**UNIVERSIDAD NACIONAL DE CÓRDOBA**

**FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES.**

**ASIGNATURA**

Comunicaciones de Dato

**CARRERA**

INGENIERIA EN COMPUTACION

**Trabajo Práctico I: Capa de Aplicación**

**Integrantes:**

* Contreras, Iván Elias
* Alaniz Ferreyra, Alvaro
* Yépez Hinostroza, Franz Alexánder
* Malano,Leandro

Introducción

Para el siguiente trabajo práctico utilizaremos una máquina virtual en la que se montará un Servidor Web corriendo bajo el sistema operativo Ubuntu Server 16.04.3 LTS.

Seguido de definir el socket del programa interpretado por python , se hará uso de los métodos Get y Post del protocolo HTTP y se procederá a analizar el tráfico por medio del sniffer WireShark.

Del lado del cliente, se utilizará la herramienta netcat para hacer uso de la capa de transporte.

El servidor tiene la ip privada 192.168.1.45 y el puerto habilitado para el script de python es el 4000.

Al usar WireShark se filtró la búsqueda para que sólo mostrase del protocolo de aplicación HTTP.

Método Get sin parámetros

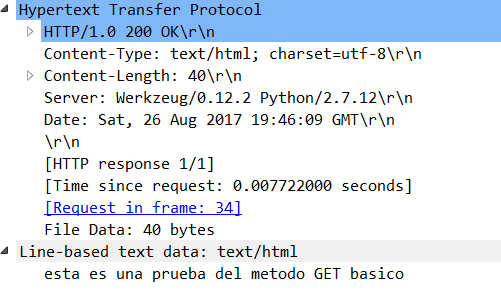
La siguiente imagen muestra lo que capturó WireShark en la sección de capa de aplicación ante la petición de Get del cliente.



Se observa en la sección de Host el socket del servidor.

En la sección User-Agent se puede apreciar que la petición fue hecha desde el Navegador Mozilla versión 5.0, corriendo desde un Sistema Operativo Windows.

La respuesta de la petición hecha se muestra a continuación:



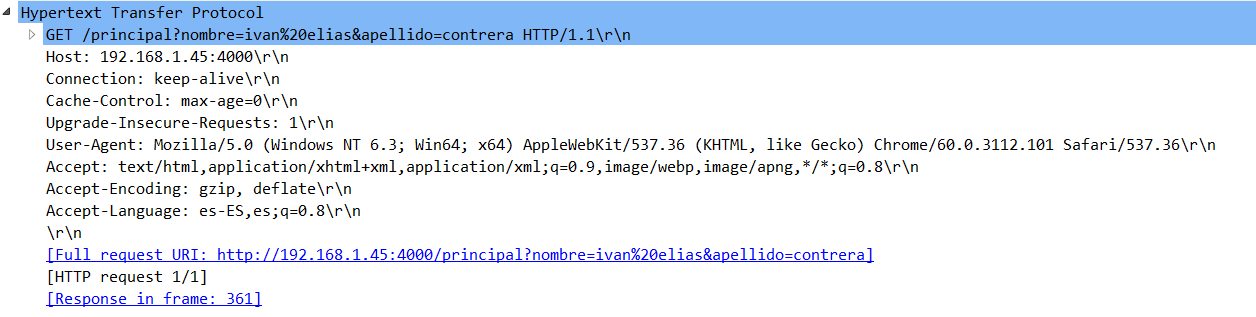
HTTP /1.0 200 = El código 200 hacer referencia a que fue una petición correcta.

Content-Length: 40 = el tamaño de la cadena que envía, en bytes.

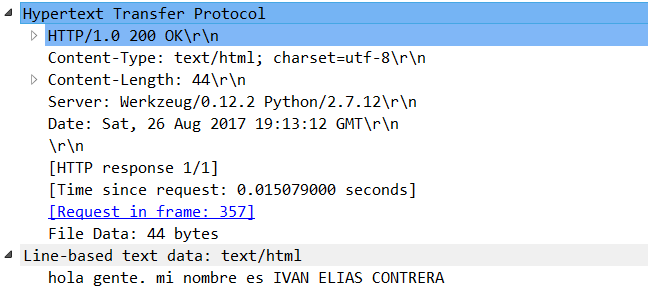
Se observa que el pedido se ubica en la trama 34, según la ordenación del WireShark.

Debajo de la sección correspondiente a la de capa de aplicación se observa la respuesta del servidor, en texto plano.

Método Get con parámetro



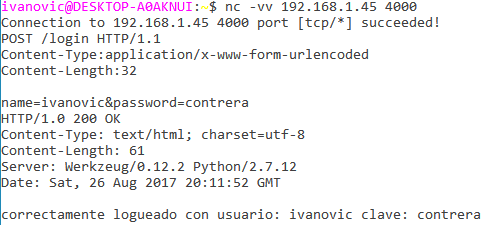
Se observa que en este caso los parámetros son pasado en texto plano sin ninguna clase de encriptación, por que que quedaría perfectamente visible por ejemplo en el historial del navegador.



Método Post

El método Post del protocolo HTTP , a diferencia del Get, su contenido va en el body de a petición, por lo que no aparecería no aparecería su contenido en la url de un navegador.

En la siguiente imagen se adjunta la petición y respuesta mediante la herramientas netcat:



Captura de la respuesta por WireShark

