

Redes de Telecomunicaciones

Práctica 3: Implementación de un servidor VoD y análisis de protocolos con Wireshark

Franklin Mauricio Gómez López, Juan Diego Tenesaca

Universidad de Cuenca, Facultad de Ingeniería,
Av. 12 de Abril y Agustín Cueva, EC010112, Cuenca, ECU
franklin.gomez@ucuenca.edu.ec, juan.tenesaca@ucuenca.edu.ec

Resumen En el presente informe se expondrá el procedimiento empleado para llevar a cabo la implementación de un servidor de Video on Demand (VoD) utilizando la herramienta Ampache. Este servidor se ejecuta sobre un servidor web Apache2 y emplea una base de datos relacional del tipo MySQL/MariaDB para la gestión de los datos. Se incluirá también un análisis de los protocolos asociados al servidor mediante la utilización de la herramienta Wireshark.

Keywords: VoD · Apache · Ampache · Ubuntu · Maradb

1. Introducción

Este informe está dividido en varias secciones. En la primera sección se encuentra la Introducción, que proporciona una visión general del tema a tratar. La segunda sección, es la del Marco Teórico, en donde se exponen los conceptos y se describe los programas que se usaran para levantar el servicio de VoD, os empleados en el informe. En la tercera sección, se encuentra el Desarrollo, sección en la que se presenta los pasos a seguir para la configuración del servidor con sus respectivas pruebas y los resultados obtenidos en la implementación usando el software de Wireshark para el análisis de paquetes que se generan en la transmisión, las conclusiones obtenidas se presentan en la cuarta sección y finalmente en la quinta sección se en listan las bibliografías en las cuales se realizaron algunas consultas teóricas para la elaboración de esta practica.

2. Marco Teórico

2.1. Apache

Apache es un servidor web gratuito y de código abierto que se usa comúnmente para servir páginas web en Internet. Se ejecuta en una variedad de sistemas operativos, incluido Ubuntu.

- Multiplataforma: Apache se ejecuta en una variedad de sistemas operativos, incluidos Ubuntu, Windows, macOS y más.
- Admite múltiples protocolos: HTTP, HTTPS, FTP, etc

- **Arquitectura modular:** los módulos se pueden agregar o quitar según sea necesario. Los módulos pueden proporcionar funciones como autenticación, compresión y gestión de sesiones.
- **Alto rendimiento:** Puede manejar una gran cantidad de consultas simultáneamente y tiene configuraciones flexibles para adaptarse a diferentes necesidades.
- **Configuración flexible:** Los administradores del sistema pueden personalizar la configuración para satisfacer las necesidades específicas de sus sitios.
- **Seguridad:** Incluye características de seguridad como SSL/TLS, autenticación de usuarios y restricciones de acceso. [1]

2.2. Ampache

Ampache es una aplicación web de código abierto que permite a los usuarios acceder y reproducir bibliotecas de música en línea. Es un sistema cliente-servidor, lo que significa que los usuarios pueden acceder a su biblioteca de música desde cualquier dispositivo con acceso a Internet.

- **Compatibilidad con múltiples formatos :** Ampache admite múltiples formatos de música y video, incluidos MP3, FLAC, OGG, AAC, mp4 y mas .
- **Compatibilidad con listas de reproducción:** los usuarios pueden crear listas de reproducción y guardarlas en su cuenta de Ampache para su uso posterior
- **Integración con otros servicios:** se puede integrar con otros servicios como Last.fm y Libre.fm para personalizar las recomendaciones musicales.
- **Gestión de usuarios y permisos:** los administradores del sistema pueden establecer permisos de usuario y restringir el acceso a ciertas funciones o áreas de la aplicación.
- **Personalización de la interfaz de usuario:** los usuarios pueden personalizar el aspecto de la interfaz de usuario según sus preferencias.
- **Funciones de transmisión:** admite la transmisión de música en vivo, lo que significa que los usuarios pueden escuchar música sin descargarla primero.
- **API:** Ampache proporciona una API (interfaz de programación de aplicaciones) que permite a los desarrolladores crear aplicaciones que interactúan con la plataforma. [2]

2.3. MySQL:

Es sistema de gestión de bases de datos relacionales de código abierto que puede manejar grandes cantidades de datos y usuarios concurrentes gracias a su arquitectura cliente-servidor. MySQL tiene características técnicas sólidas, como seguridad, escalabilidad, soporte multiplataforma, lenguaje de consulta SQL, replicación y varios motores de almacenamiento. MySQL admite múltiples lenguajes de programación y puede ejecutarse en una variedad de sistemas operativos. La replicación de datos es una de las características más destacadas de MySQL, lo que significa que los datos se pueden replicar en varios servidores para mejorar la disponibilidad y la escalabilidad. En resumen, MySQL es una base de datos robusta y escalable ampliamente utilizada en aplicaciones web y empresariales debido a sus potentes características técnicas y su capacidad para manejar grandes cantidades de datos y usuarios concurrentes. [3]

2.4. MariaDB:

Es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que es una bifurcación de MySQL. MariaDB fue creado por los desarrolladores originales de MySQL en respuesta a las preocupaciones de que Oracle Corporation estaba adquiriendo MySQL. Las características técnicas de MariaDB incluyen escalabilidad, seguridad, compatibilidad con MySQL, replicación, soporte multiplataforma y optimización del rendimiento.

La escalabilidad de MariaDB se logra a través de su capacidad para manejar grandes cantidades de datos y usuarios simultáneos. Al igual que MySQL, MariaDB utiliza una arquitectura cliente-servidor para conectarse y administrar datos. Cuando se trata de seguridad, MariaDB tiene características poderosas que incluyen autenticación de dos factores, cifrado de datos, control de acceso y autenticación. También incluye funciones de seguridad avanzadas, como el cifrado de datos en reposo y la autenticación de contraseña avanzada. MariaDB es compatible con MySQL y puede ejecutar aplicaciones y scripts originalmente escritos en MySQL, las aplicaciones que actualmente se ejecutan en MySQL se pueden migrar a MariaDB con muy pocos cambios.

La replicación de datos en MariaDB es similar a MySQL y proporciona respaldo y recuperación ante desastres. Además, MariaDB tiene varias funciones de optimización del rendimiento, como la optimización de consultas y la optimización del motor de almacenamiento. MariaDB también es multiplataforma y puede ejecutarse en diferentes sistemas operativos. Además, ofrece muchas mejoras de rendimiento a MySQL, incluido el uso de múltiples subprocesos de servidor para manejar consultas y conexiones de clientes. [4]

3. Desarrollo

Para el servicio de vídeo bajo demanda se usa un servidor VoD de Apache, se trabaja sobre una maquina con sistema operativo Ubuntu 20.04. Para este servicio se debe instalar el servidor web con una base de datos que en este caso sera *MySQL* MariaDb y otras librerías necesarias para el entorno.

```
apt-get install apache2 libapache2-mod-php php php-cli mariadb-server
php mysql php-curl php-json php-gd php-xml unzip curl git zip ffmpeg -y
```

```

root@jd:/home/jd# apt-get install apache2 libapache2-mod-php php php-cli mariadb-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
zip ya está en su versión más reciente (3.0-11build1).
fijado zip como instalado manualmente.
unzip ya está en su versión más reciente (6.0-25ubuntu1.1).
fijado unzip como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no son neces
gir1.2-goa-1.0 gstreamer1.0-vaapi libfwupdplugin1
libgstreamer-plugins-bad1.0-0 libva-wayland2 libxmlb1
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
apache2-bin apache2-data apache2-utils galera-3 gawk git-man libaio1
libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
libaprutil1-ldap libavdevice58 libavresample4 libbcgi-fast-perl
libbcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdc1394-22
liberror-perl libfcgi-perl libhtml-template-perl liblua5.2-0 libopenal-data
libopenal1 libreadline5 libstdc++2.0-0 libsigsegv2 libsndio7.0
libterm-readkey-perl mariadb-client-10.3 mariadb-client-core-10.3
mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 php-common
php7.4 php7.4-cli php7.4-common php7.4-curl php7.4-gd php7.4-imap

```

Figura 1: Descarga e instalación de archivos necesarios

En el archivo `/etc/php/7.4/apache2/php.ini` se deben editar los siguiente parámetros:

```

upload_max_filesize = 100M
post_max_size = 100M
date.timezone = America

```

```

;upload_tmp_dir =
;
; Maximum allowed size for uploaded files
; http://php.net/upload-max-filesize
upload_max_filesize = 2M
; Maximum number of files that can be upl
max_file_uploads = 20

```

Figura 2: Edición de los parámetros `upload_max_filesize` y `post_max_size` del archivo `php.ini`

```
; Defines the default timezone used by  
; http://php.net/date.timezone  
date.timezone = America  
; http://php.net/date.default-latitude  
;date.default_latitude = 31.7667
```

Figura 3: Edición del parámetro date.timezone del archivo php.ini

Se procede a reiniciar los servicios de Apache con el comando:

```
sudo systemctl restart apache2    // reiniciar servicio  
sudo systemctl status apache2     // estado del servicio
```

```
root@jd:/etc/php/7.4/apache2# sudo systemctl restart apache2  
root@jd:/etc/php/7.4/apache2# sudo systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2023-04-16 17:00:24 -05; 5s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Process: 38255 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
 Main PID: 38259 (apache2)  
    Tasks: 6 (limit: 18687)  
   Memory: 12.2M  
    CGroup: /system.slice/apache2.service  
           └─38259 /usr/sbin/apache2 -k start  
           └─38260 /usr/sbin/apache2 -k start  
           └─38261 /usr/sbin/apache2 -k start  
           └─38262 /usr/sbin/apache2 -k start  
           └─38263 /usr/sbin/apache2 -k start  
           └─38264 /usr/sbin/apache2 -k start
```

Figura 4: Estado del servidor de Apache

Ahora se procede a configurar la base de datos *Mysql*:

```
sudo mysql_secure_installation
```

```

root@jd:/home/jd# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

```

Figura 5: Instalacion de MySQL

se procede a configurar la base de datos on el siguiente comando:

```
sudo mysql -u root -p
```

Para añadir un usuario en la base de datos se coloca lo siguiente:

```

MariaDB [(none)]> CREATE DATABASE ampachedb;
MariaDB [(none)]> CREATE USER 'ampache'@'localhost' IDENTIFIED BY 'password';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON ampachedb.* TO 'ampache'@'localhost';
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;

```

```

password'' at line 1
MariaDB [(none)]> CREATE DATABASE ampachedb;
ERROR 1007 (HY000): Can't create database 'ampachedb'; database exists
MariaDB [(none)]> CREATE DATABASE ampachedb;CREATE USER 'ampache'@'localhost' IDENTIFIED BY 'password';
ERROR 1007 (HY000): Can't create database 'ampachedb'; database exists
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
MariaDB [(none)]> CREATE USER 'ampache'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON ampachedb.* TO 'ampache'@'localhost';
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> EXIT;
Bye

```

Figura 6: Instalacion de MySQL

Se procede a descargar y descomprimi Ampache de los repositorios de GitHub:

```

wget https://github.com/ampache/ampache/releases/download/4.1.1/ampache-4.1.1_all.zip
unzip ampache-4.1.1_all.zip -d /var/www/html/ampache

```

```

root@jd:/home/jd# wget https://github.com/ampache/ampache/releases/download/4.1.1/ampache-4.1.1_all.zip
--2023-04-16 17:21:45-- https://github.com/ampache/ampache/releases/download/4.1.1/ampache-4.1.1_all.zip
Resolviendo github.com (github.com)... 140.82.114.4
Conectando con github.com (github.com)[140.82.114.4]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://objects.githubusercontent.com/github-production-release-asset-2e65be/20230416%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230416T222145Z&X-Amz-Expires=300&X-Amz-Signature=...&key_id=0&repo_id=8933498&response-content-disposition=attachment%3B%20filename%3Dampache-4.1.1_all.zip
--2023-04-16 17:21:45-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/20230416%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230416T222145Z&X-Amz-Expires=300&X-Amz-Signature=...&key_id=0&repo_id=8933498&response-content-disposition=attachment%3B%20filename%3Dampache-4.1.1_all.zip
Resolviendo objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133
Conectando con objects.githubusercontent.com (objects.githubusercontent.com)[185.199.109.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 57106132 (54M) [application/octet-stream]
Guardando como: "ampache-4.1.1_all.zip"

ampache-4.1.1_all.zip 100%[=====]
2023-04-16 17:22:03 (3,09 MB/s) - "ampache-4.1.1_all.zip" guardado [57106132/57106132]

root@jd:/home/jd# unzip ampache-4.1.1_all.zip -d /var/www/html/ampache
Archive:  ampache-4.1.1_all.zip
creating: /var/www/html/ampache/lib/

```

Figura 7: Descarga e instalación de Ampache

Se cambia el propietario y grupo de todos los archivos y subdirectorios ubicados en el directorio `/var/www/html/ampache`.^{al} usuario y grupo `"www-data"`. El propósito de este comando garantiza que el usuario y el grupo del servidor web, que se ejecuta como `"www-data"`.^{en} Ubuntu, tengan los permisos adecuados para acceder y manipular los archivos de la aplicación Ampache.

```
chown -R www-data:www-data /var/www/html/ampache
```

Se reubica el directorio `.htaccess` en la carpeta correspondiente para Ubuntu 20.

```
cd /var/www/html/ampache
mv rest/.htaccess.dist rest/.htaccess
mv play/.htaccess.dist play/.htaccess
mv channel/.htaccess.dist channel/.htaccess
```

Creación de carpeta donde albergaran los archivos del servidor:

```
mkdir -p /data/Music
chown www-data:www-data /data/Music
```

Se edita la IP del servidor en el archivo ubicado en `etc/apache2/sites-available/ampache.conf`.

```
1 <VirtualHost *:80>
2     # The ServerName directive sets the request
3     # the server uses to identify itself. This i
4     # redirection URLs. In the context of virtu
5     # specifies what hostname must appear in the
6     # match this virtual host. For the default \
7     # value is not decisive as it is used as a l
8     # However, you must set it for any further \
9     #ServerName www.example.com
10
11     ServerName 192.168.18.12
12     DocumentRoot /var/www/html/ampache
13
14     # Available loglevels: trace8, ..., trace1
```

Figura 8: Configuración del archivo `.conf` de apache

Se activa el servicio de Virtual Host antes configurado y se reinicia el servidor web usando los comandos:

```
a2ensite ampache  
a2enmod expires rewrite  
systemctl reload apache2
```

En este momento ya se puede acceder a la configuración web de Ampache, en el navegador se coloca la dirección del host que en nuestro caso es 192.168.18.12.

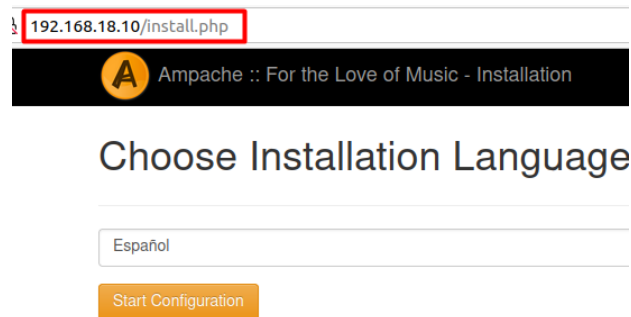


Figura 9: Entorno de inicio de Ampache

Se debe crear una cuenta inicial, para poder ingresar se debe colocar los parametros que se ingresaron cuando se procedió a configurar MariaDB.

33%

Paso 1 - Crear base de datos de Ampache

This step creates and inserts the Ampache database, so please provide a MySQL account with database creation privileges.

Paso 2 - Crear ficheros de configuración (ampache.cfg.php ...)

Paso 3 - Configurar la cuenta inicial

Introduzca Base de datos ampache

Nombre deseado para la base de datos:

Nombre del servidor MySQL:

Puerto MySQL (opcional):

Nombre administrativo de MySQL:

Contraseña administrativa de MySQL:

Crear base de datos ☒

Figura 10: Configuración de Ampache

Se coloca en modo por defecto para que aproveche de mejor manera los recursos que tiene a su disposición y además los reproductores que se van a usar

Installation Type

Optimize Ampache for your use case. The installer will enable/disable features automatically.

- ☒ Por defecto — Ampache is configured for personal use with the best features.
- ☐ Minimalista — Only essential features are enabled to simply stream your music from the Web Interface.
- ☐ Comunidad — Recommended settings when using Ampache as a frontend for a music community.

Allow Transcoding

Transcoding allows you to convert music from one encoding to another. Ampache supports on the fly transcoding. In order to transcode, Ampache takes advantage of existing binary applications such as ffmpeg, applications and ensure that they are executable by the web server.

This section applies the default transcoding configuration according to the application you want to use. You

Configuración de plantilla:

Reproductores

Ampache is more than only a web interface. Several backends are implemented to ensure you can stream your music. Select each backend to enable. Depending on the backend, you may need to perform additional configuration.

☒ Web Interface ☒ Ampache API ☒ Subsonic ☒ UPnP ☐ DAAP (iTunes) ☐ WebDAV

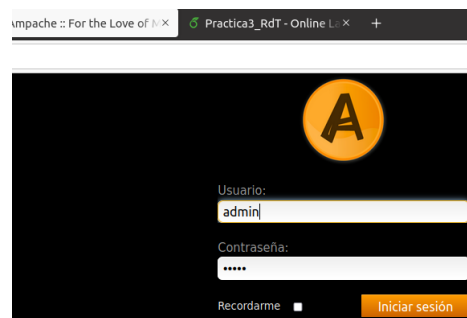
Figura 11: Configuración de Ampache

Se crea un usuario administrador para continuar con la configuración.

Crear cuenta de administrador

Usuario	<input type="text" value="admin"/>
Contraseña	<input type="password" value="*****"/>
Confirmar contraseña	<input type="password" value="*****"/>
<input type="button" value="Crear una cuenta"/>	

Figura 12: Creación de usuario Admin



Usuario:

Contraseña:

☐ Recordarme

Figura 13: Logeo al entorno de Apache con la cuenta de administrador creada

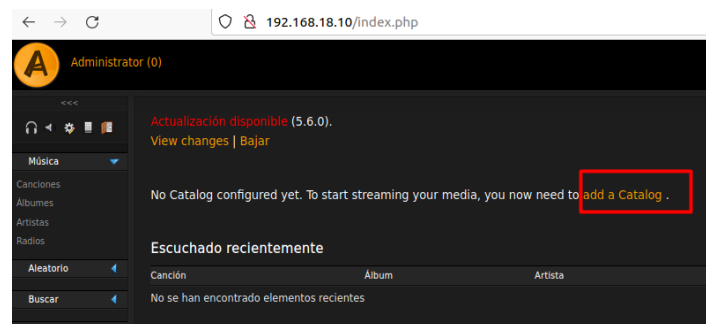


Figura 14: Entorno Ampache donde se puede agregar las categorías

En la pestaña de *configuraciones* se habilita la opción *Allow Video Features* que permite tener bibliotecas de vídeo.

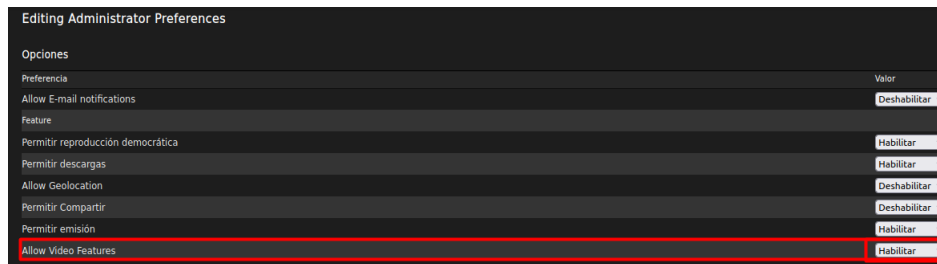


Figura 15: Configuración para vídeo

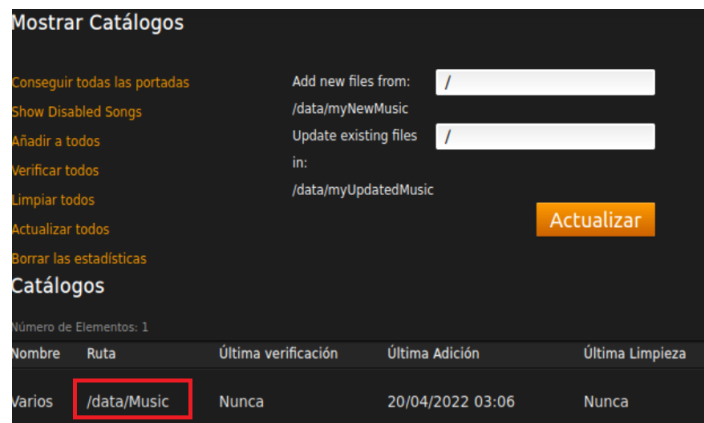


Figura 16: Creación de catalogo con la respectiva ruta de acceso a los videos en el servidor local

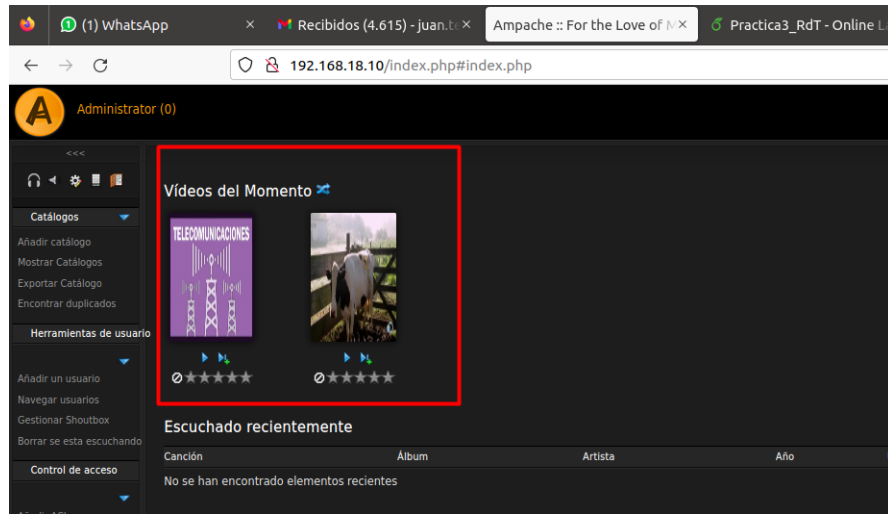


Figura 17: Vídeos disponibles en el servidor Loca

La plataforma también tiene opción para agregar usuarios y administradores a partir de la cuenta de administrador principal.

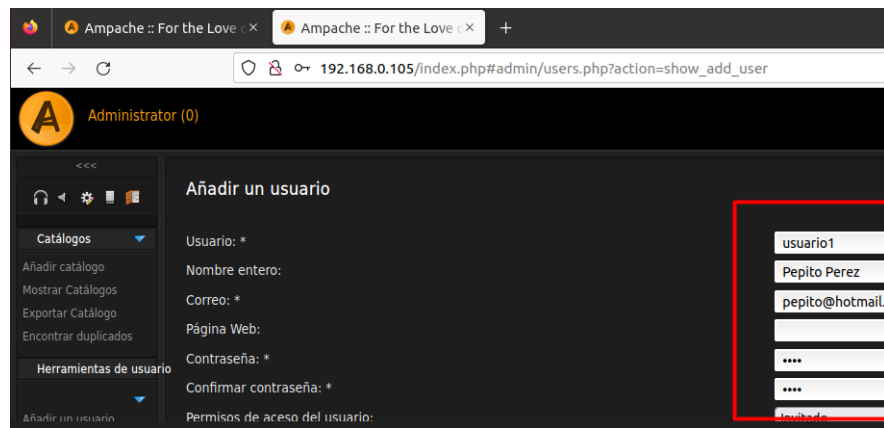


Figura 18: Añadir Usuario 1

Figura 19: Añadir Usuario 2

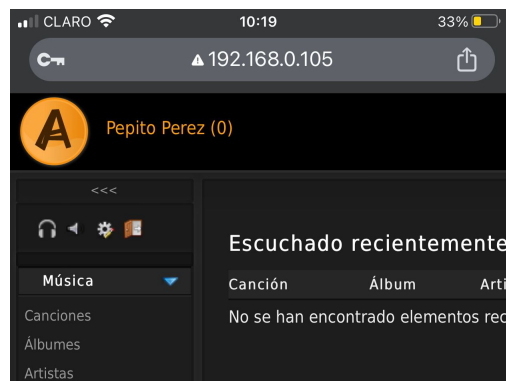


Figura 20: Logeo usando la cuenta del usuario 1

Análisis con Wireshark

A continuación se procede a capturar los paquetes cuando se realiza la reproducción de un vídeo.

Cuando se dio ha reproducir el video, se capturaron los paquetes mostrados en la Figura 21, el paquete analizado usa el protocolo HTTP, el paquete está utilizando el método POST para enviar información de autenticación a través del formulario de inicio de sesión. El parámetro */login.php* indica que la solicitud se está enviando a la página de inicio de sesión del servidor. La versión del protocolo HTTP utilizada es HTTP/1.1, los indices *r* y *n* al final de la solicitud HTTP indica que se ha terminado de enviar la solicitud y se está esperando la respuesta del servidor, además se puede observar el usuario que en este caso de análisis se lo esta haciendo a través de un iPhone.

```

707 78.554027291 192.168.1.102 192.168.1.100 TCP 68 61162 → 80 [ACK] Seq=1 Ack=1 Win=1
708 78.554679642 192.168.1.102 192.168.1.100 TCP 697 61162 → 80 [PSH, ACK] Seq=1 Ack=1
709 78.554781101 192.168.1.102 192.168.1.100 TCP 68 80 → 61162 [ACK] Seq=1 Ack=630 Win=
710 78.555111239 192.168.1.102 192.168.1.100 HTTP 535 POST /login.php HTTP/1.1
711 78.555140532 192.168.1.102 192.168.1.100 TCP 68 80 → 61162 [ACK] Seq=1 Ack=1097 Win=
720 80.394818968 192.168.1.102 224.0.0.251 IGMPv2 48 Membership Report group 224.0.0.251
721 80.453115169 192.168.1.102 224.0.0.251 MDNS 144 Standard query 0x0000 PTR 1b. dns-
Frame 710: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 192.168.1.100
Transmission Control Protocol, Src Port: 61162, Dst Port: 80, Seq: 630, Ack: 1, Len: 467
Hypertext Transfer Protocol
  POST /login.php HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /login.php HTTP/1.1\r\n]
    Request Method: POST
    Request URI: /login.php
    Request Version: HTTP/1.1
    Host: 192.168.1.100\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
    Content-Type: multipart/form-data; boundary=----WebKitFormBoundary0q3AKnmKOV2s1S32\r\n
    Origin: http://192.168.1.100\r\n
    User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko)
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Referer: http://192.168.1.100/login.php\r\n
    Content-Length: 467\r\n
    Cookie: ampache=tdvi29bo2ghv1ia3ioai1javjd\r\n
  \r\n
  [Full request URI: http://192.168.1.100/login.php]
  [HTTP request 1.0]
  [Response in frame: 847]
  [Next request in frame: 852]
  File Data: 467 bytes
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundary0q3AKnmKOV2s1S32"

```

Figura 21: Comienzo de la transmisión

Luego se analiza un siguiente paquete que usa el protocolo igualmente HTTP pero con la información de un GET como se observa en la Figura 22. El paquete está utilizando el método GET para solicitar la página de inicio del servidor. El parámetro `/index.php` indica que se está solicitando la página principal del servidor. La versión del protocolo HTTP utilizada es HTTP1.1, los parámetros `r` y `n` al final de la solicitud HTTP indica que se ha terminado de enviar la solicitud y se está esperando la respuesta.

```

848 87.996057697 192.168.1.100 192.168.1.102 TCP 981 [TCP Retransmission] 80 → 61162 [PSH, ACK] Seq=2897 Ack=1097 Win=
849 88.079377631 192.168.1.102 192.168.1.100 TCP 68 61162 → 80 [ACK] Seq=1097 Ack=1449 Win=0 TSval=2243
850 88.082343293 192.168.1.102 192.168.1.100 TCP 68 61162 → 80 [ACK] Seq=1097 Ack=3810 Win=128704 Len=0 TSval=2243
851 88.082344221 192.168.1.102 192.168.1.100 HTTP 535 GET /index.php HTTP/1.1
852 88.109157031 192.168.1.102 192.168.1.100 TCP 68 80 → 61162 [ACK] Seq=3810 Ack=1630 Win=0 TSval=128704 Len=0
853 88.109245267 192.168.1.100 192.168.1.102 TCP 68 80 → 61162 [ACK] Seq=3810 Ack=1630 Win=64128 Len=0 TSval=39906
854 88.172050002 192.168.1.100 192.168.1.102 TCP 1516 80 → 61162 [ACK] Seq=3810 Ack=1630 Win=64128 Len=1440 TSval=39906
Frame 852: 681 bytes on wire (4808 bits), 681 bytes captured (4808 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 192.168.1.100
Transmission Control Protocol, Src Port: 61162, Dst Port: 80, Seq: 1097, Ack: 3810, Len: 533
Hypertext Transfer Protocol
  GET /index.php HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /index.php HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /index.php
    Request Version: HTTP/1.1
    Host: 192.168.1.100\r\n
    Cookie: ampache=ee7rqf7odgn9n3sq41hntmcb; ampache_lang=es_ES; ampache_user=admin\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) CriOS/12.0.5615.46 Mobile/
    Referer: http://192.168.1.100/login.php\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
  \r\n
  [Full request URI: http://192.168.1.100/index.php]
  [HTTP request 2.0]
  [Prev request in frame: 710]
  [Response in frame: 853]

```

Figura 22: Comienzo de la transmisión

En la Figura 23 se muestra los primeros paquetes que generaron cuando se comenzaron a transmitir los paquetes del video, El protocolo de control de transmisión (TCP) se utiliza para establecer una conexión confiable entre dos dispositivos de red y garantizar que los datos se entreguen correctamente.

En este caso de análisis se presentaron las siguientes flags:

- SYN: Este flag se utiliza para iniciar una conexión TCP, cuando se envía un paquete con el flag SYN establecido, se está solicitando al otro extremo que inicie una conexión y sincronice los números de secuencia y efectivamente en el paquete 3771 se tiene la respuesta por parte del servidor.
- ECE: Este flag indica que se ha detectado una congestión en la red y se ha habilitado el bit ECE (Explicit Congestion Notification) en los paquetes TCP.
- CWR: Este flag indica que el emisor ha recibido un paquete con el bit ECE establecido y ha reducido su tasa de envío para evitar una congestión en la red.
- PSH: Este flag se utiliza para indicar que los datos en el paquete deben entregarse al destinatario lo antes posible, sin esperar a que se acumule más información.

Luego que se sincronizaron y se comunicara el estado de tráfico en la conexión, se pudo observar el protocolo HTTP/1.1, pertenece a un protocolo de aplicación utilizado para la transferencia de datos en la World Wide Web. El código de respuesta "200 OK" indica que la solicitud del cliente ha sido procesada correctamente y que el servidor está devolviendo la información solicitada.

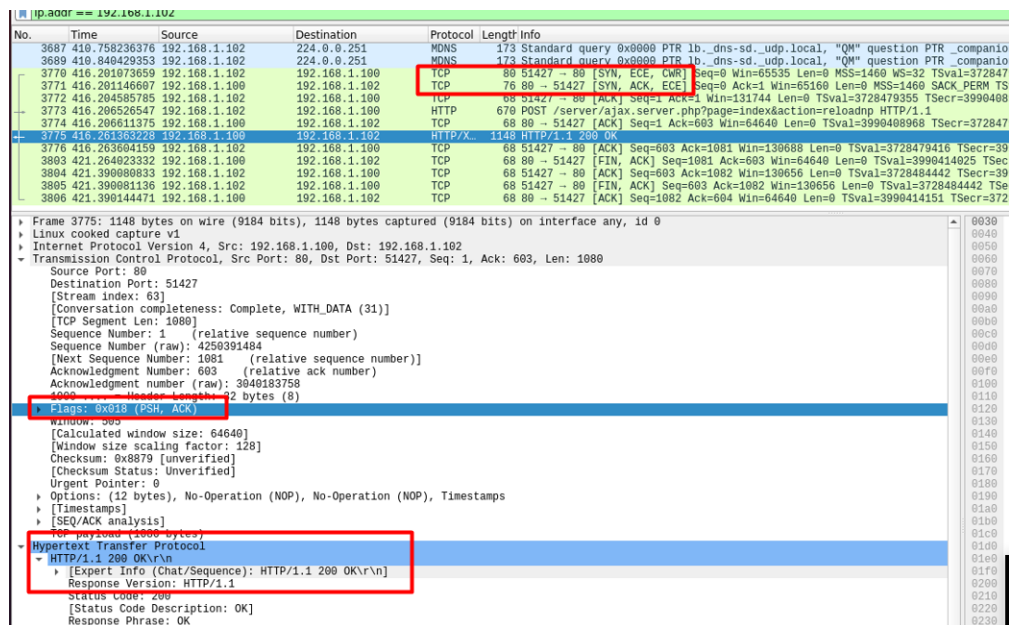


Figura 23: Fin del envío de paquetes Servidor - Cliente

En la Figura 24 ve que cuando todos los paquete de datos del video se enviaron, los últimos paquetes están bajo el protocolo TCP y en el parámetro de *flag FIN – ACK* indica que se está realizando una terminación de conexión desde ambos lados de la conexión. El *FIN* indica que el remitente ha terminado de enviar datos, y el *ACK* indica que el receptor ha confirmado la terminación.

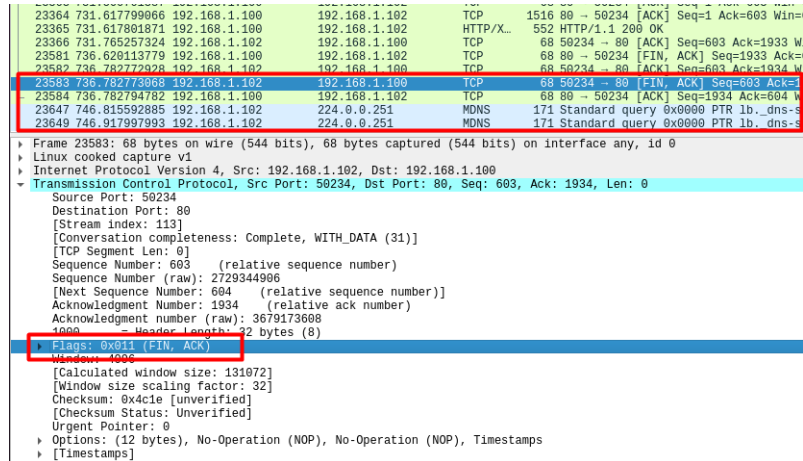


Figura 24: Fin del envío de paquetes Servidor - Cliente

4. Conclusiones

Después de llevar a cabo la práctica relacionada con la implementación de un servidor VoD, se obtuvieron las siguientes conclusiones:

- El servidor VoD implementado utiliza el protocolo TCP para la transmisión de datos. Aunque no es el protocolo más óptimo para una red pequeña con pocos clientes, resulta ser una buena alternativa.
- La interpretación de los flags TCP y el protocolo HTTP en Wireshark puede brindar una perspectiva valiosa sobre la conexión, el desempeño de la red y posibles riesgos de seguridad. No obstante, para interpretar adecuadamente la información obtenida, es necesario contar con habilidades avanzadas en redes y protocolos TCP/HTTP.
- Durante el análisis de paquetes, se pudo observar el uso de diferentes flags por parte del protocolo TCP para la transmisión de datos. Estos incluyen flags utilizados para sincronización, indicadores de congestión, inicio y finalización de la transmisión de paquetes, entre otros.
- Es recomendable considerar el uso de una dirección IP estática para montar el servidor, ya que esto evita la necesidad de reconfigurarlo en caso de reinicio del router.

Referencias

1. J. Márquez Díaz, L. Sampedro, and F. Vargas, “Instalación y configuración de apache, un servidor web gratis ingeniería y desarrollo, núm. 12, diciembre, 2002, pp. 10-23 universidad del norte barranquilla, colombia,” *Ingeniería y desarrollo*, no. 12, pp. 10–23, 2002.
2. U. González Llaguno, “Contribuciones a un proyecto open source de ámbito internacional: Ampache,” 2021.
3. M. Ahmed, M. M. Uddin, M. S. Azad, and S. Haseeb, “Mysql performance analysis on a limited resource server: Fedora vs. ubuntu linux,” in *Proceedings of the 2010 Spring Simulation Multiconference*, 2010, pp. 1–7.
4. G. G. Urtiaga, *Administrar MySQL y MariaDB: Aprende a administrar MySQL y MariaDB fácilmente*. AprendeIT, 2020.