```

Step 2: Sending HEAD request

Command: curl -I 192.45.178.3

```
root@attackdefense:~# curl -I 192.45.178.3
HTTP/1.1 200 OK
Date: Tue, 05 May 2020 18:05:07 GMT
Server: Apache
X-Powered-By: PHP/5.5.9-1ubuntu4.25
Set-Cookie: PHPSESSID=uge9aok453c6iu8c193tjiohm2; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html

root@attackdefense:~#
```

Step 3: Sending OPTIONS request.

Command: curl -X OPTIONS 192.45.178.3

```
root@attackdefense:~# curl -X OPTIONS 192.45.178.3
root@attackdefense:~#
root@attackdefense:~# curl -X OPTIONS 192.45.178.3 -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> OPTIONS / HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 05 May 2020 18:05:29 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=8mnj2ntm271utsuvrhdhorvfq3; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Allow: GET,HEAD,OPTIONS
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

The supported methods are GET, HEAD and OPTIONS. Accessing the web page should produce an error

Step 4: Sending POST Request.

Command: curl -X POST 192.45.178.3

```
root@attackdefense:~# curl -X POST 192.45.178.3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
</head><body>
<h1>Method Not Allowed</h1>
<p>The requested method POST is not allowed for the URL /.</p>
</body></html>
root@attackdefense:~#
```

Step 5: Sending PUT Request

Command: curl -XPUT 192.45.178.3

```
root@attackdefense:~# curl -XPUT 192.45.178.3/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
</head><body>
<h1>Method Not Allowed</h1>
<p>The requested method PUT is not allowed for the URL /.</p>
</body></html>
root@attackdefense:~#
```

Interacting with the login.php page with CURL.

Step 1: Sending OPTIONS Request

Command: curl -X OPTIONS 192.45.178.3/login.php

```
root@attackdefense:~# curl -X OPTIONS 192.45.178.3/login.php -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> OPTIONS /login.php HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 05 May 2020 21:34:32 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=e90eld4vs2iabuff9kac9f2ct5; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Allow: GET,POST,HEAD,OPTIONS
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

The allowed methods include: GET,POST,HEAD,OPTIONS.

Step 2: Sending POST Request.

Command: curl -X POST 192.45.178.3/login.php

```
root@attackdefense:~# curl -X POST 192.45.178.3/login.php

<!-- This snippet uses Font Awesome 5 Free as a dependency. You can download it at fontawesome.io! -->
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Attack Defense Demo Lab</title>

  <!-- Bootstrap core CSS -->
  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom fonts for this template -->
  <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
  <!-- <link href='css/lora.css' rel='stylesheet' type='text/css'> -->
  <!-- <link href='css/open-sans.css' rel='stylesheet' type='text/css'> -->

  <!-- Custom styles for this template -->
  <link href="css/clean-blog.min.css" rel="stylesheet">
  <style>

:root {
  --input-padding-x: 1.5rem;
  --input-padding-y: .75rem;
}
```

Unlike the home page (index.php). The login page supports POST method.

Step 3: Passing the username and password to the login.php page.

Command: curl -X POST 192.45.178.3/login.php -d "name=john&password=password" -v

```
root@attackdefense:~# curl -X POST 192.45.178.3/login.php  -d "name=john&password=password" -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> POST /login.php HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
> Content-Length: 27
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 27 out of 27 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 302 Found
< Date: Tue, 05 May 2020 21:48:30 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=q7lkn4dchm73kor4fdg4755d90; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Set-Cookie: Name=John; expires=Thu, 04-Jun-2020 21:48:30 GMT; Max-Age=2592000; path=/
< Location: /index.php
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

The login page returned a different response than before. The response contains 302 redirect.

Interacting with the post.php page with CURL.

Step 1: Sending OPTIONS request.

Commands: curl -X OPTIONS 192.45.178.3/post.php

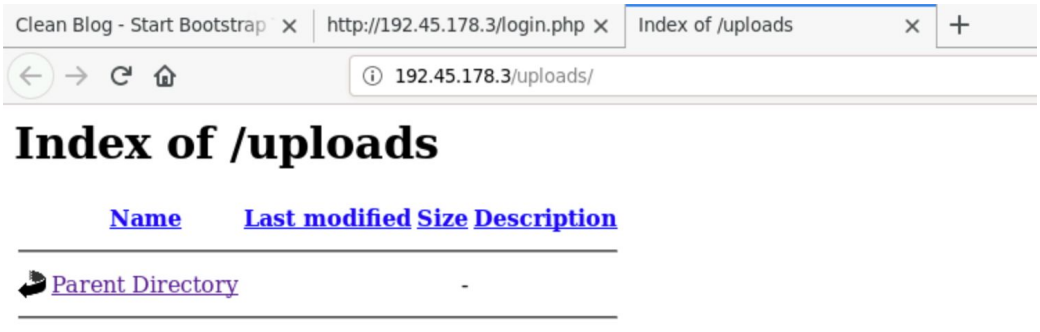
```
root@attackdefense:~# curl -X OPTIONS 192.45.178.3/post.php -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> OPTIONS /post.php HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
```

```
root@attackdefense:~# curl -X OPTIONS 192.45.178.3/post.php -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> OPTIONS /post.php HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 05 May 2020 21:52:25 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=1mqf75e49l0t52vt6i5taq47a2; path=/
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Allow: GET,POST,HEAD,OPTIONS
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

Similar to login.php, post.php has GET, POST, HEAD and OPTIONS method enabled.

Interacting with uploads directory

Step 1: Checking the content of /uploads directory.



Step 2: Sending OPTIONS request to /uploads directory.

Commands:
curl -X OPTIONS 192.45.178.3/uploads/
curl -X OPTIONS 192.45.178.3/uploads/ -v

```
root@attackdefense:~#
root@attackdefense:~# curl -X OPTIONS 192.45.178.3/uploads/
root@attackdefense:~#
root@attackdefense:~# curl -X OPTIONS 192.45.178.3/uploads/ -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> OPTIONS /uploads/ HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 05 May 2020 18:27:56 GMT
< Server: Apache
< DAV: 1,2
< DAV: <http://apache.org/dav/propset/fs/1>
< MS-Author-Via: DAV
< Allow: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK
< Content-Length: 0
< Content-Type: httpd/unix-directory
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

The Webdav module is enabled on the Apache Server, Webdav module allows file upload via PUT method.

Step 3: Uploading a file with PUT method.

Commands:
echo "Hello World" > hello.txt
curl 192.45.178.3/uploads/ --upload-file hello.txt

```
root@attackdefense:~# echo "Hello World" > hello.txt
root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# curl 192.45.178.3/uploads/ --upload-file hello.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
100    83  100    71  100    12  17750   3000 --:--:-- --:--:-- --:--:-- 20750
<title>root@attackdefense:~#
```

Step 4: Checking content of /uploads directory.

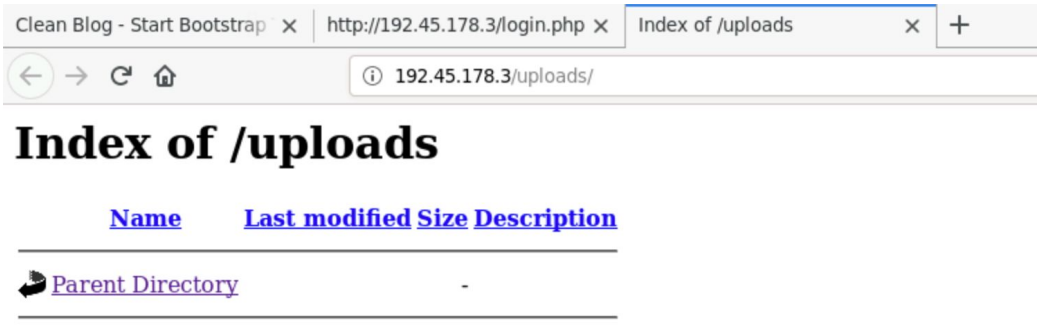
The file "hello.txt" was uploaded successfully.

Step 5: Using the DELETE method to delete the file.

Command: curl -XDELETE 192.45.178.3/uploads/hello.txt

```
root@attackdefense:~#
root@attackdefense:~# curl -X DELETE 192.45.178.3/uploads/hello.txt -v
*   Trying 192.45.178.3:80...
* TCP_NODELAY set
* Connected to 192.45.178.3 (192.45.178.3) port 80 (#0)
> DELETE /uploads/hello.txt HTTP/1.1
> Host: 192.45.178.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 204 No Content
< Date: Tue, 05 May 2020 22:16:02 GMT
< Server: Apache
< Content-Type: text/plain
<
* Connection #0 to host 192.45.178.3 left intact
root@attackdefense:~#
```

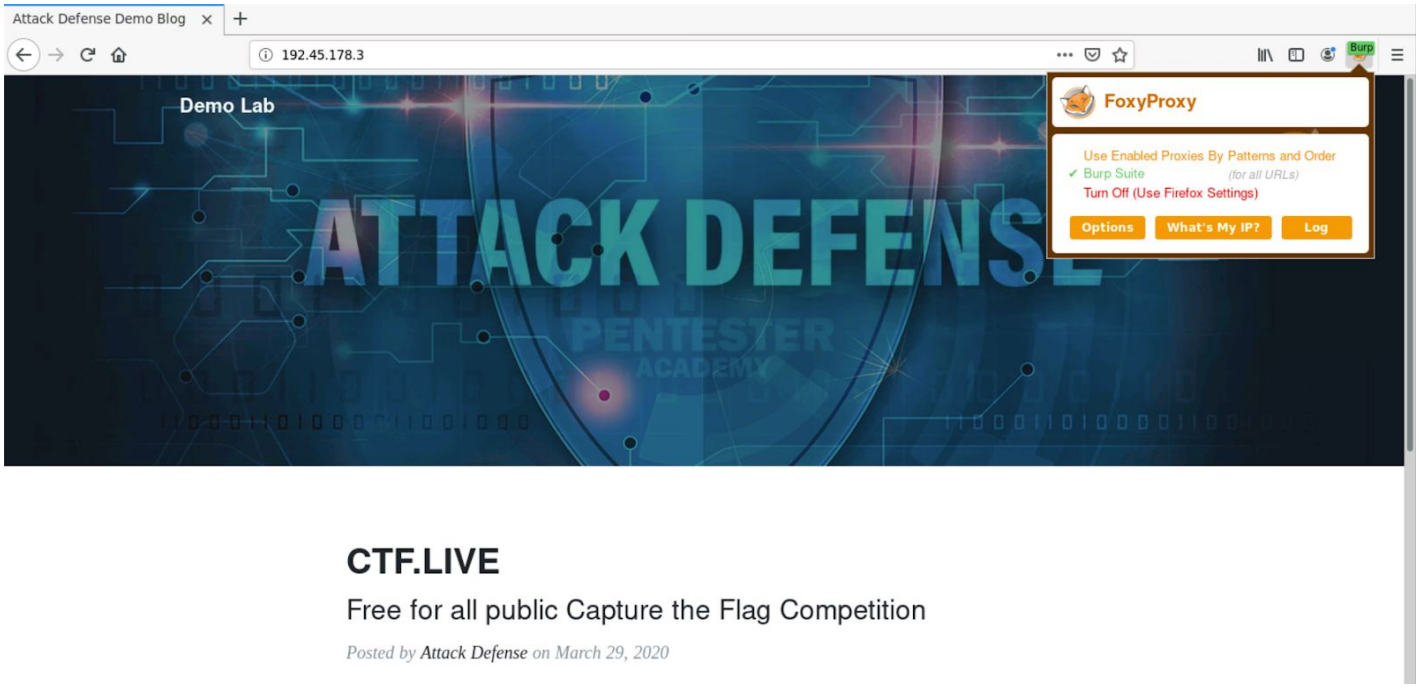
Step 6: Checking the content of /uploads directory.



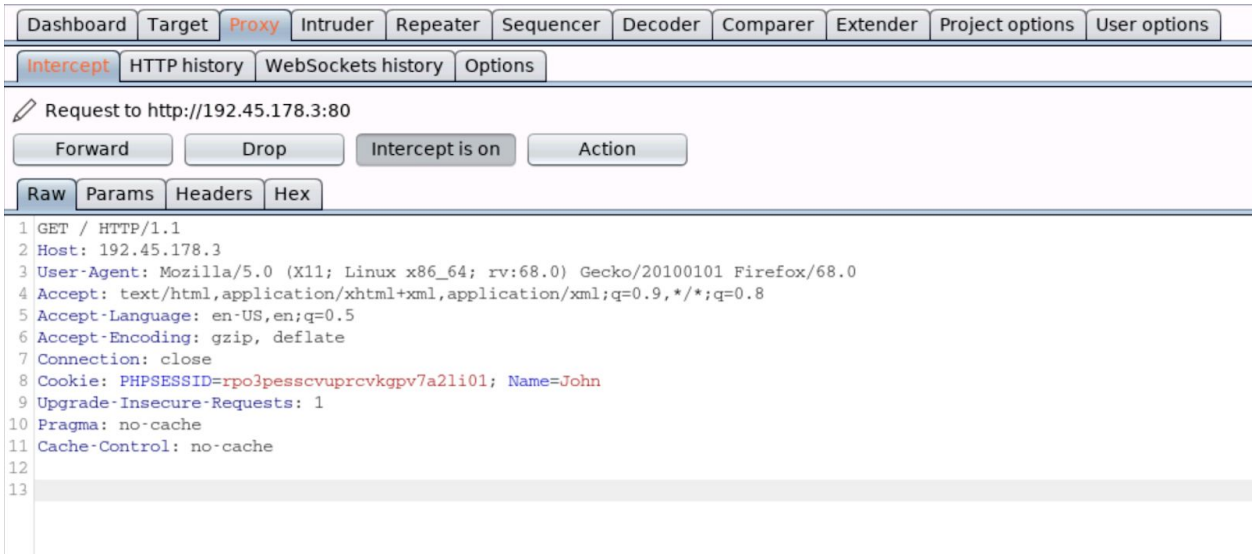
The file was deleted successfully.

Interacting with the web page with Burp Suite

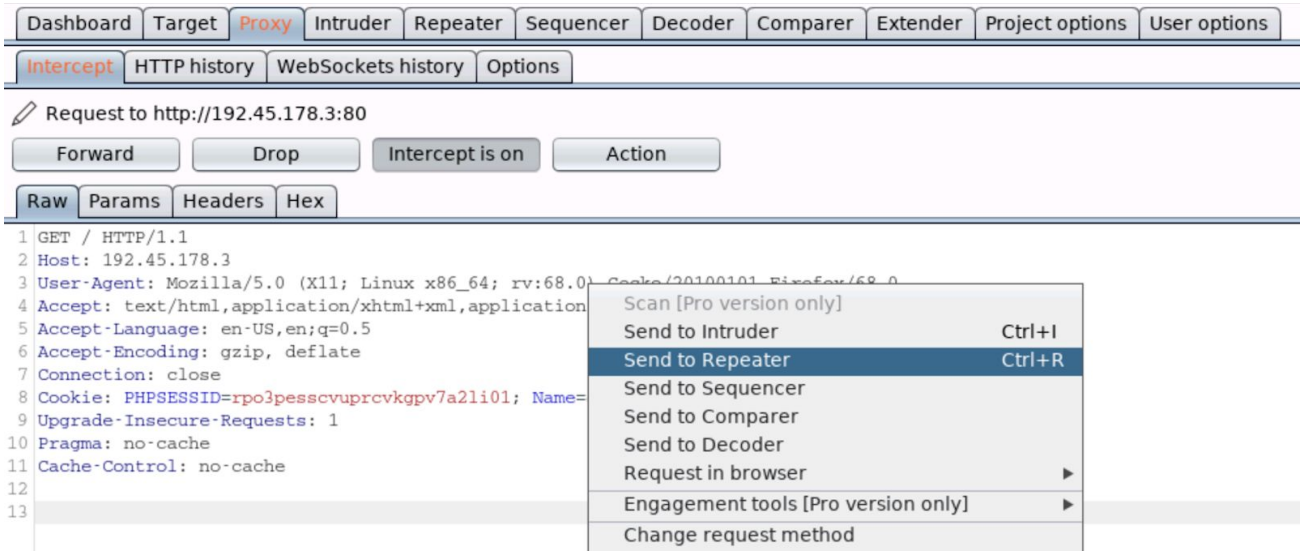
Step 1: Set the FoxyProxy to use the bup proxy. Click on the Fox icon and select "Burp Suite"



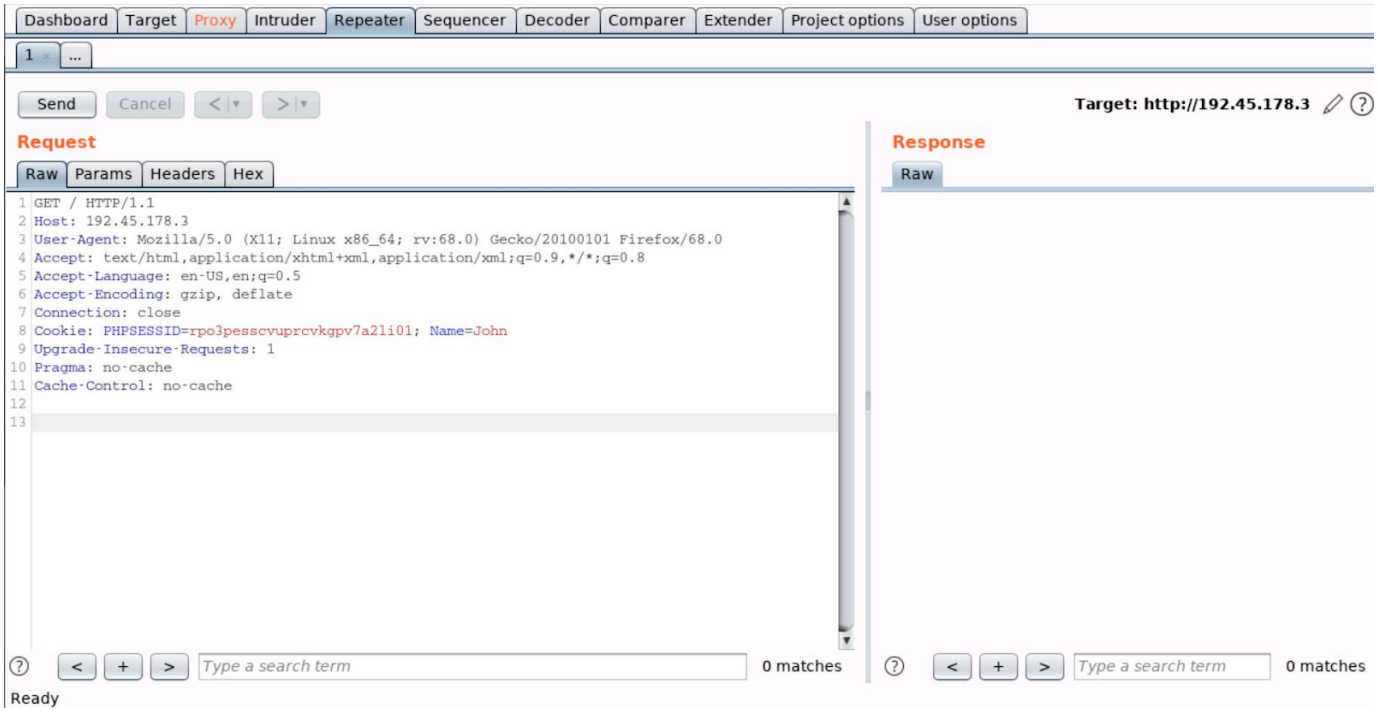
Step 2: Start burp suite. Reload the page and the request will be intercepted



Step 3: Sending request to Repeater



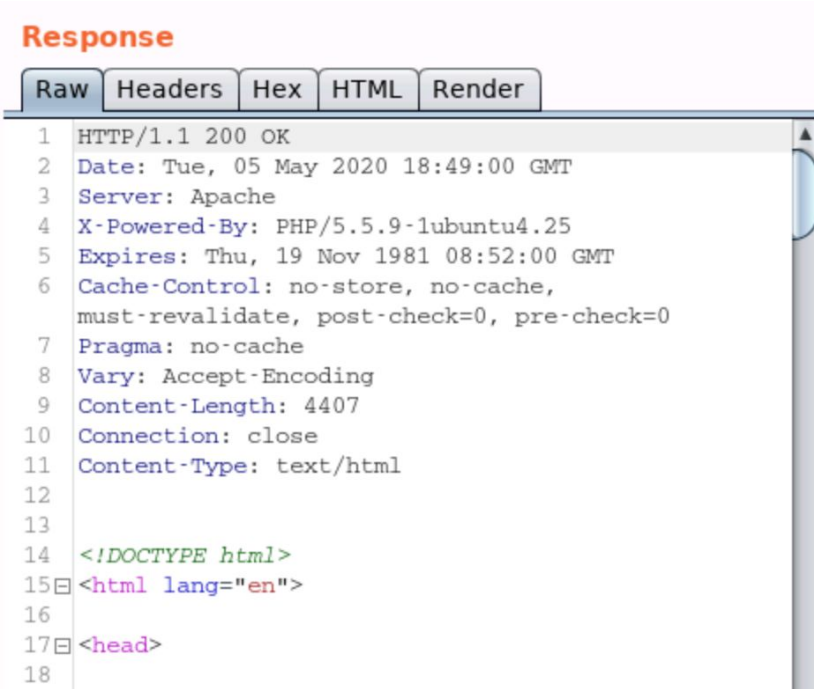
Repeater Tab:



Step 4: Sending GET Request.

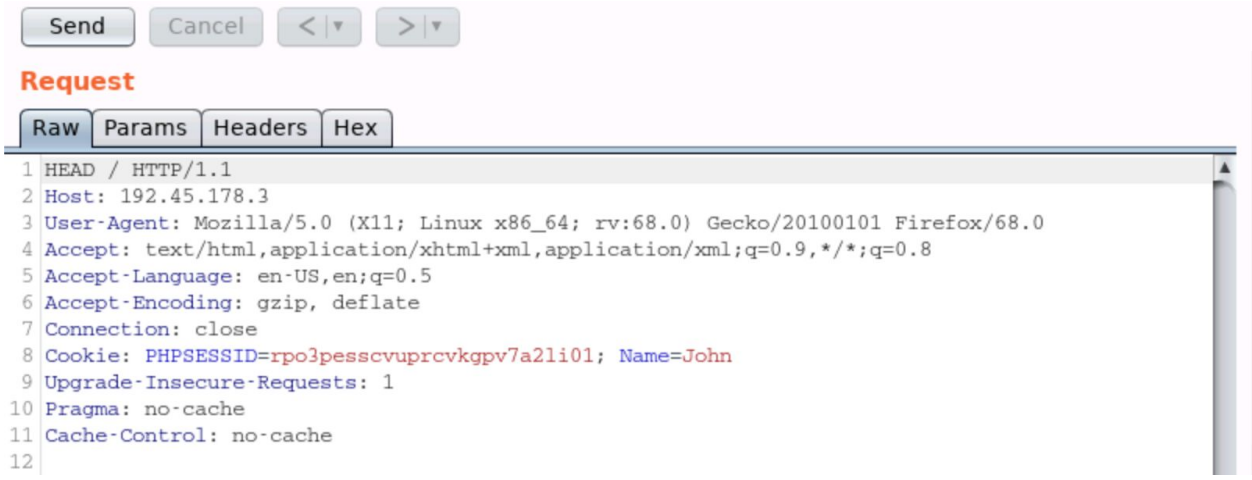
Repeater Tab:

Response Tab:



Step 5: Sending HEAD Request

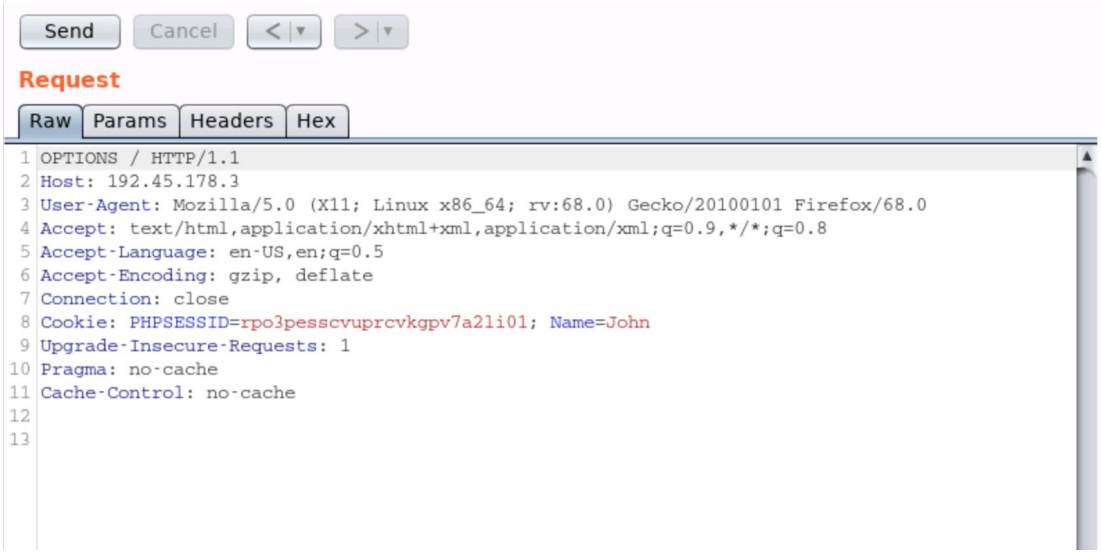
Request Tab:



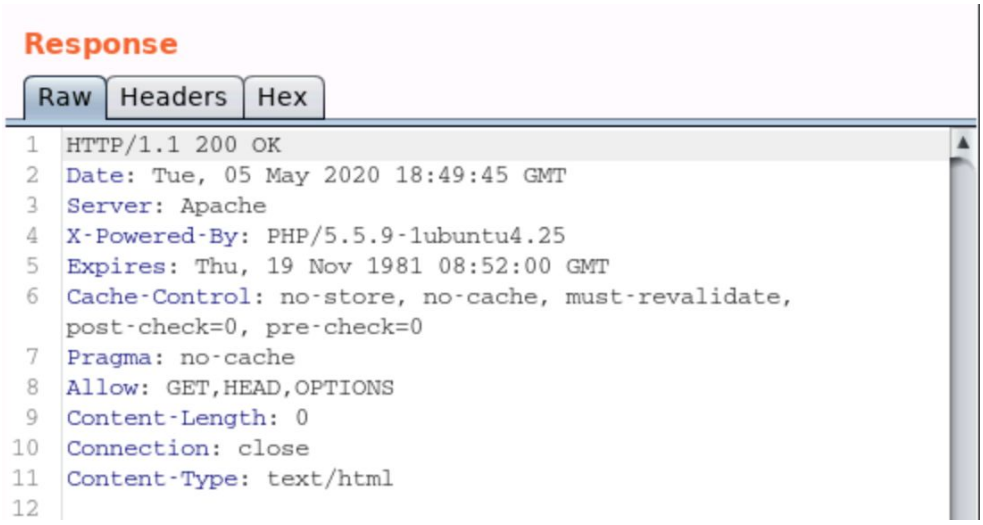
Response Tab:

Step 6: Sending OPTIONS request.

Request Tab:

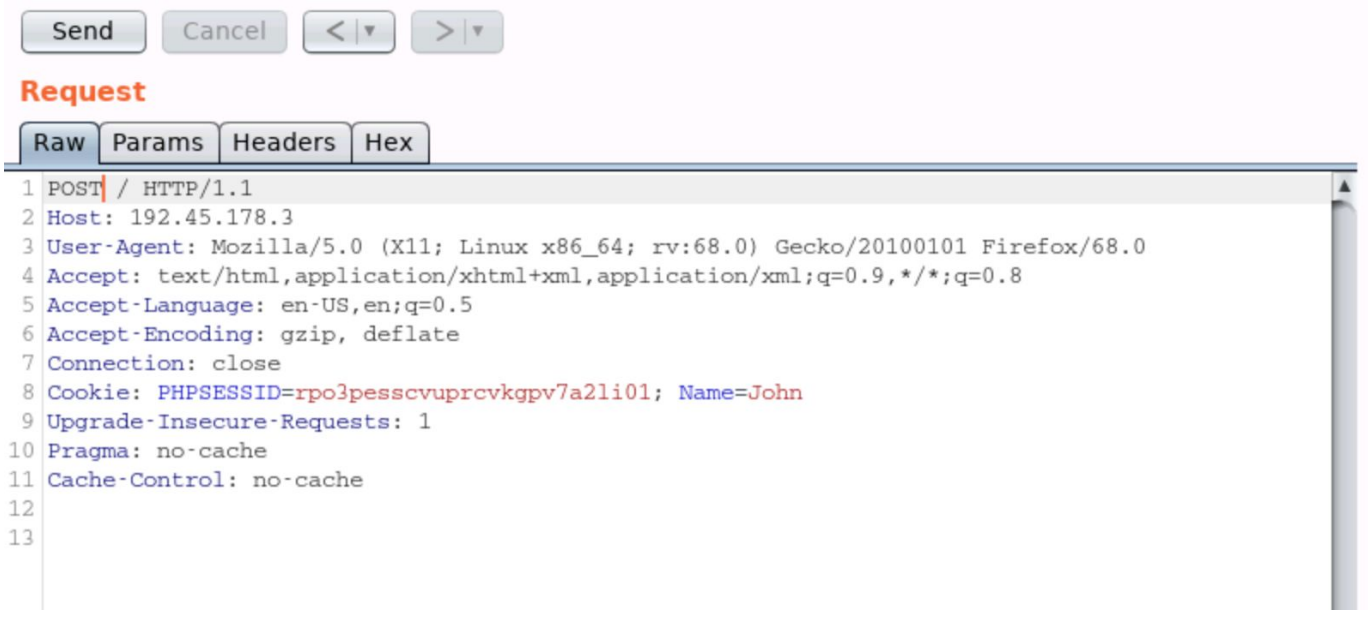


Response Tab:

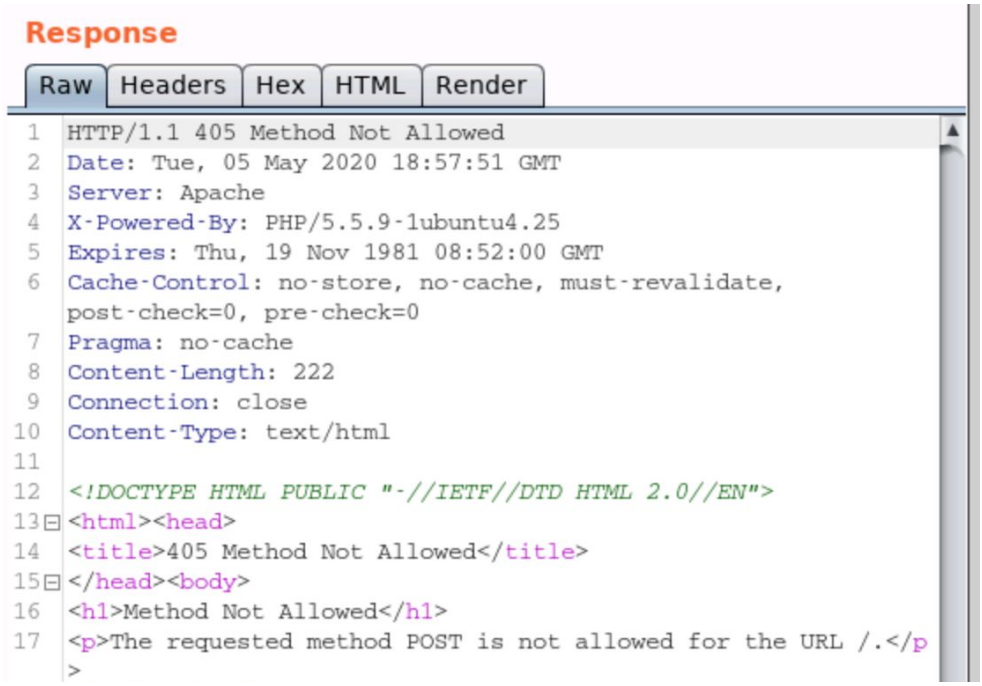


Step 7: Sending POST request.

Request Tab:



Response Tab:

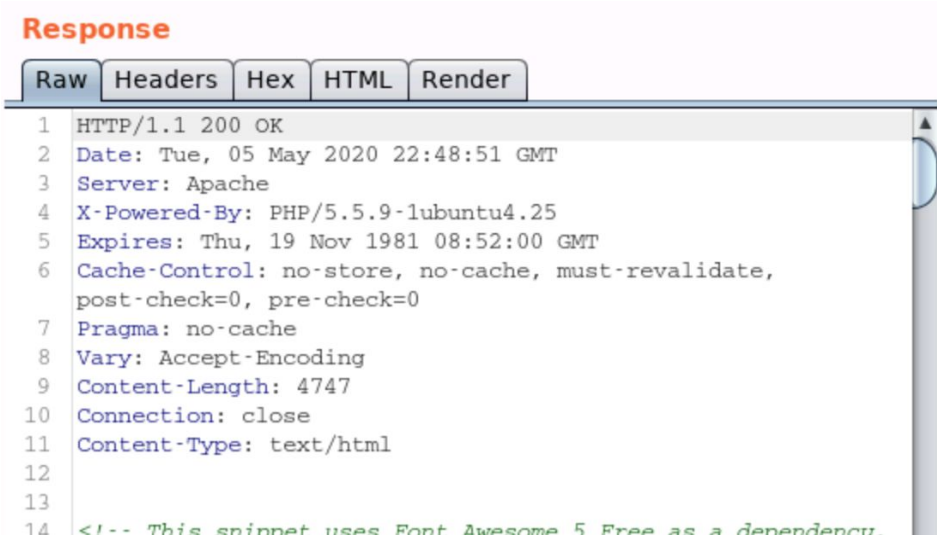


POST method is not allowed.

Step 8: Sending POST request to login.php with incorrect login credentials.

Request Tab:

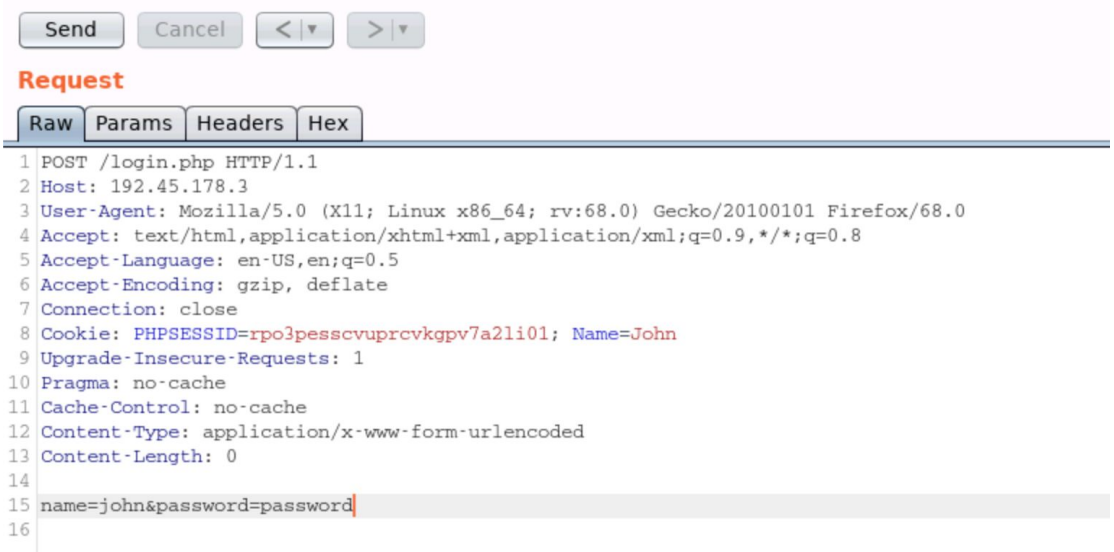
Response Tab:



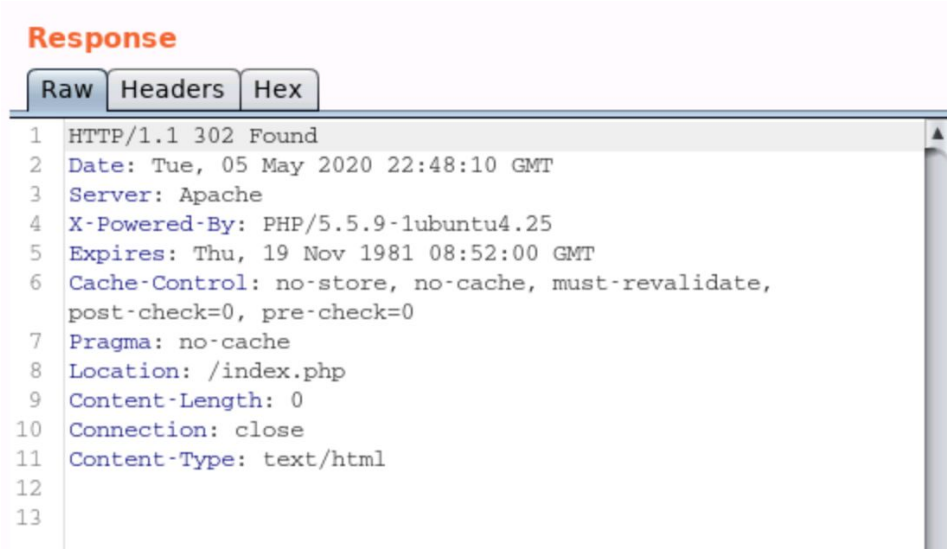
200 OK response is received.

Step 9: Sending POST request with valid login credentials.

Request Tab:



Response Tab:

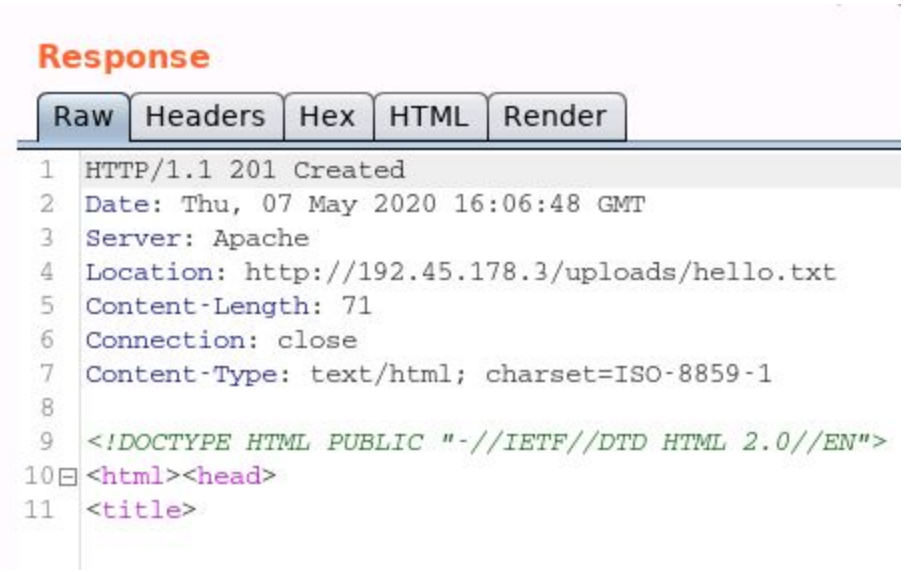


The login credentials were correct and as a result 302 response was received to index.php.

Step 10: Uploading file with PUT method

Request Tab:

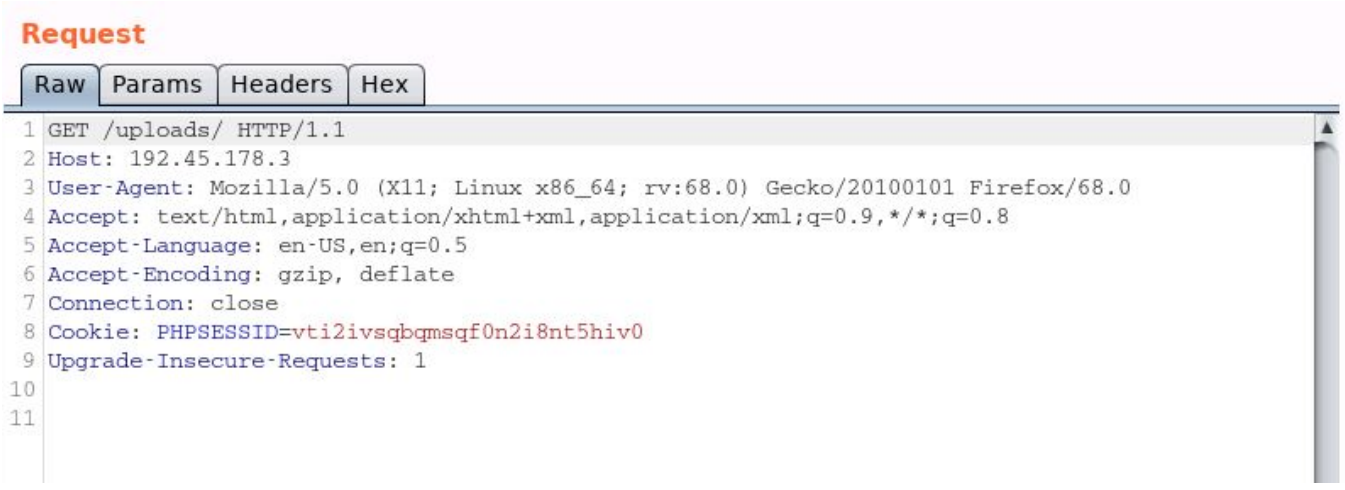
Response Tab:



The file was uploaded Successfully.

Check the files in /uploads directory.

Request Tab:



Response Tab:

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Date: Thu, 07 May 2020 16:07:26 GMT
Server: Apache
Vary: Accept-Encoding
Content-Length: 866
Connection: close
Content-Type: text/html;charset=UTF-8

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /uploads</title>
 </head>
 <body>
<h1>Index of /uploads</h1>
  <table>
   <tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th><a href="?C=N;O=D">Name
</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a
href="?C=D;O=A">Description</a></th></tr>
   <tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><img src="/icons/back.gif" alt="[PARENTDIR]"></td><td><a href="/">Parent
Directory</a></td><td> </td><td align="right">  - </td><td> </td></tr>
<tr><td valign="top"><img src="/icons/text.gif" alt="[TXT]"></td><td><a href="hello.txt">
hello.txt</a></td><td align="right">2020-05-07 16:06  </td><td align="right"> 11 </td><td> 
</td></tr>
```

The file "hello.txt" was uploaded successfully.

Checking Content of uploaded file.

Request Tab:

Request

Raw | Params | Headers | Hex

```
GET /uploads/hello.txt HTTP/1.1
Host: 192.45.178.3
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=vti2ivsqbqmsqf0n2i8nt5hiv0
Upgrade-Insecure-Requests: 1


```

Response Tab:

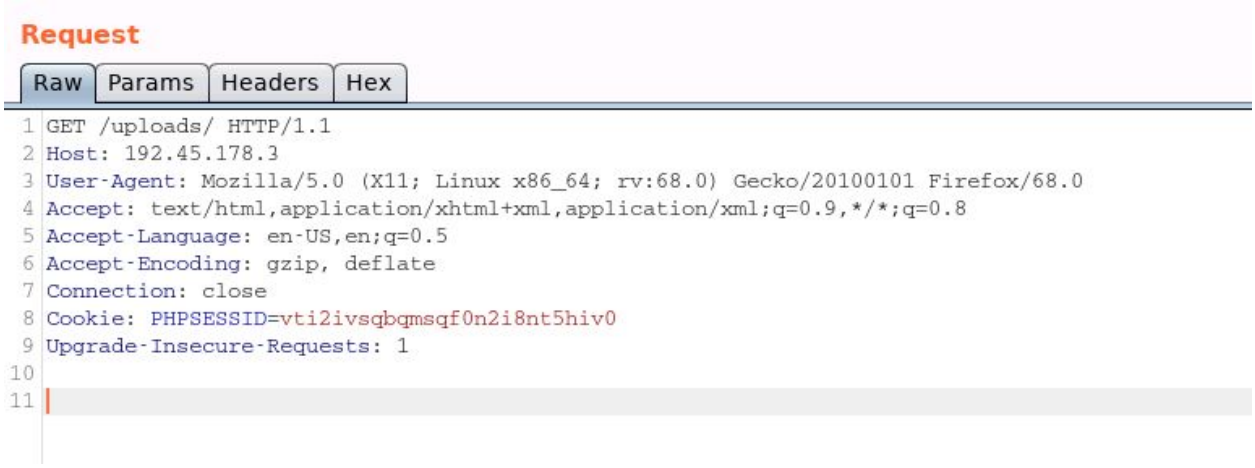
Step 11: Deleting the File.

Request Tab:

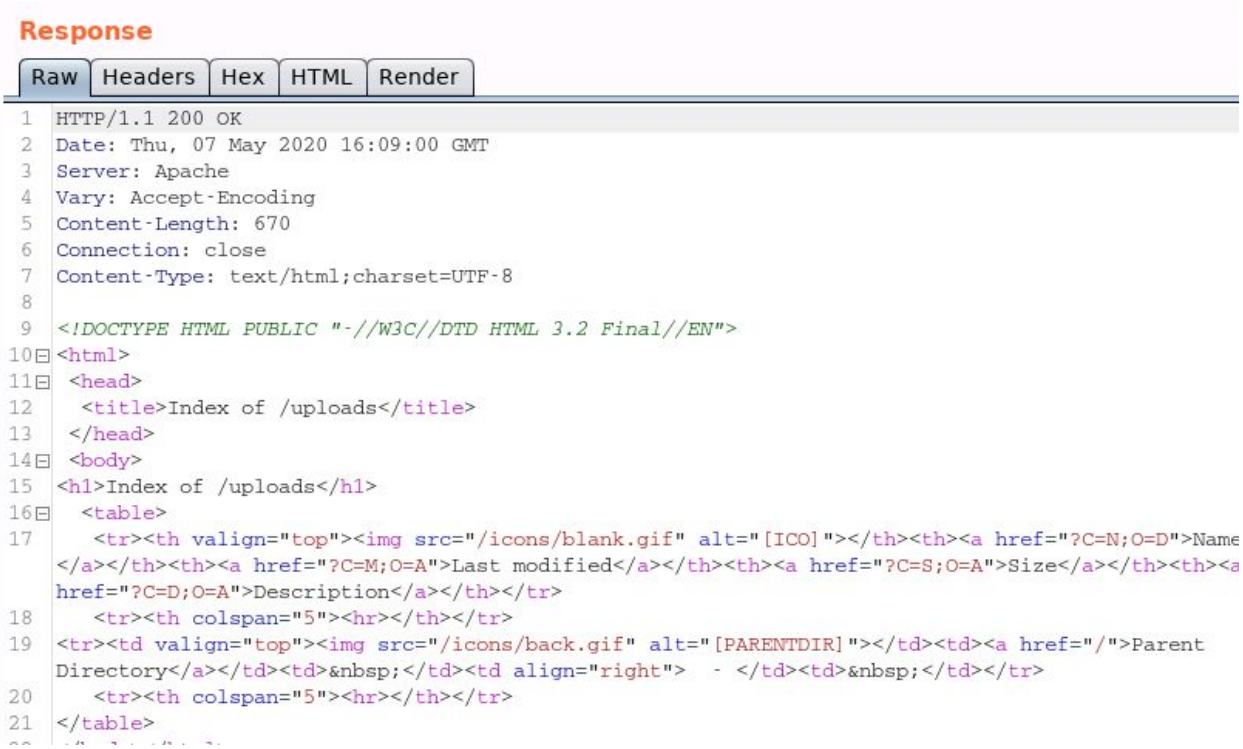
Response Tab:

The file was deleted. Check the files in the uploads directory.

Request Tab:



Response Tab:



References:

1. Curl (https://linux.die.net/man/1/curl)
2. Burp Suite (https://portswigger.net/burp/documentation/desktop/getting-started)
3. Dirb (https://tools.kali.org/web-applications/dirb)