

PRACTICA 3.5 HACER PETICIONES HTML Y CAPTURAR CON WIRESHARK

ALEJANDRO ALMENDRAS NINA

1. WEBS DE USUARIO Y FORMULARIOS CON EL MÉTODO POST Y GET

Creamos un usuario sin permisos de administrador, **usuweb**.

```
administrador-204@ubuntu-204:~$ sudo adduser usuweb
[sudo] password for administrador-204:
Adding user `usuweb' ...
Adding new group `usuweb' (1001) ...
Adding new user `usuweb' (1001) with group `usuweb' ...
Creating home directory `/home/usuweb' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuweb
Enter the new value, or press ENTER for the default
  Full Name []: usuweb
    Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n]
```

Dentro de la carpeta home del usuario creamos una carpeta llamada 'public_html'.

```
usuweb@ubuntu-204:~$ mkdir public_html
```

Le damos los permisos 755 y que pertenezca al grupo www-data.

```
administrador-204@ubuntu-204:/home/usuweb$ sudo chmod 755 public_html
administrador-204@ubuntu-204:/home/usuweb$ sudo chown :www-data public_html
administrador-204@ubuntu-204:/home/usuweb$
```

Dentro de esa carpeta creamos un documento **usuweb.html** y un directorio **listar** que contiene los archivos listar1.html, listar2.html, ejem_get.php y ejem_post.php

```
usuweb@ubuntu-204:~$ cd public_html/
usuweb@ubuntu-204:~/public_html$ touch usuweb.html
usuweb@ubuntu-204:~/public_html$ mkdir listar
usuweb@ubuntu-204:~/public_html$ touch ./listar/listar1.html
usuweb@ubuntu-204:~/public_html$ touch ./listar/listar2.html
usuweb@ubuntu-204:~/public_html$ touch ./listar/ejem_get.php
usuweb@ubuntu-204:~/public_html$ touch ./listar/ejem_post.php
usuweb@ubuntu-204:~/public_html$ tree .
.
├── listar
│   ├── ejem_get.php
│   ├── ejem_post.php
│   ├── listar1.html
│   └── listar2.html
└── usuweb.html

1 directory, 5 files
```

Habilitamos los módulos: userdir y autoindex

```
administrador-204@ubuntu-204:~$ sudo a2enmod userdir
[sudo] password for administrador-204:
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
administrador-204@ubuntu-204:~$ sudo a2enmod autoindex
Module autoindex already enabled
administrador-204@ubuntu-204:~$ sudo systemctl restart apache2
administrador-204@ubuntu-204:~$
```

Dentro de **public_html** creamos un archivo .htaccess de configuración para la página.

```
GNU nano 4.8 public_html/.htaccess
DirectoryIndex usuweb.html
Require all granted
```

Reiniciamos el servicio de apache.

```
administrador-204@ubuntu-204:~$ sudo systemctl restart apache2
administrador-204@ubuntu-204:~$ sudo systemctl status apache2
```

Veamos ahora el código de cada una de las páginas.

```
GNU nano 4.8 public_html/usuweb.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>USUWEB</title><style>body{font-family:sans-serif}</style></head>
<body><h2>Para ir a listar1 pulse <a href="./listar/listar1.html">aquí</a></h2>
<h2>Para ir a listar2 pulse <a href="./listar/listar2.html">aquí</a></h2></body></html>
```

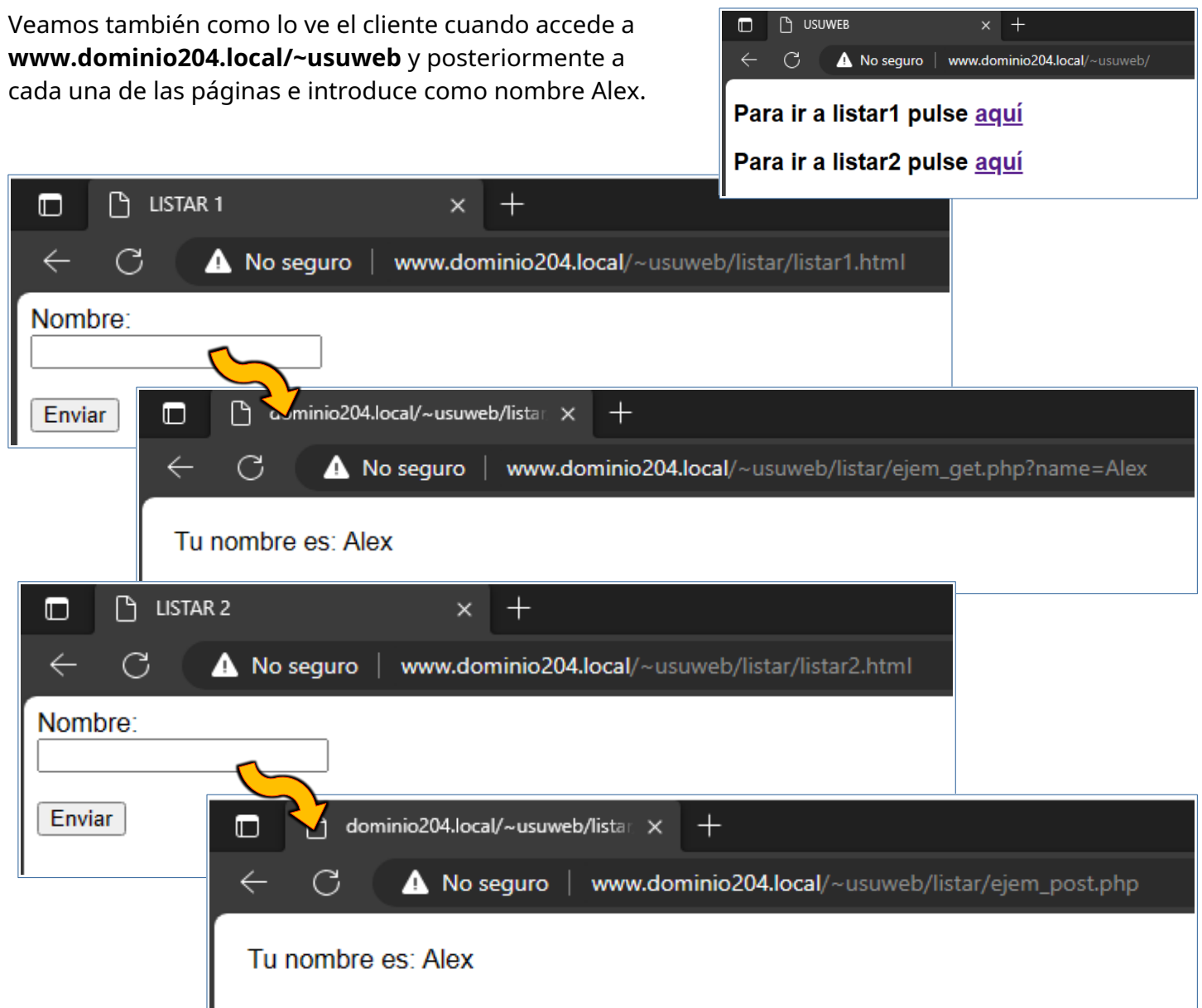
```
GNU nano 4.8 public_html/listar/listar1.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>LISTAR 1</title><style>body{font-family:sans-serif}</style></head>
<body>
<form action="./ejem_get.php" method="get">
  <label>Nombre: </label><br>
  <input type="text" id="name" name="name"><br><br>
  <input type="submit" value="Enviar">
</form>
</body></html>
```

```
GNU nano 4.8 public_html/listar/listar2.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>LISTAR 2</title><style>body{font-family:sans-serif}</style></head>
<body>
<form action="./ejem_post.php" method="post">
  <label>Nombre: </label><br>
  <input type="text" id="name" name="name"><br><br>
  <input type="submit" value="Enviar">
</form>
</body></html>
```

```
GNU nano 4.8 public_html/listar/ejem_get.php
<?php
echo "<style>body{font-family:sans-serif;margin:20px;}</style>";
echo "Tu nombre es: " . htmlspecialchars($_GET["name"]);
?>
```

```
GNU nano 4.8 public_html/listar/ejem_post.php
<?php
echo "<style>body{font-family:sans-serif;margin:20px;}</style>";
echo "Tu nombre es: " . htmlspecialchars($_POST["name"]);
?>
```

Veamos también como lo ve el cliente cuando accede a **www.dominio204.local/~usuweb** y posteriormente a cada una de las páginas e introduce como nombre Alex.



2. PÁGINAS DE ERROR PERSONALIZADAS

Para definir las páginas de errores, en **/etc/apache2/conf-available/localized-error-pages.conf** y lo configuramos como en la imagen.

```
GNU nano 4.8 /etc/apache2/conf-available/localized-error-pages.conf
<IfModule mod_negotiation.c>
  <IfModule mod_include.c>
    <IfModule mod_alias.c>

      Alias /error/ "/usr/share/apache2/error/"

      <Directory "/usr/share/apache2/error">
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en cs de es fr it nl sv pt-br ro
        ForceLanguagePriority Prefer Fallback
      </Directory>
# ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
# ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/error403.html
ErrorDocument 404 /error/error404.html
# ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
# ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
# ErrorDocument 410 /error/HTTP_GONE.html.var
```

Recordemos además activar el módulo include ejecutando **sudo a2enmod include**.

El contenido de esos documentos de error y su ubicación es el siguiente:

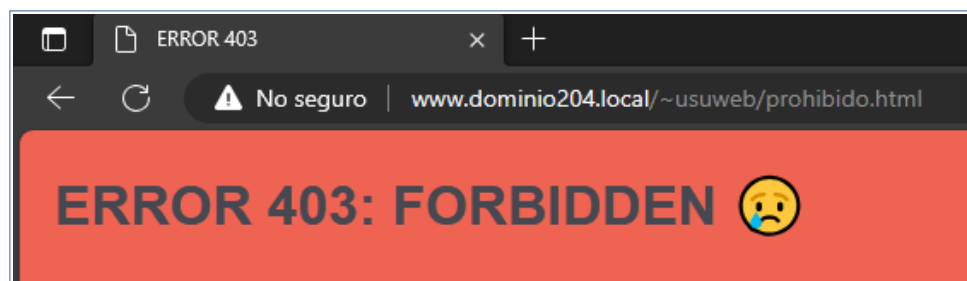
```
GNU nano 4.8 /usr/share/apache2/error/error403.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>ERROR 403</title>
<style>body {color: #454851; font-family:sans-serif; background-color:#EE6352; margin:20px;}</style>
</head><body><h1>ERROR 403: FORBIDDEN &#128546</h1></body></html>
```

```
GNU nano 4.8 /usr/share/apache2/error/error404.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>ERROR 404</title>
<style>body {color: #454851; font-family:sans-serif; background-color:#EE6352; margin:20px;}</style>
</head><body><h1>ERROR 404: Página no encontrada &#128546</h1></body></html>
```

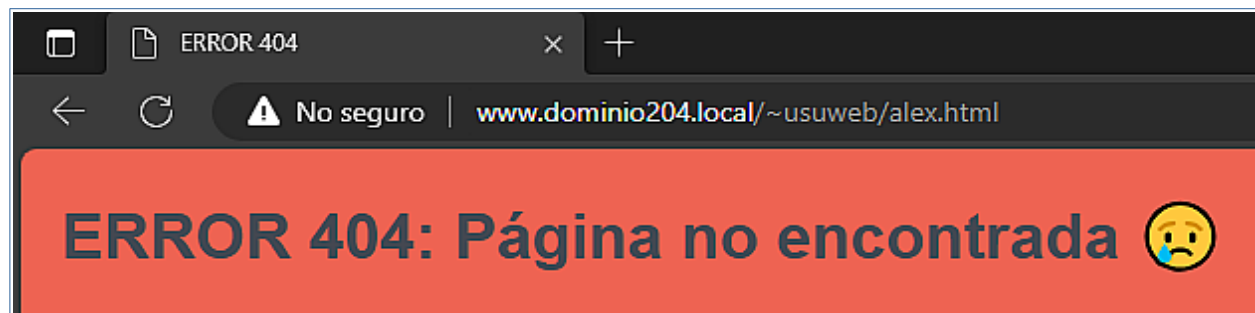
Para comprobar el error 403, bastará con crear un archivo sin permisos para otros.

```
usuweb@ubuntu-204:~/public_html$ ls -l prohibido.html
-rwxr-x--- 1 usuweb usuweb 19 nov 28 09:57 prohibido.html
```

El cual se visualizará desde el cliente de la siguiente manera:



Y para comprobar el 404, intentaremos acceder a un archivo que no existe.



3. WIRESHARK

3.1) Visualiza la información de la cabeceras de los siguientes servidores webs:

<http://www.dominio204.local>

Time	Source	Destination	Protocol	Length	Info
6 0.012497	172.16.204.151	172.16.204.202	HTTP	480	GET / HTTP/1.1
13 0.051201	172.16.204.202	172.16.204.151	HTTP	611	HTTP/1.1 200 OK (text/html)

Tanto para éste caso como para los siguientes de éste apartado vemos que las páginas se solicitan por el cliente y son servidas por el servidor exitosamente, viendo en el primer caso que pide la raíz del dominio, en el segundo caso pide el subdirectorio del usuario usuweb, siguiendo la URI que se detalla en el bloque desplegado. De la misma manera en la tercera imagen vemos la ruta completa de la página solicitada, y que se sirve correctamente ya que todas estas conexiones no están cifradas.

http://www.dominio204.local/~usuweb

6	1.777412	172.16.204.151	172.16.204.202	HTTP	488	GET /~usuweb/ HTTP/1.1
8	1.778494	172.16.204.202	172.16.204.151	HTTP	623	HTTP/1.1 200 OK (text/html)

Frame 6: 488 bytes on wire (3904 bits), 488 bytes captured (3904 bits) on interface \Device\NPF_{CE37F9D2-B0D4-4642-84DE-A14D496317A5}, Ethernet II, Src: PCSSystemtec_76:75:44 (08:00:27:76:75:44), Dst: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d)
Internet Protocol Version 4, Src: 172.16.204.151, Dst: 172.16.204.202
Transmission Control Protocol, Src Port: 50016, Dst Port: 80, Seq: 1, Ack: 1, Len: 434

Hypertext Transfer Protocol

> GET /~usuweb/ HTTP/1.1\r\n
Host: www.dominio204.local\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\nAccept-Encoding: gzip, deflate\r\nAccept-Language: es\r\n\r\n
[Full request URI: http://www.dominio204.local/~usuweb/]
[HTTP request 1/1]
[Response in frame: 8]

http://www.dominio204.local/~usuweb/listar/listar1.html

No.	Time	Source	Destination	Protocol	Length	Info
4	0.017069	172.16.204.151	172.16.204.202	HTTP	554	GET /~usuweb/listar/listar1.html HTTP/1.1
6	0.018080	172.16.204.202	172.16.204.151	HTTP	658	HTTP/1.1 200 OK (text/html)

3.2) Obtén el contenido de las páginas y comprueba el método utilizado, así como los parámetros y contenidos de los mensajes :

http://www.dominio204.local/~usuweb/listar/listar1.html

No.	Time	Source	Destination	Protocol	Length	Info
7	0.736205	172.16.204.151	172.16.204.202	HTTP	554	GET /~usuweb/listar/listar1.html HTTP/1.1
9	0.737163	172.16.204.202	172.16.204.151	HTTP	658	HTTP/1.1 200 OK (text/html)

> Frame 7: 554 bytes on wire (4432 bits), 554 bytes captured (4432 bits) on interface \Device\NPF_{CE37F9D2-B0D4-4642-84DE-A14D496317A5}, Ethernet II, Src: PCSSystemtec_76:75:44 (08:00:27:76:75:44), Dst: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d)
> Internet Protocol Version 4, Src: 172.16.204.151, Dst: 172.16.204.202
> Transmission Control Protocol, Src Port: 50195, Dst Port: 80, Seq: 1, Ack: 1, Len: 500

Hypertext Transfer Protocol

> GET /~usuweb/listar/listar1.html HTTP/1.1\r\n
Host: www.dominio204.local\r\n

http://www.dominio204.local:83 (web con autenticación)

10	1.364037	172.16.204.151	172.16.204.202	HTTP	483	GET / HTTP/1.1
12	1.364890	172.16.204.202	172.16.204.151	HTTP	801	HTTP/1.1 401 Unauthorized (text/html)
23	6.869780	172.16.204.151	172.16.204.202	HTTP	552	GET / HTTP/1.1
25	6.893202	172.16.204.202	172.16.204.151	HTTP	658	HTTP/1.1 200 OK (text/html)

Frame 23: 552 bytes on wire (4416 bits), 552 bytes captured (4416 bits) on interface \Device\NPF_{CE37F9D2-B0D4-4642-84DE-A14D496317A5}, Ethernet II, Src: PCSSystemtec_76:75:44 (08:00:27:76:75:44), Dst: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d)
Internet Protocol Version 4, Src: 172.16.204.151, Dst: 172.16.204.202
Transmission Control Protocol, Src Port: 50140, Dst Port: 82, Seq: 1, Ack: 1, Len: 498

Hypertext Transfer Protocol

> GET / HTTP/1.1\r\n
Host: www.dominio204.local:82\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Authorization: Basic ZW1wbGVhZG8yOmFiYW==\r\n
Credentials: empleado2:abc
Upgrade-Insecure-Requests: 1\r\n

Vemos que en ambos casos los métodos HTTP usados son de tipo GET porque están solicitando las páginas, en el primer caso enviando solamente como parámetro la ruta de la página solicitada.

Mientras que en el segundo caso las credenciales también son enviadas como parámetros y visibles para nosotros al no ser una conexión segura-cifrada.

3.3) Comprueba como se manda información al servidor mediante el método GET en las URL:

<http://www.dominio204.local/~usuweb/listar/listar1.html?nombre=Ana>

6	0.011736	172.16.204.151	172.16.204.202	HTTP	582 GET /~usuweb/listar/ejem_get.php?name=Ana HTTP/1.1
9	0.013328	172.16.204.202	172.16.204.151	HTTP	394 HTTP/1.1 200 OK (text/html)

Internet Protocol Version 4, Src: 172.16.204.151, Dst: 172.16.204.202
Transmission Control Protocol, Src Port: 50153, Dst Port: 80, Seq: 1, Ack: 1, Len: 528
Hypertext Transfer Protocol

- GET /~usuweb/listar/ejem_get.php?name=Ana HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): GET /~usuweb/listar/ejem_get.php?name=Ana HTTP/1.1\r\n]Request Method: GET
 - Request URI: /~usuweb/listar/ejem_get.php?name=Ana
 - Request URI Path: /~usuweb/listar/ejem_get.php
 - Request URI Query: name=Ana
 - Request URI Query Parameter: name=Ana
 - Request Version: HTTP/1.1

Vemos que se usa el método GET y que tanto en la cabecera como dentro de consulta (query URI) se pueden observar la variable enviada y el valor.

<https://aulavirtual.murciaeduca.es/course/view.php?id=95019>

No.	Time	Source	Destination	Protocol	Length	Info
14	0.040376	172.16.204.151	147.84.223.12	TLSv1.3	600	Client Hello (SNI=aulavirtual.murciaeduca.es)
15	0.041924	147.84.223.12	172.16.204.151	TCP	60	443 → 50156 [ACK] Seq=1 Ack=547 Win=65535 Len=0
16	0.046938	172.16.204.202	172.16.204.151	DNS	290	Standard query response 0x6428 HTTPS nav-edge.smartscreen.microsoft.com
17	0.047486	172.16.204.202	172.16.204.151	DNS	229	Standard query response 0x5493 A nav-edge.smartscreen.microsoft.com CNAME
18	0.048004	172.16.204.151	20.31.251.109	TCP	66	50157 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
19	0.063194	147.84.223.12	172.16.204.151	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
20	0.063194	147.84.223.12	172.16.204.151	TCP	1514	443 → 50156 [ACK] Seq=1461 Ack=547 Win=65535 Len=1460 [TCP segment of a
21	0.063256	172.16.204.151	147.84.223.12	TCP	54	50156 → 443 [ACK] Seq=547 Ack=2921 Win=64240 Len=0
22	0.064123	147.84.223.12	172.16.204.151	TLSv1.3	1344	Application Data, Application Data, Application Data
23	0.065294	172.16.204.151	147.84.223.12	TLSv1.3	134	Change Cipher Spec, Application Data
24	0.065736	172.16.204.151	147.84.223.12	TLSv1.3	762	Application Data
25	0.066168	147.84.223.12	172.16.204.151	TCP	60	443 → 50156 [ACK] Seq=4211 Ack=627 Win=65535 Len=0
26	0.066168	147.84.223.12	172.16.204.151	TCP	60	443 → 50156 [ACK] Seq=4211 Ack=1335 Win=65535 Len=0
27	0.081532	147.84.223.12	172.16.204.151	TLSv1.3	660	Application Data, Application Data
28	0.098440	20.31.251.109	172.16.204.151	TCP	60	443 → 50157 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460

> Frame 24: 762 bytes on wire (6096 bits), 762 bytes captured (6096 bits) on interface \Device\NPF_{CE37F9D2-B0D4-4642-84DE-A14D496317A5},
> Ethernet II, Src: PCSSystemtec_76:75:44 (08:00:27:76:75:44), Dst: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d)
> Internet Protocol Version 4, Src: 172.16.204.151, Dst: 147.84.223.12
> Transmission Control Protocol, Src Port: 50156, Dst Port: 443, Seq: 627, Ack: 4211, Len: 708
> Transport Layer Security
> TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

A diferencia del caso anterior la conexión está cifrada y sin detallar demasiado vemos que se producen paquetes TLS sobre TCP, que elaboran un intercambio primero con mensajes de handshake, consenso del cifrado, y finalmente se solicita y sirve la página todo ello sin ser posible la visualización de la página o parámetros enviados como las credenciales usadas.

3.4) Usando el comando POST (que envía el contenido en el cuerpo) manda tu nombre a la página:

http://www.dominio204.local/~usuweb/listar/ejem_post.php

6	0.014109	172.16.204.151	172.16.204.202	HTTP	715 POST /~usuweb/listar/ejem_post.php HTTP/1.1 (application/x-www-form-urlencoded)
8	0.015670	172.16.204.202	172.16.204.151	HTTP	395 HTTP/1.1 200 OK (text/html)

> Frame 6: 715 bytes on wire (5720 bits), 715 bytes captured (5720 bits) on interface \Device\NPF_{CE37F9D2-B0D4-4642-84DE-A14D496317A5}
> Ethernet II, Src: PCSSystemtec_76:75:44 (08:00:27:76:75:44), Dst: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d)
> Internet Protocol Version 4, Src: 172.16.204.151, Dst: 172.16.204.202
> Transmission Control Protocol, Src Port: 50035, Dst Port: 80, Seq: 1, Ack: 1, Len: 661
> Hypertext Transfer Protocol

- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "name" = "Alex"
 - Key: name
 - Value: Alex

Vemos que a diferencia del 3.3 con GET, aquí el método es POST y la variable y su valor no se pasan en la cabecera ni en Query URI, sino se pasa en el cuerpo, en el apartado que vemos en la imagen detallando que se trata de un apartado de formulario con su clave y valor.

3.5) Captura el mensaje de error 404 que diseñaste en el apartado 2 de esta práctica realizando una búsqueda de una página que no exista en tu servidor.

No.	Time	Source	Destination	Protocol	Length	Info
6	0.021582	172.16.204.151	172.16.204.202	HTTP	497	GET /~usuweb/alex.html HTTP/1.1
9	0.022527	172.16.204.202	172.16.204.151	HTTP	574	HTTP/1.1 404 Not Found (text/html)

> Frame 9: 574 bytes on wire (4592 bits), 574 bytes captured (4592 bits) on interface \Device\NPF_{CE37F9D2-B...
 > Ethernet II, Src: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d), Dst: PCSSystemtec_76:75:44 (08:00:27:76:75:44)
 > Internet Protocol Version 4, Src: 172.16.204.202, Dst: 172.16.204.151
 > Transmission Control Protocol, Src Port: 80, Dst Port: 50206, Seq: 1, Ack: 444, Len: 520
 > Hypertext Transfer Protocol
 > HTTP/1.1 404 Not Found\r\n

En la imagen anterior y la siguiente vemos que se solicitan páginas pero ambas no se sirven correctamente una por inexistente y otro por falta de permisos, en los mensajes por parte del servidor se detalla el estado de la petición, es decir los códigos de error, y se devuelven con ello las páginas que hemos definido con anterioridad.

No.	Time	Source	Destination	Protocol	Length	Info
13	0.149222	172.16.204.151	172.16.204.202	HTTP	502	GET /~usuweb/prohibido.html HTTP/1.1
18	0.150432	172.16.204.202	172.16.204.151	HTTP	562	HTTP/1.1 403 Forbidden (text/html)

Frame 18: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits) on interface \Device\NPF_{CE37F9...}
 Ethernet II, Src: PCSSystemtec_c1:96:0d (08:00:27:c1:96:0d), Dst: PCSSystemtec_76:75:44 (08:00:27:76:75:44)
 Internet Protocol Version 4, Src: 172.16.204.202, Dst: 172.16.204.151
 Transmission Control Protocol, Src Port: 80, Dst Port: 50210, Seq: 1, Ack: 449, Len: 508
 Hypertext Transfer Protocol
 > HTTP/1.1 403 Forbidden\r\n
 Date: Tue, 28 Nov 2023 11:25:12 GMT\r\n