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
```

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
```

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
```

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>
```

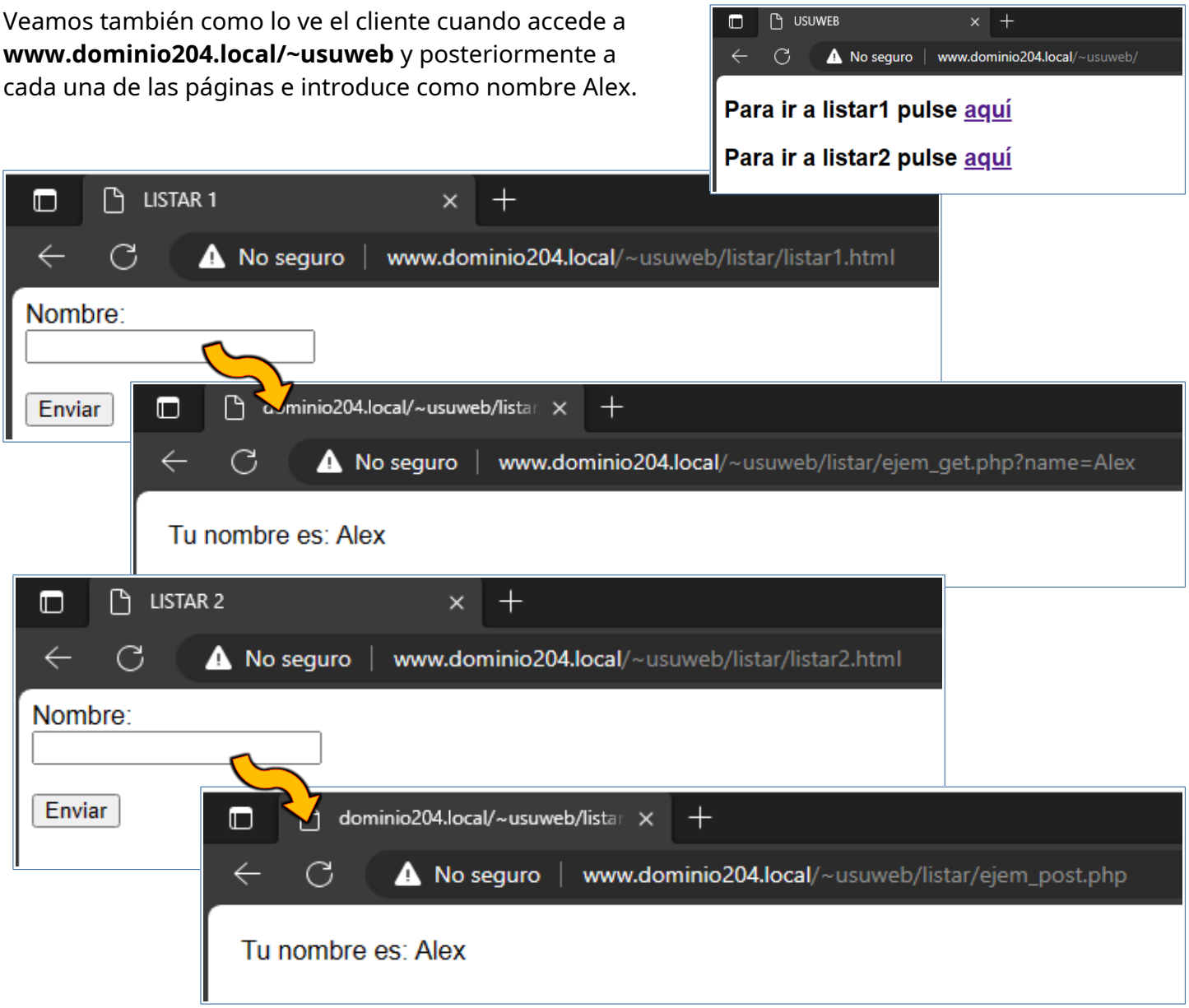
```
  GNU nano 4.8                                  public_html/listar/listar1.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>LISTAR 1</title><style>body{font-family:sans-serif}</style></head>
<body>
<form action="./ejem_get.php" method="get">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

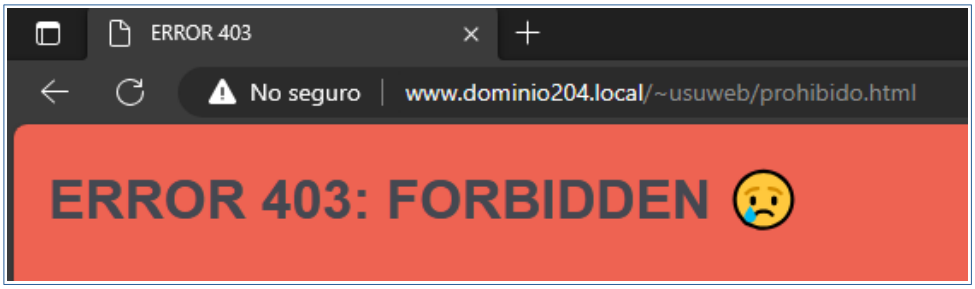
```
 GNU nano 4.8                    /usr/share/apache2/error/error403.html
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>ERROR 403</title>
<style>body {color: #454851; font-family:sans-serif; background-color:#EE6352; margin:20px;}</style>
</head>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

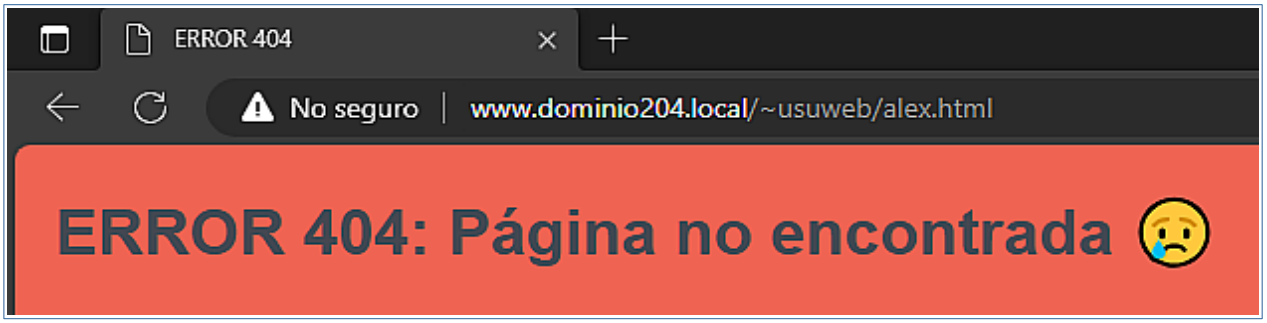
Para comprobar el error 403, bastará con crear un archivo sin permisos para otros.

```
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

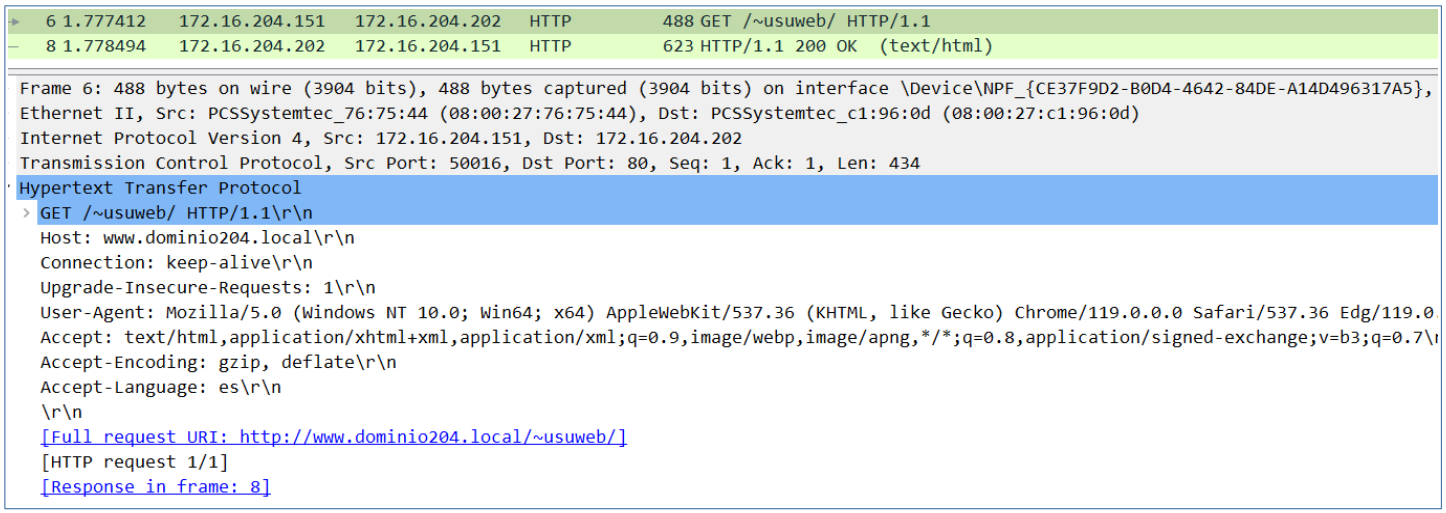
3.1) Visualiza la información de la cabeceras de los siguientes servidores webs:

http://www.dominio204.local

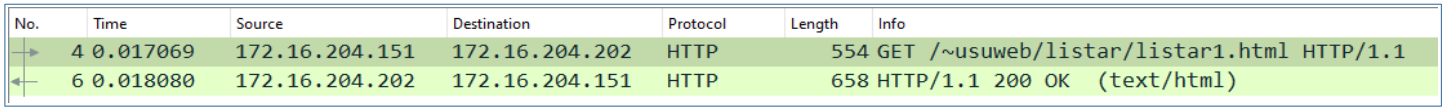
	Time	Source	Destination	Protocol	Length	Info
6	0.012497	172.16.204.151	172.16.204.202	HTTP	480	GET / HTTP/1.1
13	0.051201	172.16.204.202	172.16.204.151	HTTP	611	HTTP/1.1 200 OK (text/html)

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

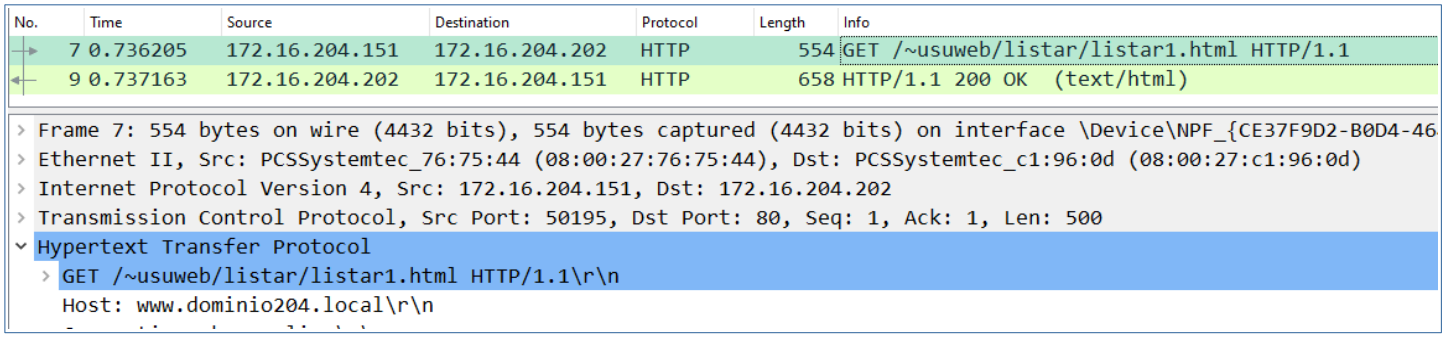


http://www.dominio204.local/~usuweb/listar/listar1.html

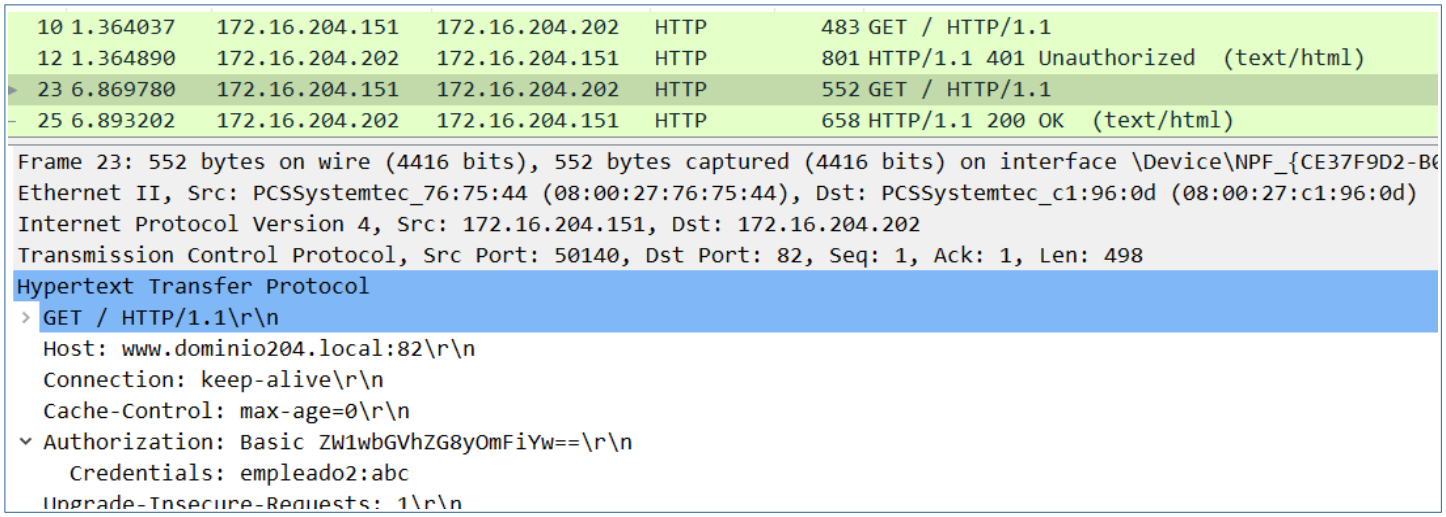


3.2) Obtén el contenido de las páginas y comprueba el método utilizado, así como los parámetros y contenidos de los mensajes :

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



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

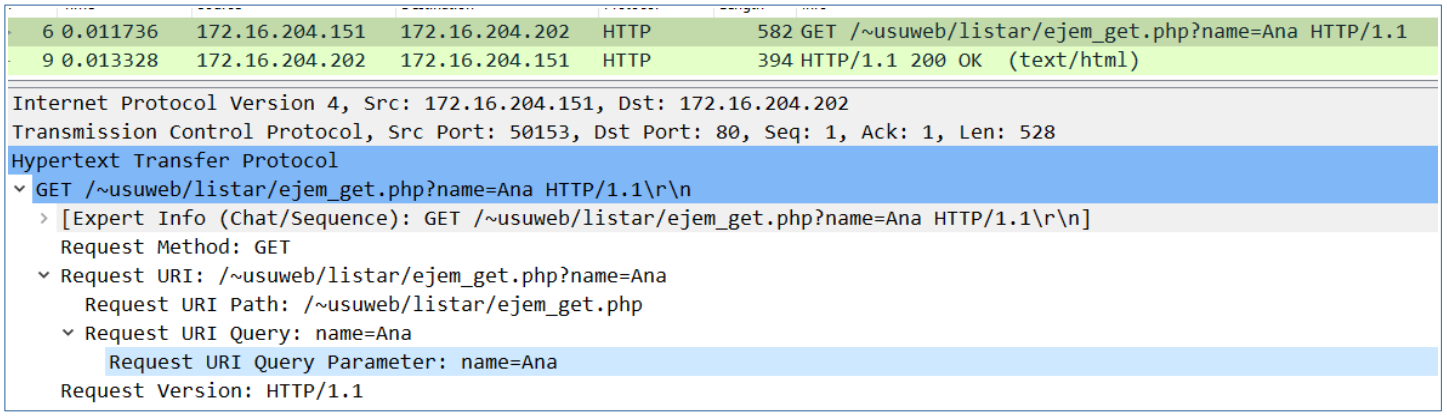


Vemos que en ambos casos los métodos HTTP usados son de tipo GET porque están solicitando las páginas, en el primer caso enviando solamente como parámetro la ruta de la página solicitada.

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

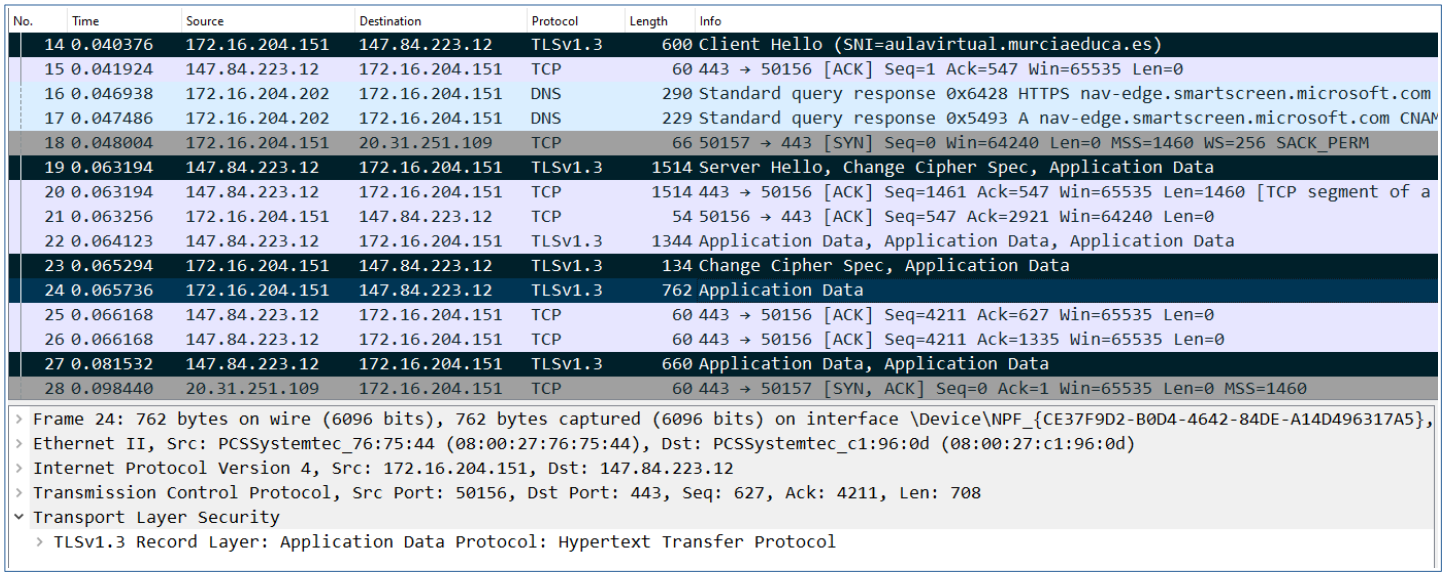
3.3) Comprueba como se manda información al servidor mediante el método GET en las URL:

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



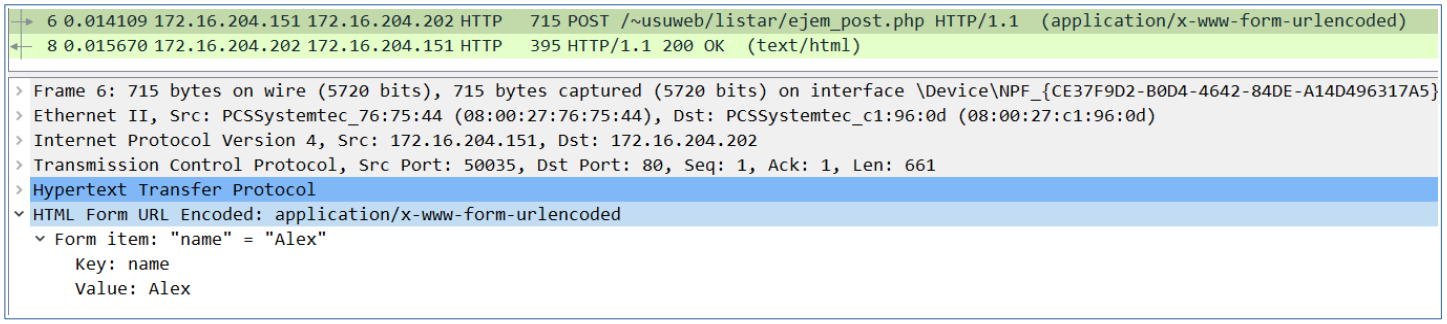
Vemos que se usa el método GET y que tanto en la cabecera como dentro de consulta (query URI) se pueden observar la variable enviada y el valor.

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



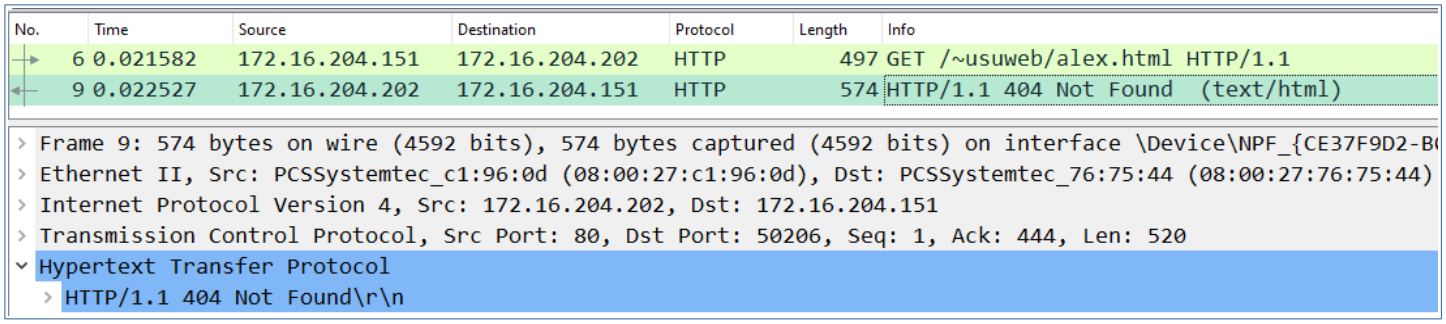
A diferencia del caso anterior la conexión está cifrada y sin detallar demasiado vemos que se producen paquetes TLS sobre TCP, que elaboran un intercambio primero con mensajes de handshake, consenso del cifrado, y finalmente se solicita y sirve la página todo ello sin ser posible la visualización de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



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.



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.