

MÓDULO: CONFIGURACIÓN DE ENTORNOS Y LABORATORIOS

Actividad de aprendizaje:

Tipo de Ejercicio: Grupal

Para esta actividad los grupos deben investigar acerca de los métodos de comunicación usados por el protocolo HTTP y realizar una prueba de concepto de las principales funcionalidades de cada uno de estos, donde de forma práctica se demuestren sus características.

MÉTODOS DE HTTP

Los métodos del protocolo HTTP son las acciones que se pueden realizar sobre un recurso web. Se definen en la cabecera de la petición HTTP y se utilizan para indicar al servidor qué se quiere hacer con dicho recurso.

Los métodos HTTP más utilizados son los siguientes:

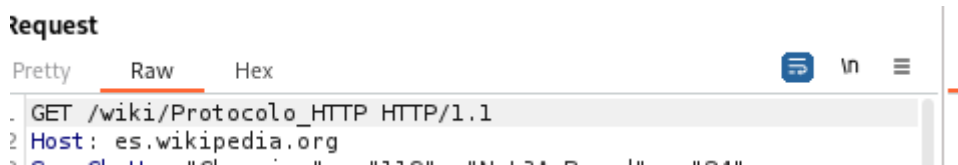
- GET: Solicita una representación de un recurso. Es el método más utilizado en la web, ya que se utiliza para cargar páginas web, imágenes, archivos, etc.

Para realizar una prueba de concepto del método GET, podemos utilizar un navegador web para cargar una página web. En la barra de direcciones del navegador, podemos escribir la URL de la página web que queremos cargar. Si la página web existe, el navegador enviará una petición GET al servidor web. El servidor web responderá con una representación de la página web.

Cargando la página web de Wikipedia sobre el protocolo HTTP.

Lo interceptaré con Burp Suite. El navegador enviará la siguiente petición GET al servidor web:

https://es.wikipedia.org/wiki/Protocolo_HTTP



El servidor web responderá con la siguiente representación de la página web:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3866
```

```
<html lang="es">
<head>
<title>Protocolo HTTP</title>
</head>
<body>
...
</body>

</html>
```

- POST: Crea un nuevo recurso o modifica un recurso existente.
Se utiliza para enviar datos al servidor, como por ejemplo para crear una cuenta de usuario, subir un archivo, etc.

Para realizar una prueba de concepto del método POST, podemos utilizar un formulario web para enviar datos al servidor. En el formulario, podemos incluir los datos que queremos enviar al servidor. Cuando se envía el formulario, el navegador enviará una petición POST al servidor web. El servidor web recibirá los datos enviados por el formulario y los procesará.

Podemos crear un formulario web para enviar nuestro nombre y apellidos al servidor:

```
<form action="/" method="post">  
  <input type="text" name="nombre" placeholder="Nombre">  
  <input type="text" name="apellidos" placeholder="Apellidos">  
  <button type="submit">Enviar</button>  
</form>
```

Cuando se envía este formulario, el navegador enviará la siguiente petición POST al servidor web:

```
POST / HTTP/1.1  
Host: localhost  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 35  
  
nombre=JOA&apellidos=NETO
```

El servidor web recibirá los datos enviados por el formulario y los procesará. Por ejemplo, el servidor web podría almacenar los datos en una base de datos.

- PUT: Actualiza un recurso existente. Es similar al método POST, pero se utiliza para reemplazar completamente el contenido del recurso.

Para realizar una prueba de concepto del método PUT, podemos utilizar un cliente HTTP para enviar datos a un recurso existente. Los datos enviados con el método PUT reemplazarán completamente el contenido del recurso.

Por ejemplo, podemos utilizar la herramienta curl para enviar un archivo a un servidor web:

```
curl -X PUT -T archivo.txt http://localhost/archivo.txt
```

En este caso, curl enviará la siguiente petición PUT al servidor web:

```
PUT /archivo.txt HTTP/1.1
Host: localhost
Content-Type: application/octet-stream
Content-Length: 12345
```

- DELETE: Elimina un recurso.

Para realizar una prueba de concepto del método DELETE, podemos utilizar un cliente HTTP para eliminar un recurso.

Usamos la herramienta curl para eliminar un archivo de un servidor web:

```
curl -X DELETE http://localhost/archivo.txt
```

En este caso, curl enviará la siguiente petición DELETE al servidor web:

```
DELETE /archivo.txt HTTP/1.1
```

```
Host: localhost
```

El servidor web recibirá la petición DELETE y eliminará el recurso correspondiente.