

# Laboratorio HTTP

Francisco Bolzan - Lucio Trinchero - Joel Torboli

October 27, 2018

## 1 Introducción

Para comenzar, leímos la documentación de los slides sobre Webserver proveniente de la página de Netkit, así como el laboratorio y su configuración básica. Aquí aprendimos sobre la herramienta 'links' para acceder a un servidor web, así como la configuración del mismo y los archivos necesarios. Esto nos permitió especular como sería el funcionamiento si el laboratorio fuese ampliado, pero como no había más documentación que leer, proseguimos al desarrollo de las actividades solicitadas.

## 2 Envío de datos

Como primer paso, creamos un archivo 'index.html', el cual posee la simple función de enlazar a 2 formularios; uno que envía los datos a través de GET y otro mediante POST. Tras esto activamos la captura de tramas con el siguiente comando:

```
tcpdump -w /hosthome/'captura'.pcap
```

Finalizado esto, mediante 'links' ingresamos al servidor mediante su IP, donde completamos el formulario (Tanto mediante POST como GET).

Una vez realizado el paso anterior, estas tramas fueron analizadas con WireShark, donde nos centramos en buscar las diferencias entre GET y POST y su funcionamiento.

Al analizar el envío de datos a nivel de protocolo, nos dimos cuenta que GET envía los mismo a través de la URL de la página, situación que puede llegar a ser indeseable cuando se trabaja con información privada o personal, tal como contraseñas, tarjetas de crédito, etc. Por su parte, POST envía los datos como un paquete 'oculto', por lo que si alguien captura la URL del usuario, no podrá extraer ninguna información de esta misma.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

Agregar imágenes, señalando o mostrando la diferencia entre POST y GET.

## 3 Persistencia

HTTP 1.1 es persistente ya que utiliza TCP para mantener una pseudo sesión por la cual se envían y reciben los paquetes. El tiempo de dicha conexión está dictado por Apache, normalmente van de 15 a 5 segundos, lo que permite que durante estas ventanas de conexión múltiples

paquetes sean enviados/recibidos pero tampoco saturan al servidor con muchas conexiones constantes y paralelas.

## 4 DNS y WWW

La WWW necesita del DNS para iniciar el proceso de enrutamiento, en el cual, utilizando un buscador se introduce una URL la cual DNS asocia con su respectiva IP de servidor, HTTP luego procesa el resto del enrutamiento para así obtener al principio el HTML básico de la página y luego pedir el resto de paquetes necesarios.

Falta agregar como lo configuramos

Falta agregar como lo configuramos

Falta agregar como lo configuramos

Falta agregar como lo configuramos

Falta agregar como lo configuramos

Falta agregar como lo configuramos

Falta agregar como lo configuramos

## 5 Configuración

La configuración de los subdominios y sus alias fue realizada en el router de la red ya que este también actúa de NS. Donde se asignó el subdominio "www.tec" al primer servidor con IP 11.0.0.2 y "www.mit" al segundo con IP 12.0.0.2.

## 6 Apache

Para la actividad final modificamos los DNS para que apunten a uno solo de los servidores y que el redireccionamiento quede a cargo de Apache. Para esto lo primero fue modificar los archivos DNS, solo fue necesario cambiar la IP de algún servidor por la del otro y listo. Para que Apache reconociese a ambos servidores individualmente primero creamos dos directorios, uno para cada servidor con páginas distintas dentro de cada uno. Buscando un poco sobre las opciones y configuración de Apache descubrimos que el mismo posee una carpeta de sitios disponibles dentro de la cual están los .conf de dichas páginas, con esto pudimos modificar dichos archivos para que cada uno supiese el nombre de su servidor, el alias y donde buscar la página del mismo. Una vez hecho esto con ambos servidores desactivamos los sitios "default" de Apache con `a2dissite default` (en `/etc/apache2/sites-available`), luego activamos los nuestros con `a2ensite`, reiniciamos Apache y comprobamos satisfactoriamente el resultado.