



Instituto Tecnológico y de Estudios Superiores de Monterrey
Escuela de Ingeniería y Ciencias
Ingeniería en Ciencias de Datos y Matemáticas

Aplicación de Criptografía y Seguridad

Actividad 3.2.6 Laboratorio HTTP Method Enumeration

Samantha Ruelas Valtierra	A01704564
Ángel David Ávila Pérez	A01562833
Héctor David Bahena Garza	A01284661
Manzur Macías Pineda	A01198234
María Fernanda Lee Ponce	A00830974
Gonzalo Garza Moreno	A01284950

Profesores Óscar E. Labrada Gómez y Alberto F. Martínez Herrera

Socio Formador IPC Services

30/08/2023

Monterrey, Nuevo León

Actividad 3.2.5 Laboratorio Privilege Escalation I (AppArmor)

ÍNDICE:

Introducción.....	3
Objetivo.....	3
Procedimiento.....	3
- Acceso al laboratorio.....	3
- Inspeccionar la aplicación web.....	5
- Utilizar dirb para identificar directorios ocultos.....	8
- Interactuar con la página de inicio con CURL.....	9
- Interactuar con la página login.php con CURL.....	11
- Interactuar con la página post.php con CURL.....	13
- Interactuar con el directorio de cargas.....	14
- Interactuar con la página web con Burp Suite.....	17
Conclusión:.....	31

Introducción

Checar los métodos HTTP permitidos en un sitio es de gran importancia al considerar su seguridad contra ataques cibernéticos. No controlar qué métodos están permitidos puede llegar a ocasionar denegación de servicios, ejecución remota de código, entre otros ataques.

Objetivo

Utilizar *Burp Suite* y *Curl* para enumerar los métodos HTTP permitidos por la página/directorio web.

Procedimiento

- Acceso al laboratorio

En la página de <https://attackdefense.com/freelabs>.com insertamos el título del laboratorio en el cual se va a trabajar “HTTP Method Enumeration”.

The screenshot shows the AttackDefense.com Freelabs dashboard. On the left, there's a sidebar with navigation links like Dashboard, Search, Ongoing Labs (0), Latest Additions, and Community Labs. Below that are sections for EARN CREDENTIALS (Verifiable Badges), WINDOWS SECURITY (Reconnaissance, Basic Exploitation, Post Exploitation, Service Exploitation, Privilege Escalation, Maintaining Access), and CLOUD SECURITY. The main area has a greeting "Hi, A01284***.", a welcome message "Welcome to PentesterAcademy Attack-Defense Labs!", and a search bar with "Search over 2000+ Labs!" and "HTTP Method Enumeration" entered. There are four large cards: "Total Labs" (2399), "Ongoing Labs" (0), "New this Month" (0), and "Total Labs Played" (583917). Below these are sections for "Latest Labs!", "Site Activity Today", and "New Members Today". The "Latest Labs!" section lists "Dolibarr SQLi (CVE-2018-10094) in cve-2018" and "SuiteCRM Auth SQLi (CVE-...)".

Fig. 1: “Búsqueda de laboratorio HTTP Method Enumeration”. Fuente: Elaboración Propia.

Abrimos el laboratorio y podemos ver la página inicial del laboratorio:

The screenshot shows the Attack Defense platform interface. On the left, there's a sidebar with navigation links like Dashboard, Search, Ongoing Labs, Latest Additions, Community Labs, EARN CREDENTIALS, Verifiable Badges, WINDOWS SECURITY, CLOUD SECURITY, and LINUX SECURITY. The main content area is titled "HTTP Method Enumeration" and shows a sub-section for "webapp-web-app-basics-basics" at "Level: Easy". It includes a "Run" button, a "Server: US-East" dropdown, and a "Lab Scoreboard" section with statistics: 4339 # Played on AD and 0 # Played by you. Below this is a network diagram showing a "Target" server connected to a "switch", which is then connected to a user icon labeled "You". At the bottom, there are tabs for Mission, Prohibited Activities, and Technical Support, along with a "Download Lab Manual" button.

Fig. 2: “Página inicial del laboratorio HTTP Method Enumeration”. Fuente: Elaboración Propia.

Esta actividad se llevará a cabo utilizando el manual proporcionado por Attack Defense. El manual contiene varios sub-objetivos con sus pasos, con el propósito de aprender cómo utilizar *burp suite* y *curl* para enumerar los métodos HTTP utilizados por varias páginas web.

Se inicia el laboratorio presionando el botón de “Run” y después de cargarse, abriendo el Laboratorio haciendo clic en el botón de Lab Link.

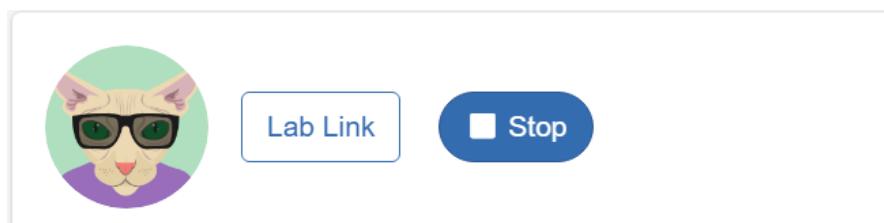
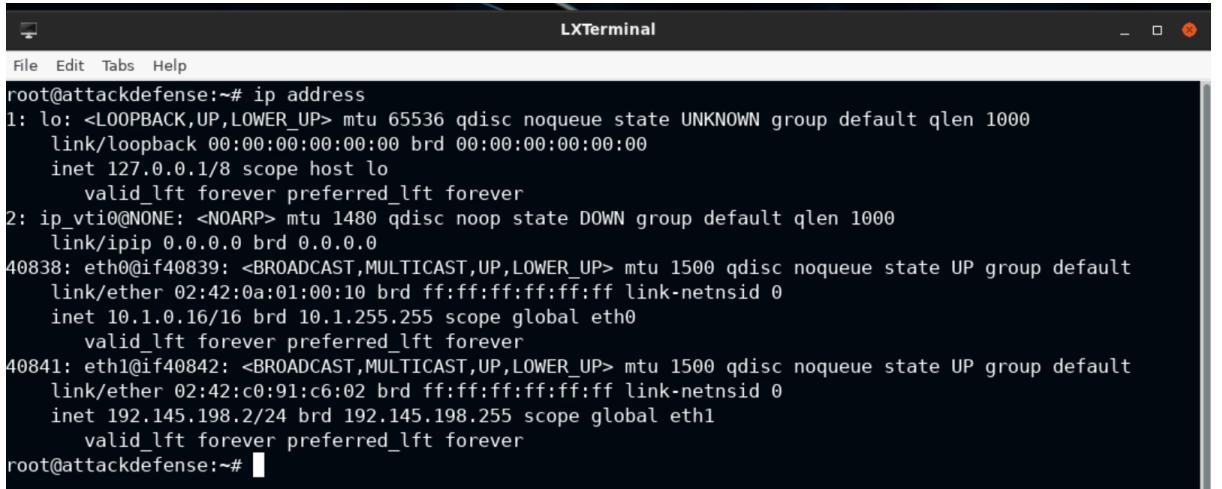


Fig. 3: “Botones que muestran que el laboratorio HTTP Method Enumeration está corriendo”. Fuente: Elaboración Propia.

- **Inspeccionar la aplicación web.**

Para abrir la aplicación web, lo primero que tenemos que hacer es buscar su dirección ip en la terminal para poder abrir la página en Mozilla Firefox.

Comando: ip address



```

root@attackdefense:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
40838: eth0@if40839: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:00:10 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.0.16/16 brd 10.1.255.255 scope global eth0
        valid_lft forever preferred_lft forever
40841: eth1@if40842: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:91:c6:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.145.198.2/24 brd 192.145.198.255 scope global eth1
        valid_lft forever preferred_lft forever
root@attackdefense:~#

```

Fig. 4: “Dirección ip”, Fuente: Elaboración Propia.

Insertamos la dirección ip en el buscador de Mozilla Firefox y se abre la página de *Attack Defense Demo Blog*.

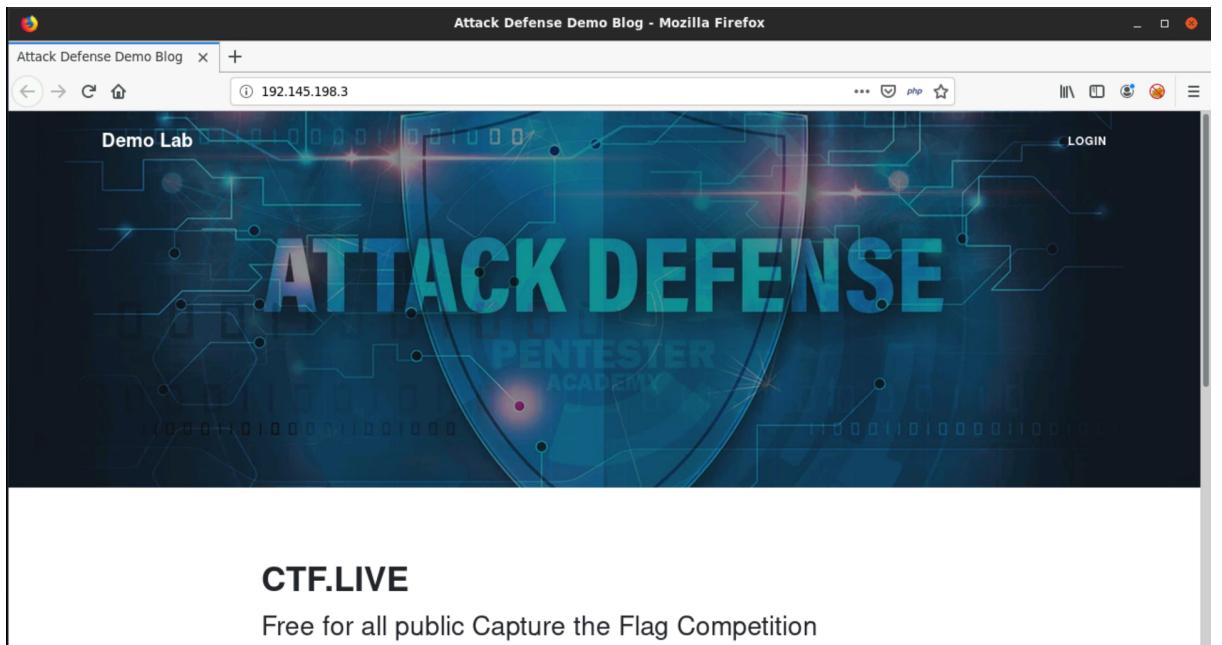


Fig. 5: “Attack Defense Demo Blog index”, Fuente: Elaboración Propia.

- **Paso 1:** Siguiendo Links

Se hace clic en Login, que nos redirige a <http://192.145.198.3/login.php>.

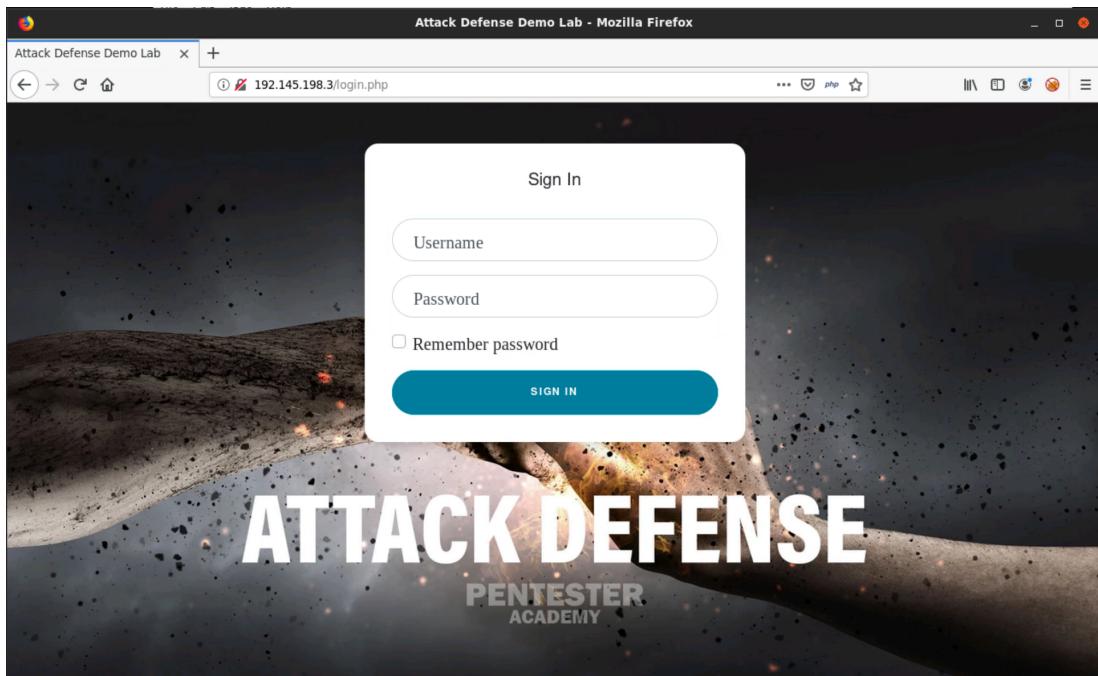


Fig. 6: “Attack Defense Demo Blog login”, Fuente: Elaboración Propia.

- **Paso 2:** Identificación del endpoint que procesa los campos

Se hace clic derecho en la página web y se selecciona *View Page Source*, haciendo scroll a la página se puede ver que los parámetros se pasan en el método *POST* a la página de “login.php”.

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">
        <div class="card card-signin my-5">
          <div class="card-body">
            <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>
        </div>
      </div>
    </div>
  </div>
</body>
```

Fig. 7: “Attack Defense Demo Blog Page Source”, Fuente: Elaboración Propia.

- **Paso 3:** Iniciar sesión

Utilizando las credenciales proporcionadas por Attack Defense, hacemos login en la aplicación web.

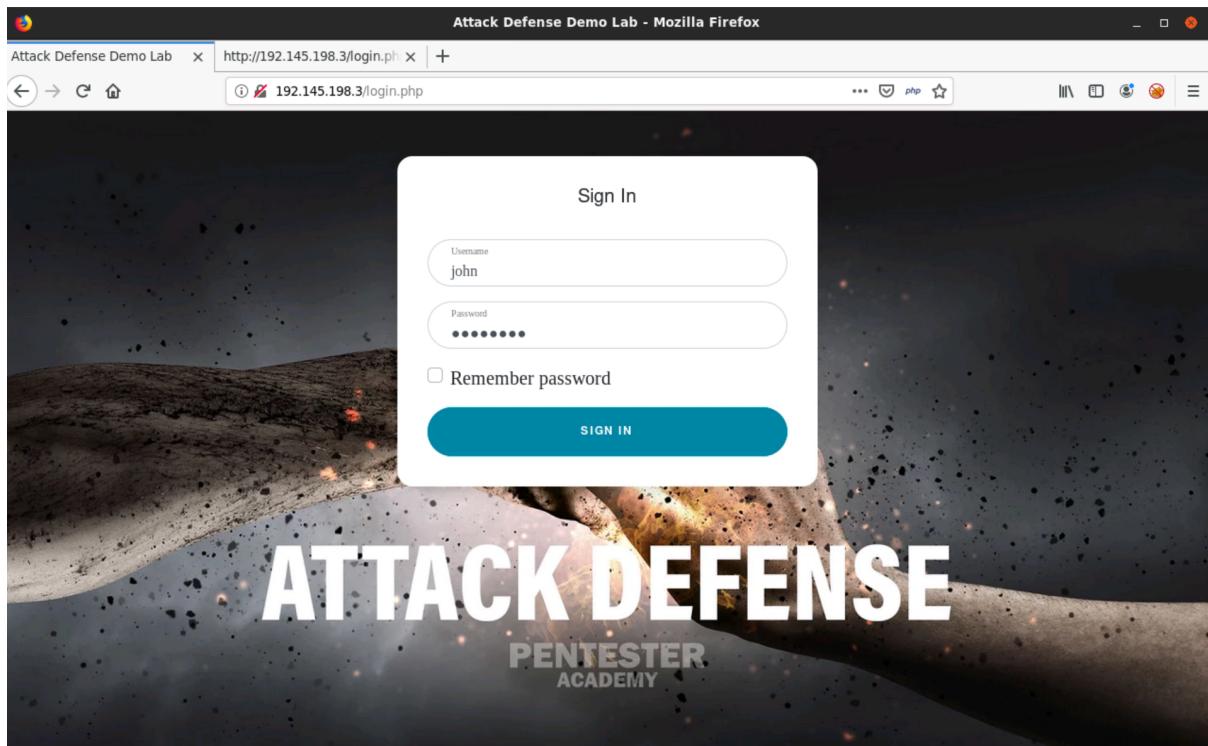


Fig. 8: “Attack Defense Demo Blog login con credenciales”, Fuente: Elaboración Propia.

- **Paso 4:** Abrir el blog post.

Abrimos el otro link disponible que es el blog “Attack Defense Lab Overview”.

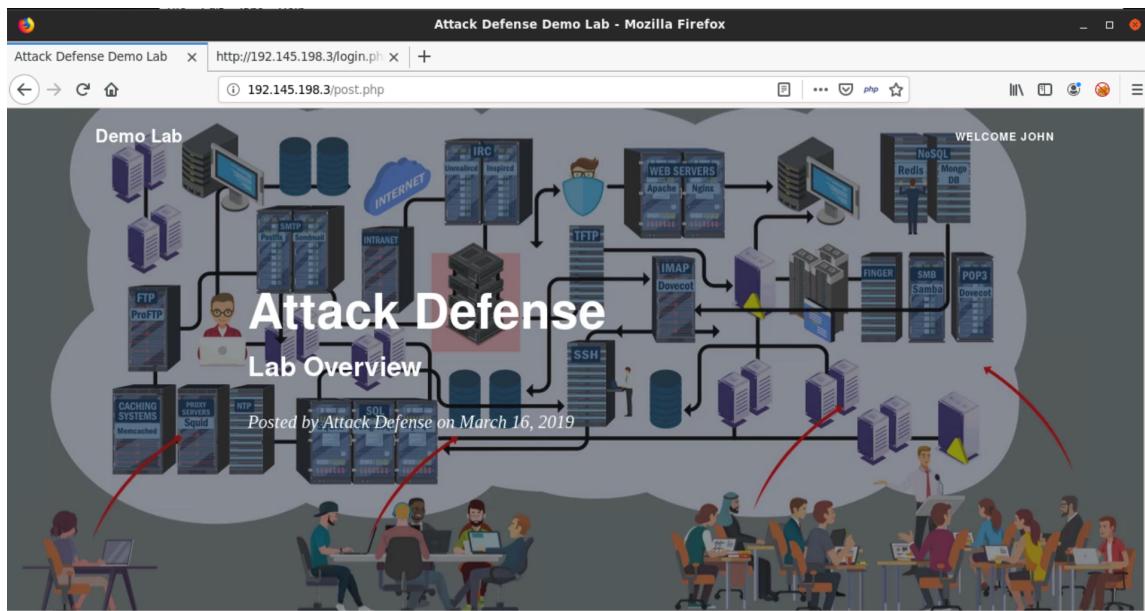


Fig. 9: “Attack Defense Demo Blog post”, Fuente: Elaboración Propia.

Podemos ver que sí iniciamos sesión en el paso anterior por el mensaje de “Welcome John” en la esquina superior derecha. También cabe notar que las 3 páginas web que podemos acceder en el sitio son: index.php, login.php y post.php

- **Utilizar dirb para identificar directorios ocultos**

Comando: dirb <http://192.145.198.3>

```
---- Scanning URL: http://192.145.198.3/ ----
+ http://192.145.198.3/.git/HEAD (CODE:200|SIZE:23)
+ http://192.145.198.3/cgi-bin/ (CODE:403|SIZE:210)
=> DIRECTORY: http://192.145.198.3/css/
=> DIRECTORY: http://192.145.198.3/img/
+ http://192.145.198.3/index.php (CODE:200|SIZE:4408)
=> DIRECTORY: http://192.145.198.3/js/
+ http://192.145.198.3/LICENSE (CODE:200|SIZE:10273)
=> DIRECTORY: http://192.145.198.3/mail/
+ http://192.145.198.3/phpinfo.php (CODE:200|SIZE:74271)
+ http://192.145.198.3/server-status (CODE:403|SIZE:215)
=> DIRECTORY: http://192.145.198.3/uploads/
=> DIRECTORY: http://192.145.198.3/vendor/

---- Entering directory: http://192.145.198.3/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.145.198.3/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.145.198.3/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.145.198.3/mail/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.145.198.3/uploads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.145.198.3/vendor/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
          (Use mode '-w' if you want to scan it anyway)
```

Fig. 10: “Directorio de la página Attack Defense Demo Blog”, Fuente: Elaboración Propia.

Podemos ver que los directorios presentes en el servidor son *css*, *img*, *js*, *mail*, *uploads* y *vendor*.

- ***Interactuar con la página de inicio con CURL***

- **Paso 1:** Mandar un GET request

Comando: curl -X GET 192.145.198.3

```
root@attackdefense:~# curl -X GET 192.145.198.3
<!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>

<body>

    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-light fixed-top" id="mainNav">
        <div class="container">
            <a class="navbar-brand" href="index.php">Demo Lab</a>
            <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
```

Fig. 11: “GET request a Attack Defense Demo Blog”, Fuente: Elaboración Propia.

- **Paso 2:** Mandar un HEAD request

Comando: curl -I 192.145.198.3

```
root@attackdefense:~# curl -I 192.145.198.3
HTTP/1.1 200 OK
Date: Sat, 26 Aug 2023 17:49:56 GMT
Server: Apache
X-Powered-By: PHP/5.5.9-1ubuntu4.25
Set-Cookie: PHPSESSID=pmegce9gc0d4qo4ev7453618a6; 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
```

Fig. 12: “HEAD request a Attack Defense Demo Blog”, Fuente: Elaboración Propia.

- **Paso 3:** Mandar un OPTIONS request

Comando: curl -X OPTIONS 192.145.198.3

```
root@attackdefense:~# curl -X OPTIONS 192.145.198.3
root@attackdefense:~#
root@attackdefense:~# curl -X OPTIONS 192.145.198.3 -v
*   Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> OPTIONS / HTTP/1.1
> Host: 192.145.198.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 26 Aug 2023 17:54:04 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=7nbo4n95o0a7b6d0tntbvq2b32; 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 192.145.198.3 left intact
```

Fig. 13: “OPTIONS request a Attack Defense Demo Blog”, Fuente: Elaboración Propia.

Esto nos indica que los únicos métodos permitidos para usar en la página web son GET, HEAD y OPTIONS.

- **Paso 4:** Mandar un POST request

Comando: curl -X POST 192.145.198.3

```
root@attackdefense:~# curl -X POST 192.145.198.3
<!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>
```

Fig. 14: “POST request a Attack Defense Demo Blog”, Fuente: Elaboración Propia.

No se nos permite utilizar el método POST.

- **Paso 5:** Mandar un PUT request.

Comando: curl -X PUT 192.145.198.3

```
root@attackdefense:~# curl -X PUT 192.145.198.3
<!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>
```

Fig. 15: “PUT request a Attack Defense Demo Blog”, Fuente: Elaboración Propia.

No se nos permite utilizar el método PUT.

- **Interactuar con la página login.php con CURL**

- **Paso 1:** Mandar un OPTIONS request

Comando: curl -X OPTIONS 192.145.198.3/login.php

```

root@attackdefense:~# curl -X OPTIONS 192.145.198.3/login.php -v
*   Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> OPTIONS /login.php HTTP/1.1
> Host: 192.145.198.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 26 Aug 2023 18:02:47 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=q95vdusqn9ivepbpbku8gd8uj4; 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

```

Fig. 16: “OPTIONS request a Attack Defense Demo Blog login”, Fuente: Elaboración Propia.

En esta página sí tenemos permitido utilizar el método POST.

- **Paso 2:** Mandar un POST request.

Comando: curl -X POST 192.145.198.3/login.php

```

root@attackdefense:~# curl -X POST 192.145.198.3/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;
}

```

Fig. 17: “POST request a Attack Defense Demo Blog login”, Fuente: Elaboración Propia.

- **Paso 3:** Pasar el usuario y contraseña a la página login.php.

Comando: curl -X POST 192.145.198.3/login.php -d

"name=john&password=password" -v

```
root@attackdefense:~# curl -X POST 192.145.198.3/login.php -d "name=john&password=password" -v
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> POST /login.php HTTP/1.1
> Host: 192.145.198.3
> 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: Sat, 26 Aug 2023 18:07:54 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=5bttv2i2hmp3h9b3lnpflovr83; 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=Mon, 25-Sep-2023 18:07:54 GMT; Max-Age=2592000; path=/
< Location: /index.php
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 192.145.198.3 left intact
```

Fig. 18: “POST request a Attack Defense Demo Blog login con credenciales”,
Fuente: Elaboración Propia.

- ***Interactuar con la página post.php con CURL***

- **Paso 1:** Mandar un OPTIONS request

curl -X OPTIONS 192.145.198.3/post.php

```

root@attackdefense:~# curl -X OPTIONS 192.145.198.3/post.php -v
*   Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> OPTIONS /post.php HTTP/1.1
> Host: 192.145.198.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 26 Aug 2023 18:22:48 GMT
< Server: Apache
< X-Powered-By: PHP/5.5.9-1ubuntu4.25
< Set-Cookie: PHPSESSID=mjpi7vv0haq5be6qcjd6378j71; 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 192.145.198.3 left intact

```

Fig. 19: “OPTIONS request a Attack Defense Demo Blog post”, Fuente: Elaboración Propia

Tiene los mismos permisos que login.php

- ***Interactuar con el directorio de cargas***
- **Paso 1:** Checar el contenido en el directorio /uploads

Escribimos 198.145.178.3/uploads en el buscador de Mozilla Firefox.

Name	Last modified	Size	Description
Parent Directory		-	

Fig. 20: “Attack Defense Demo Blog uploads”, Fuente: Elaboración Propia

- **Paso 2:** Mandar OPTIONS request al directorio /uploads

Comandos: curl -X OPTIONS 192.145.198.3/uploads/

```
curl -X OPTIONS 192.145.198.3/uploads/ -v
```

```
root@attackdefense:~# curl -X OPTIONS 192.145.198.3/uploads/
root@attackdefense:~#
root@attackdefense:~# curl -X OPTIONS 192.145.198.3/uploads/ -v
*   Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> OPTIONS /uploads/ HTTP/1.1
> Host: 192.145.198.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Sat, 26 Aug 2023 18:34:36 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 192.145.198.3 left intact
```

Fig. 21: “OPTIONS request a Attack Defense Demo Blog uploads”, Fuente: Elaboración Propia

El Webdav está permitido en el Apache Server, permitiendo la subida de archivos con el método PUT.

- **Paso 3:** Subir un archivo con el método PUT

Comandos: echo “Hello World” > hello.txt

```
curl 192.145.198.3/uploads/ --upload-file hello.txt
```

```
root@attackdefense:~# echo "Hello World" > hello.txt
root@attackdefense:~# curl 192.145.198.3/uploads/ --upload-file hello.txt
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload Upload Total Spent   Left Speed
0       0     0      0      0       0      0 ---:---:---:--- 0<!DOCTYPE HTML PUBLIC "-//IE
TF//DTD HTML 2.0//EN">
<html><head>
100  83 100    71 100     12 14200   2400 ---:---:---:--- 16600
```

Fig. 22: “Subida de archivo con PUT a Attack Defense Demo Blog uploads”, Fuente: Elaboración Propia

- **Paso 4:** Checar el contenido del directorio /uploads

Name	Last modified	Size	Description
Parent Directory	-		
hello.txt	2023-08-26 18:39	12	

Fig. 23: “Attack Defense Demo Blog uploads con nuevo archivo”, Fuente: Elaboración Propia

Podemos ver el archivo que subimos.

- **Paso 5:** Usar el método DELETE para borrar el archivo.

Comando: curl -XDELETE 192.145.198.3/uploads/hello.txt

```
root@attackdefense:~# curl -X DELETE 192.145.198.3/uploads/hello.txt -v
*   Trying 192.145.198.3:80...
* TCP_NODELAY set
* Connected to 192.145.198.3 (192.145.198.3) port 80 (#0)
> DELETE /uploads/hello.txt HTTP/1.1
> Host: 192.145.198.3
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404 Not Found
< Date: Sat, 26 Aug 2023 18:44:28 GMT
< Server: Apache
< Content-Length: 215
< Content-Type: text/html; charset=iso-8859-1
```

Fig. 24: “Eliminación del archivo en Attack Defense Demo Blog uploads con DELETE”, Fuente: Elaboración Propia

- **Paso 6:** Checar el contenido del directorio /uploads

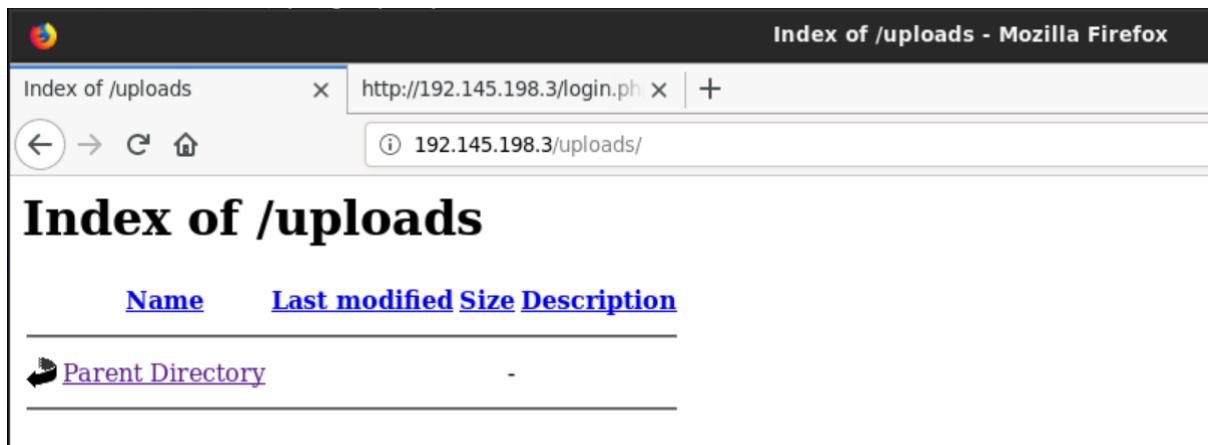
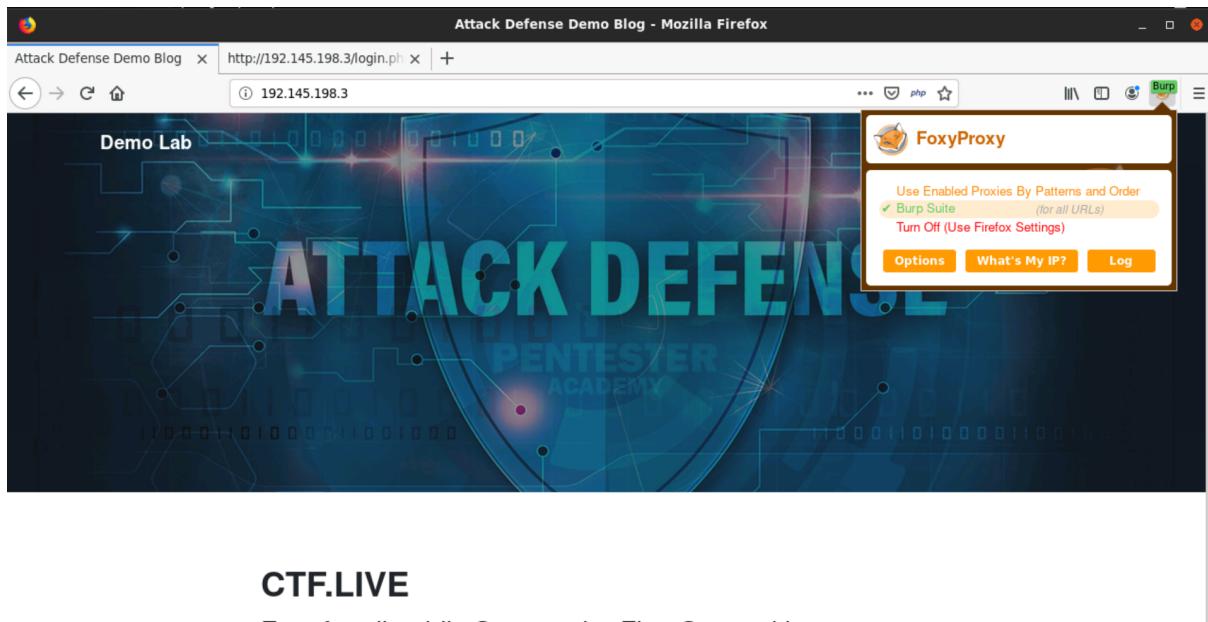


Fig. 25: “Attack Defense Demo Blog uploads sin el nuevo archivo”, Fuente: Elaboración Propia

El archivo fue eliminado.

- **Interactuar con la página web con Burp Suite**
- **Paso 1:** Establecer el *FoxyProxy* para usar el *burp proxy*.

Hacemos clic en el ícono de *Fox* y seleccionamos *Burp Suite*.



CTF.LIVE

Free for all public Capture the Flag Competition

Posted by Attack Defense on March 29, 2020

Fig. 26: “Cambio en FoxyProxy a usar Burp Suite”, Fuente: Elaboración Propia

- **Paso 2:** Iniciar *burp suite*

Abrimos *burp suite* y volvemos a cargar la página web.

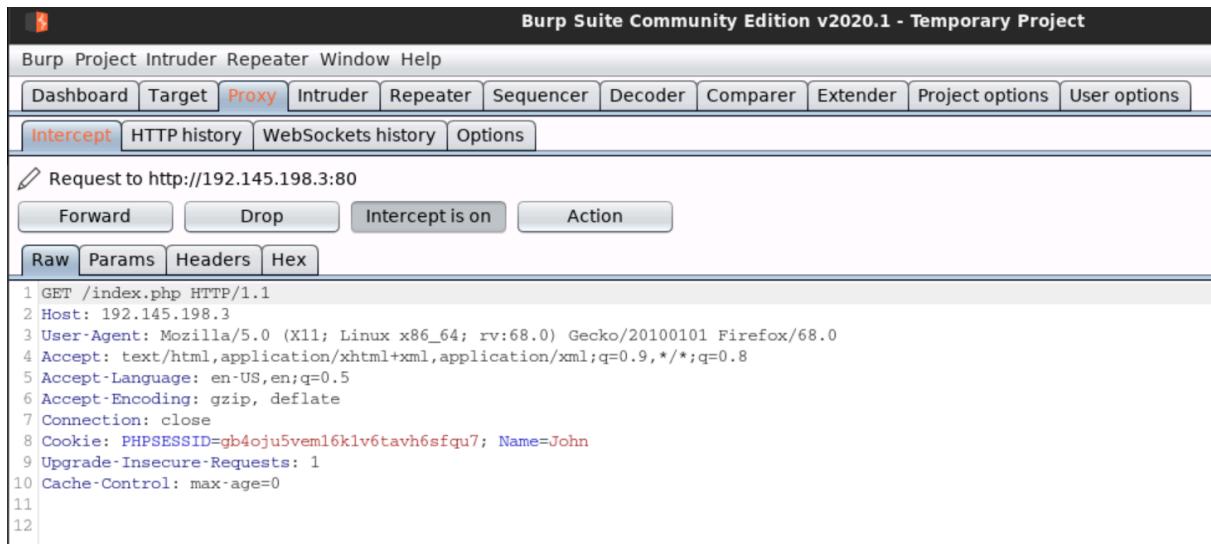


Fig. 27: “Página de Proxy en Burp Suite”, Fuente: Elaboración Propia

- **Paso 3: Mandar petición a Repeater**

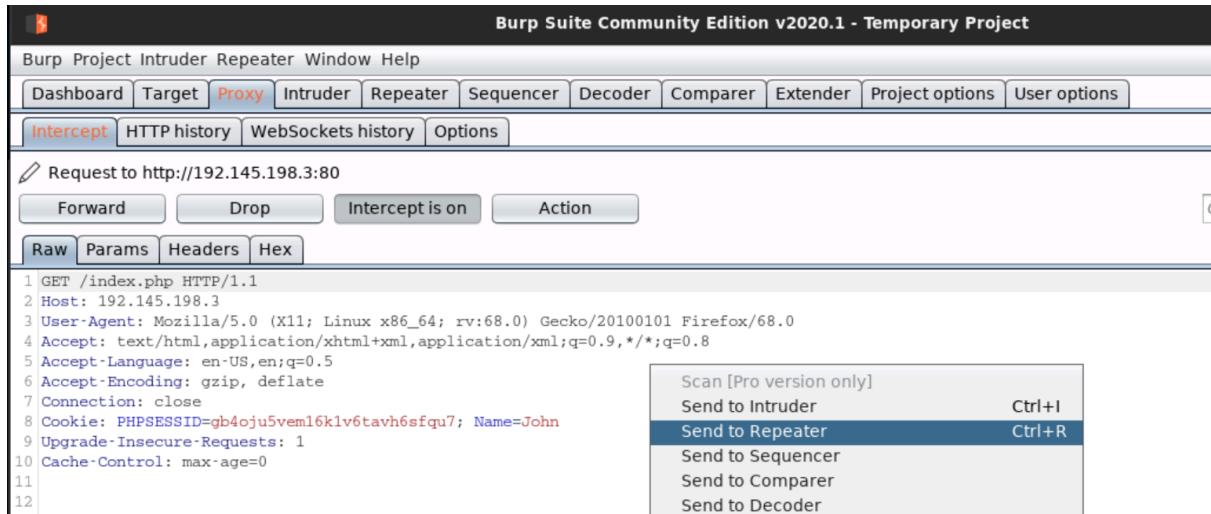


Fig. 28: “Enviar contenido de la página de Proxy a Repeater en Burp Suite”,
Fuente: Elaboración Propia

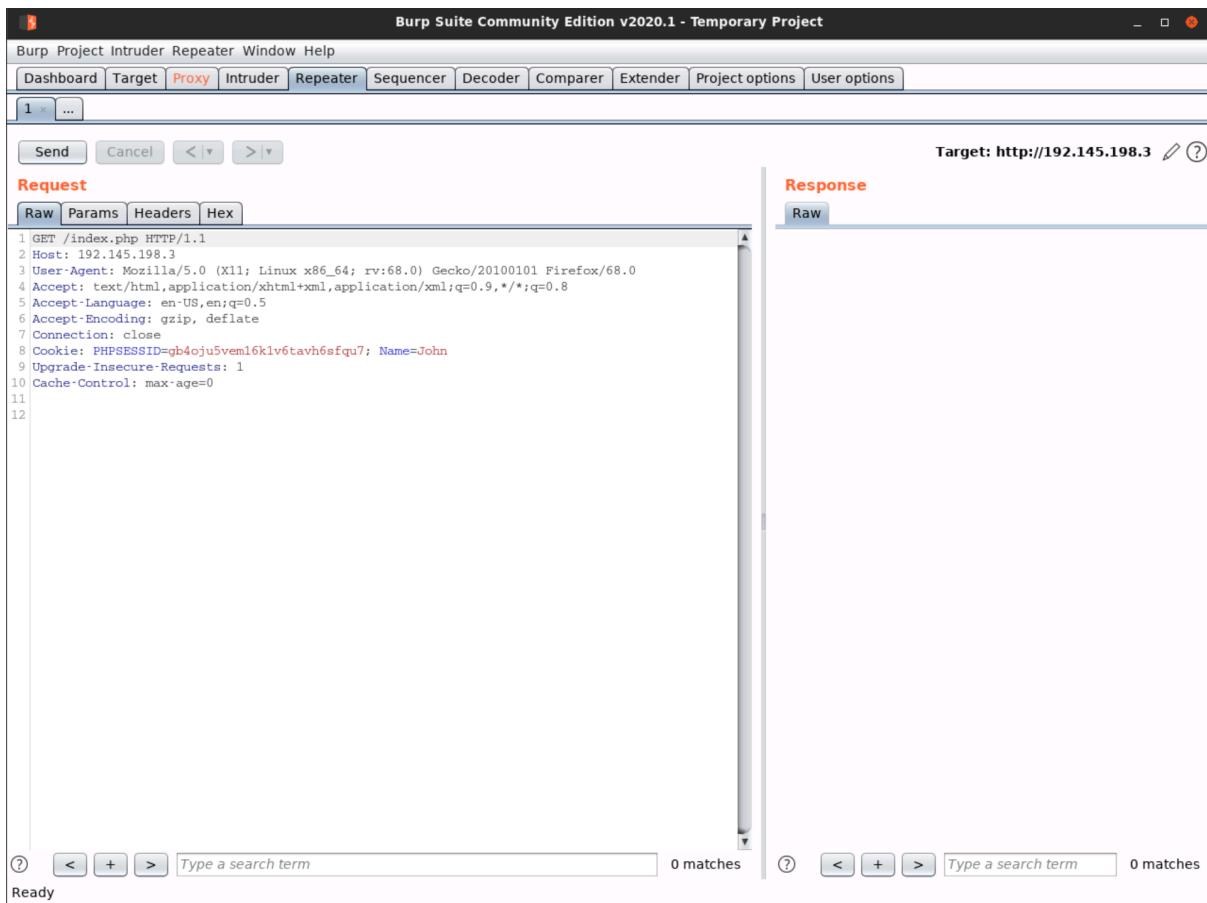


Fig. 29: “Página de Repeater en Burp Suite”, Fuente: Elaboración Propia

- **Paso 4: Mandar un GET request**

Hacemos clic en el botón de send.

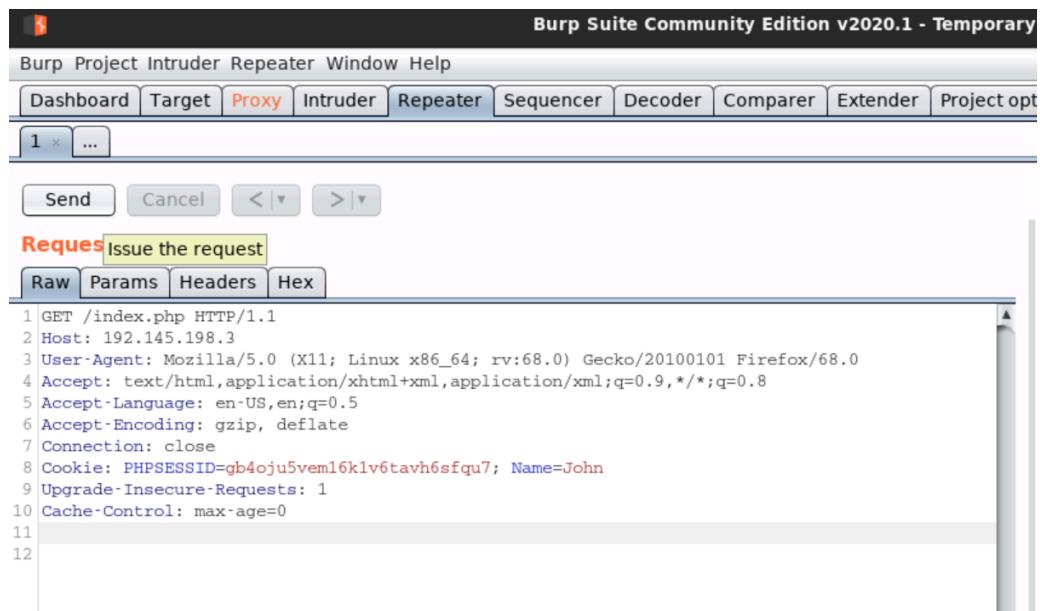


Fig. 30: “GET request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propria

Resultados:

The screenshot shows the Burp Suite interface with the target set to `http://192.145.198.3`. The response tab is selected, displaying the following HTTP response headers and body:

```
1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 19:13:55 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-lubuntu4.25
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 4415
10 Connection: close
11 Content-Type: text/html
12
13
14 <!DOCTYPE html>
15<html lang="en">
16
17<head>
18
```

Fig. 31: “Resultados de GET request a la página en Repeater de Burp Suite”,
Fuente: Elaboración Propia

- **Paso 5:** Mandar un HEAD request

Cambiamos la primera línea para hacer un HEAD request.

The screenshot shows the Burp Suite interface with the target set to `http://192.145.198.3`. The request tab is selected, displaying the following HEAD request:

```
Send Cancel < | > | ▾
Request
Raw Params Headers Hex
1 HEAD /index.php HTTP/1.1
2 Host: 192.145.198.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=gb4oju5vem16k1v6tavh6sfqu7; Name=John
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

Fig. 32: “HEAD request a la página en Repeater de Burp Suite”, Fuente:
Elaboración Propia

Resultados:

The screenshot shows the 'Response' tab in Burp Suite's Repeater tool. The response code is HTTP/1.1 200 OK. The response headers are as follows:

```
1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 19:16:54 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-1ubuntu4.25
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
7 Pragma: no-cache
8 Connection: close
9 Content-Type: text/html
10
11
```

Fig. 33: “Resultados de HEAD request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

- **Paso 6:** Mandar un OPTIONS request

Cambiamos la primera línea para hacer un OPTIONS request.

The screenshot shows the 'Raw' tab in Burp Suite's Repeater tool. The request line is set to 'OPTIONS /index.php HTTP/1.1'. The request headers are as follows:

```
1 OPTIONS /index.php HTTP/1.1
2 Host: 192.145.198.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=gb4oju5vem16k1v6tavh6sfqu7; Name=John
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

Fig. 34: “OPTIONS request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

Response

Raw

Headers

Hex

```
1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 19:18:46 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-1ubuntu4.25
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
7 Pragma: no-cache
8 Allow: GET,HEAD,OPTIONS
9 Content-Length: 0
10 Connection: close
11 Content-Type: text/html
12
13
```

Fig. 35: “Resultados de OPTIONS request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

- **Paso 7:** Mandar un POST request

Cambiamos la primera línea para hacer un POST request

Send

Cancel

< | ▾

▶ | ▾

Request

Raw

Params

Headers

Hex

```
1 POST /index.php HTTP/1.1
2 Host: 192.145.198.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=gb4oju5vem16k1v6tavh6sfqu7; Name=John
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

Fig. 36: “POST request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

Response

Raw

Headers

Hex

HTML

Render

```
1 HTTP/1.1 405 Method Not Allowed
2 Date: Sat, 26 Aug 2023 19:19:43 GMT
3 Server: Apache
4 Allow:
5 Content-Length: 231
6 Connection: close
7 Content-Type: text/html; charset=iso-8859-1
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
10<html><head>
11 <title>405 Method Not Allowed</title>
12</head><body>
13 <h1>Method Not Allowed</h1>
14 <p>The requested method POST is not allowed for the
URL /index.php.</p>
15 </body></html>
16
```

Fig. 37: “Resultados de POST request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

No podemos usar el método POST

- **Paso 8:** Mandar un POST request a login.php con credenciales de acceso incorrectas. *Se perdió la conexión al laboratorio y se tuvo que crear uno nuevo, que nos dió una dirección ip diferente.

The screenshot shows the Burp Suite interface with the 'Request' tab selected. At the top, there are buttons for 'Send', 'Cancel', and navigation arrows. Below the tabs, the raw request is displayed:

```
1 POST /login.php HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11 Content-Length: 28
12
13 name=john1&password=password
```

Fig. 38: “POST request con credenciales incorrectas al login de la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

The screenshot shows the Burp Suite interface with the 'Response' tab selected. At the top, it displays the target URL: **Target: http://192.228.111.3**. Below the tabs, the response headers are listed:

```
1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 20:24:32 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-1ubuntu4.25
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 4747
10 Connection: close
11 Content-Type: text/html
12
13
```

Fig. 39: “Resultados de POST request con credenciales incorrectas al login de la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Con el usuario incorrecto, nos regresa como respuesta “200 OK”

- **Paso 9:** Mandar un POST request con credenciales de acceso válidas

The screenshot shows the Burp Suite Repeater interface. At the top, there are buttons for 'Send', 'Cancel', and navigation arrows. Below that is a section titled 'Request' with tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected, displaying the following POST request:

```
1 POST /login.php HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11 Content-Length: 28
12
13 name=john&password=password
```

Fig. 40: “POST request con credenciales válidas al login de la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

The screenshot shows the Burp Suite Repeater interface with the 'Response' tab selected. Below it are tabs for 'Raw', 'Headers', and 'Hex'. The 'Headers' tab is selected, displaying the following response:

```
1 HTTP/1.1 302 Found
2 Date: Sat, 26 Aug 2023 20:46:05 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.5.9-1ubuntu4.25
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache,
must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Set-Cookie: Name=John; expires=Mon, 25-Sep-2023
20:46:05 GMT; Max-Age=2592000; path=/
9 Location: /index.php
10 Content-Length: 0
11 Connection: close
12 Content-Type: text/html
13
```

Fig. 41: “Resultados de POST request con credenciales válidas al login de la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

- **Paso 10:** Subir un archivo con el método PUT

The screenshot shows the Burp Suite Repeater interface. At the top, there are buttons for 'Send', 'Cancel', and navigation arrows. Below that is a section titled 'Request' with tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected, displaying the following request:

```

1 PUT /uploads/hello.txt HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.228.111.3/index.php
8 Connection: close
9 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0; Name=John
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12 Content-Length: 11
13
14 Hello World

```

Fig. 42: “PUT request con archivo a la página en Repeater de Burp Suite”,
Fuente: Elaboración Propia

Resultados:

The screenshot shows the Burp Suite Repeater interface with a 'Response' tab at the top. Below it are tabs for 'Raw', 'Headers', 'Hex', 'HTML', and 'Render'. The 'HTML' tab is selected, displaying the following response:

```

1 HTTP/1.1 201 Created
2 Date: Sat, 26 Aug 2023 20:50:42 GMT
3 Server: Apache
4 Location: http://192.228.111.3/uploads/hello.txt
5 Content-Length: 71
6 Connection: close
7 Content-Type: text/html; charset=ISO-8859-1
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
10<html><head>
11 <title>

```

Fig. 43: “Resultados de PUT request con archivo a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

El archivo se subió exitosamente. Para confirmar, vamos a checar los archivos en el directorio /uploads.

Request

```
Raw Params Headers Hex
1 GET /uploads/ HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.228.111.3/index.php
8 Connection: close
9 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0; Name=John
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13
```

Fig. 44: “GET request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

Response

```
Raw Headers Hex HTML Render
1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 20:52:45 GMT
3 Server: Apache
4 Vary: Accept-Encoding
5 Content-Length: 866
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
10 <html>
11   <head>
12     <title>Index of /uploads</title>
13   </head>
14   <body>
15     <h1>Index of /uploads</h1>
16     <table>
17       <tr><th valign="top"></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>
18       <tr><th colspan="5"><hr></th></tr>
19       <tr><td valign="top"></td><td><a href="/">Parent
Directory</a></td><td>&ampnbsp</td><td align="right"> - </td><td>&ampnbsp</td></tr>
20       <tr><td valign="top"></td><td><a href="hello.txt">
hello.txt</a></td><td align="right">2023-08-26 20:50 </td><td align="right"> 11 </td><td>
&ampnbsp</td></tr>
21       <tr><th colspan="5"><hr></th></tr>
22     </table>
23   </body></html>
```

Fig. 45: “Resultados de GET request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Sí se subió el archivo “hello.txt”. También podemos checar el contenido del archivo.

The screenshot shows the Burp Suite Repeater interface. At the top, there are buttons for 'Send', 'Cancel', and navigation arrows. Below that, a red 'Request' label is followed by tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected, displaying the following HTTP request:

```

1 GET /uploads/hello.txt HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.228.111.3/index.php
8 Connection: close
9 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0; Name=John
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13

```

Fig. 46: “GET request al nuevo archivo en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

The screenshot shows the Burp Suite Repeater interface with the target set to **http://192.228.111.3**. A red 'Response' label is followed by tabs for 'Raw', 'Headers', 'Hex', and 'Render'. The 'Raw' tab is selected, displaying the following HTTP response:

```

1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 20:58:19 GMT
3 Server: Apache
4 Last-Modified: Sat, 26 Aug 2023 20:50:42 GMT
5 ETag: "b-603d99d4d7b65"
6 Accept-Ranges: bytes
7 Content-Length: 11
8 Connection: close
9 Content-Type: text/plain
10
11 Hello World

```

Fig. 47: “Resultados de GET request al nuevo archivo en Repeater de Burp Suite”, Fuente: Elaboración Propia

- **Paso 11:** Eliminar el archivo.

Send Cancel < | > |

Request

Raw Params Headers Hex

```
1 DELETE /uploads/hello.txt HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.228.111.3/index.php
8 Connection: close
9 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0; Name=John
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13
```

Fig. 48: “DELETE request al nuevo archivo en Repeater de Burp Suite”,
Fuente: Elaboración Propia

Resultados:

Response

Raw Headers Hex Render

```
1 HTTP/1.1 204 No Content
2 Date: Sat, 26 Aug 2023 20:59:09 GMT
3 Server: Apache
4 Connection: close
5 Content-Type: text/plain
6
7
```

Fig. 49: “Resultados de DELETE request al nuevo archivo en Repeater de Burp Suite”, Fuente: Elaboración Propia

Checamos los archivos en el directorio de uploads.

Send Cancel < | > |

Request

- [Raw](#)
- [Params](#)
- [Headers](#)
- [Hex](#)

```

1 GET /uploads/ HTTP/1.1
2 Host: 192.228.111.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.228.111.3/index.php
8 Connection: close
9 Cookie: PHPSESSID=5np849161fpcpe580pduuvb1f0; Name=John
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13

```

Fig. 50: “GET request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Resultados:

Response

- [Raw](#)
- [Headers](#)
- [Hex](#)
- [HTML](#)
- [Render](#)

```

1 HTTP/1.1 200 OK
2 Date: Sat, 26 Aug 2023 21:00:25 GMT
3 Server: Apache
4 Vary: Accept-Encoding
5 Content-Length: 670
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
10 <html>
11   <head>
12     <title>Index of /uploads</title>
13   </head>
14   <body>
15     <h1>Index of /uploads</h1>
16     <table>
17       <tr><th valign="top"></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>
18       <tr><th colspan="5"><hr></th></tr>
19       <tr><td valign="top"></td><td><a href="/">Parent Directory</a></td><td>&ampnbsp</td><td align="right"> - </td><td>&ampnbsp</td></tr>
20       <tr><th colspan="5"><hr></th></tr>
21     </table>
22   </body></html>
23

```

Fig. 51: “Resultados de GET request a la página en Repeater de Burp Suite”, Fuente: Elaboración Propia

Conclusión:

Esta práctica nos mostró lo importante que es asegurarse que si se está trabajando en un sitio web, que todas sus páginas tengan claro cuáles son los métodos HTTP permitidos. Un atacante podría fácilmente ver qué métodos están permitidos y descubrir si el sitio tiene un método permitido que le haría capaz de hackear el sitio web. Para el desarrollo del laboratorio

Durante el laboratorio se aprendió a utilizar dos opciones para checar los métodos HTTP de una página web. Primero con curl directo de la terminal y luego con la aplicación de Burp Suite, se nos mostró que podíamos hacer. Se utilizó el OPTIONS request para ver los métodos HTTP, y se denota la importancia de no tener permitido métodos como PUT que le podría permitir a un atacante subir un archivo a la página web.

Bibliografía:

PentesterAcademy, “HTTP Method Enumeration”. Attack Defense.

<https://attackdefense.com/challengedetails?cid=1802>