**Universidad Nacional Autónoma de Nicaragua**

**UNAN-León**

**Facultad de Ciencias y Tecnología**

**Departamento de Computación**

**Ingeniería en Telemática**

**V año**

****

**Componente: Laboratorio de Seguridad de Redes**

**Tema: HTTP + TLS**

**Realizado por:**

**Br. Jhonatan Uziel Espinoza Ortega Carnet: 15-00737-0**

**Dirigido a:**

**MSc. Aldo Martinez**

**León, Nicaragua lunes 8 de julio del 2019.**

**1 Introducción**

Se denomina genéricamente servidor Web seguro a un servidor Web con soporte del protocolo HTTPS (HTTP por encima de SSL). Nuestro objetivo será construir un servidor Web seguro a partir del servidor Web apache (http://www.apache.org). Apache es el servidor Web más utilizado en la red, entorno al 60% de los sitios Web utilizan apache. Hay dos formas de añadir soporte de

HTTPS a apache:

 Utilizando apache\_ssl.

 Utilizando mod\_ssl.

Ambas formas proporcionan una interfaz al servidor Web para utilizar TLS y se basan en una implementación de dicho protocolo proporcionada por el paquete OpenSSL. Independientemente de si utilizamos una u otra estaremos obligados a instalar dicho paquete que podemos encontrar en http://www.openssl.org. Nosotros utilizaremosmos\_ssl que no es más que un módulo de los disponibles para apache (como mos\_php para poder utilizar PHP, por ejemplo) y que se configura mediante una serie de directivas que tendremos que incluir en la configuración del servidor.

**2 Descripción**

Los pasos a seguir son los siguientes:

 Instalar el servidor Web Apache.

 Instalar OpenSSL.

 Generar los certificados del servidor.

 Configurar el mod\_ssl (Crear un Host Virtual).

 Generar los certificados del cliente.

 Comprobar el buen funcionamiento del sistema.

**2.1 Configuración del servidor**

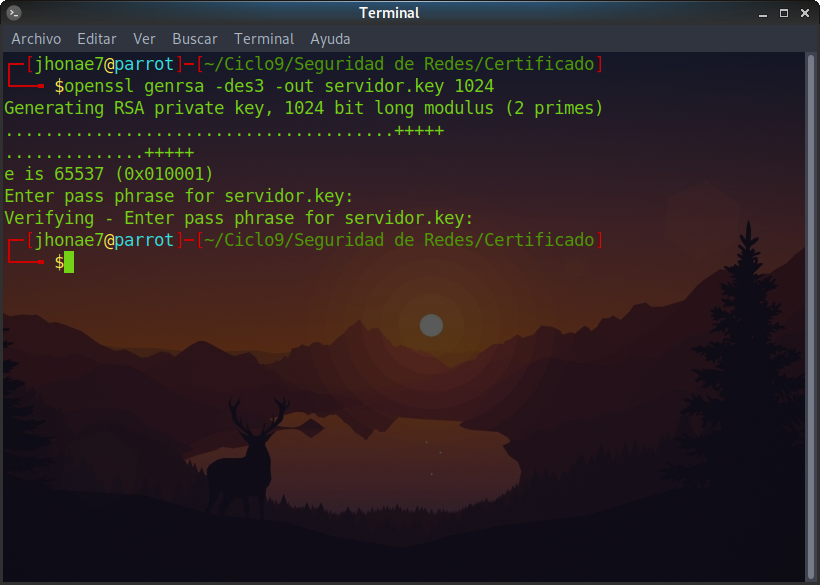
Los pasos a seguir de cara a la instalación del servidor seguro son:

 Generación del certificado del servidor.

 Instalación y configuración del certificado del servidor.

 Configuración de apache.

**2.2 Generación del certificado del servidor**



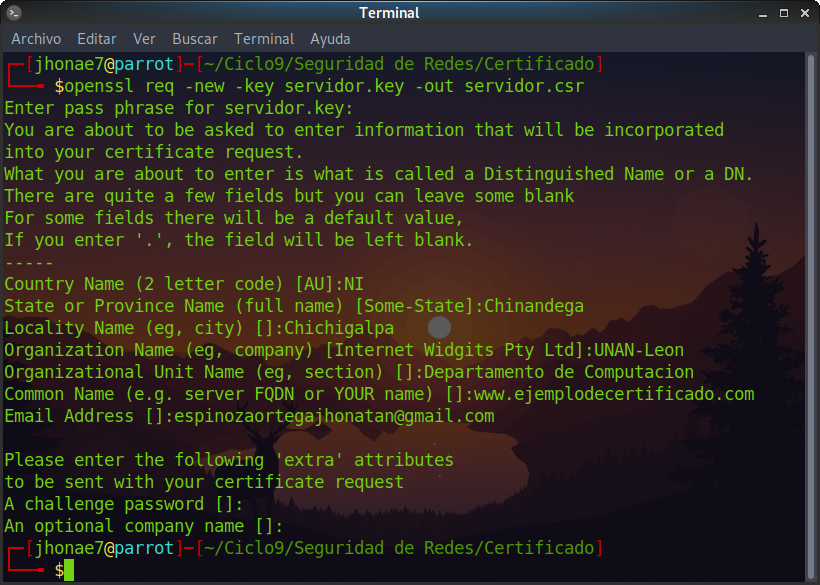
Donde:

\* genrsa: genera un par de llaves (clave publica y privada) con RSA

\* des3: es el algoritmo para cifrar las claves antes de emitirlas

\* out: especifica el archivo donde se almacenaran las claves generadas

\* 1024: indica que el tamaño de las claves es de 1024 bits



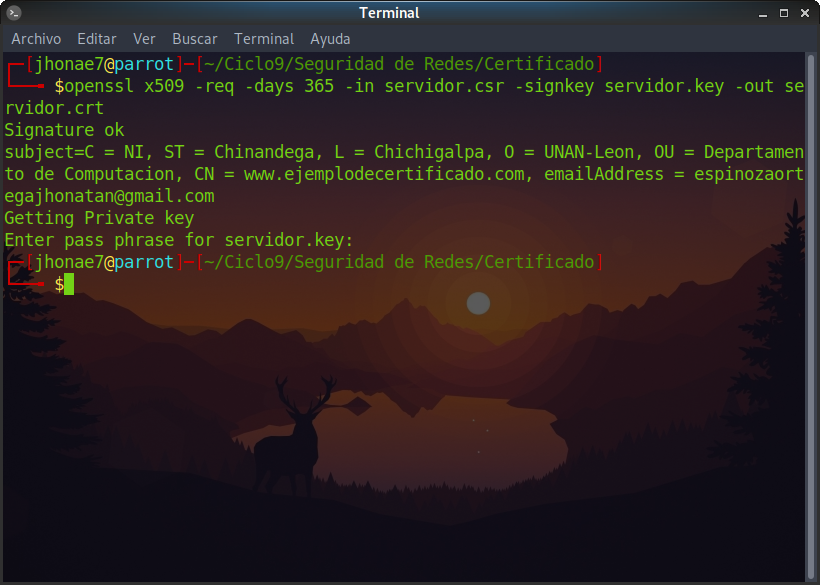
Donde:

\* req: crea y procesa solicitudes de certificados.

\* new: genera una nueva solicitud de certificado, pidiendole al usuario los valores de campo revelante

\* key: se pasan las claves

\* out: se indica el archivo donde se almacenara el resultado de la operacion

Donde:

\* x509: es para firmar el certificado

\* req: solicita y procesa el certificado

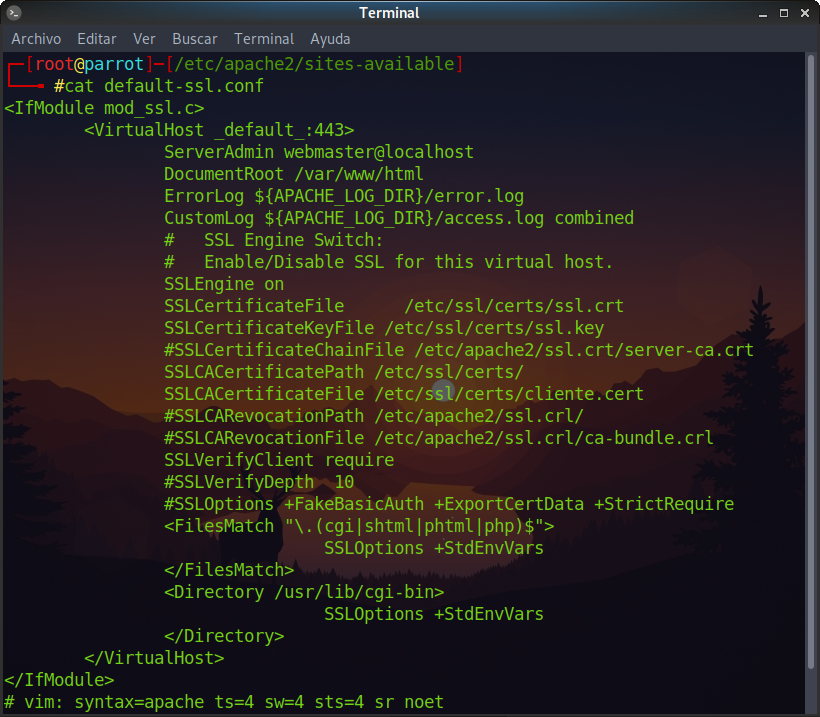
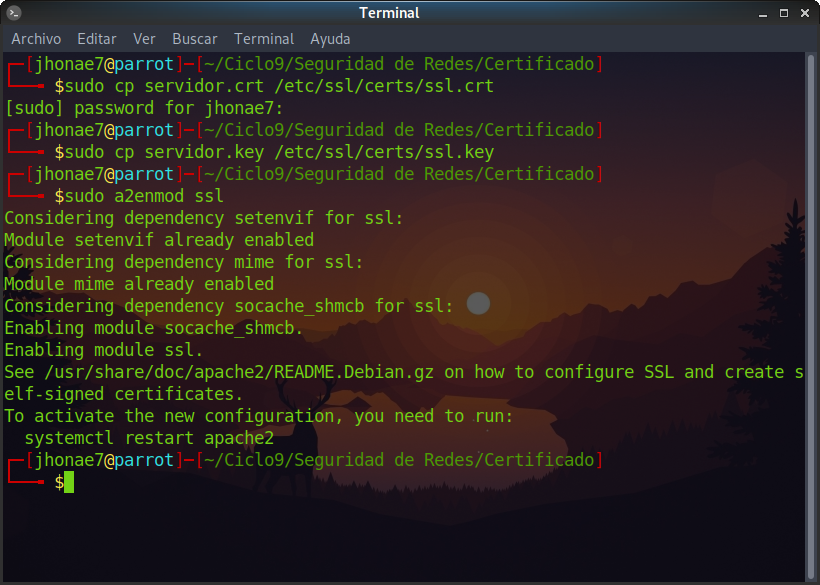
\* days: se indica en dias el periodo de validez del certificado

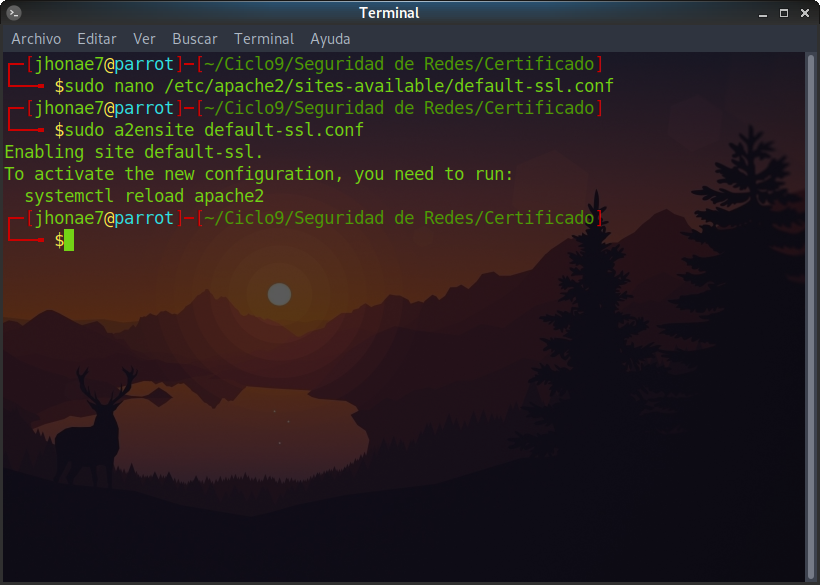
\* in: se pasa el archivo que contiene la solicitud de certificado

\* signkey: firma el certificado con las claves

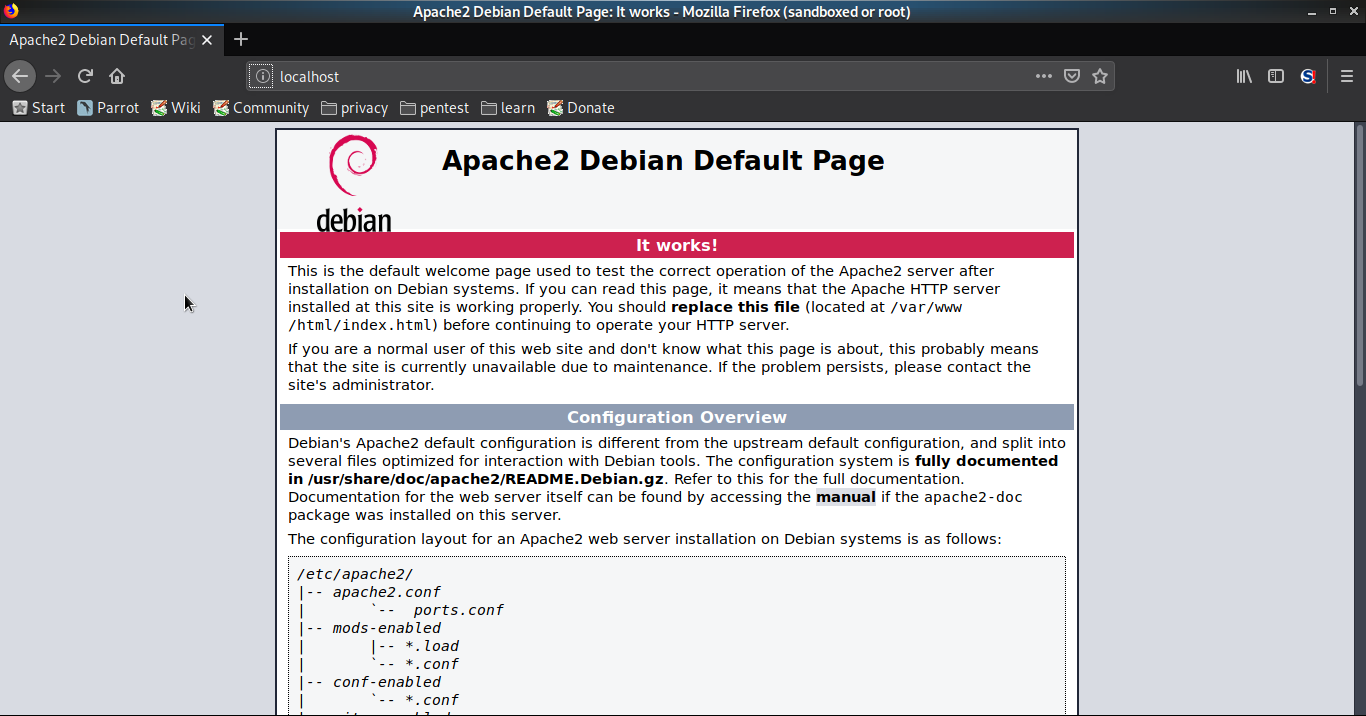
\* out: indica el archivo de salida que contiene el certificado firmado

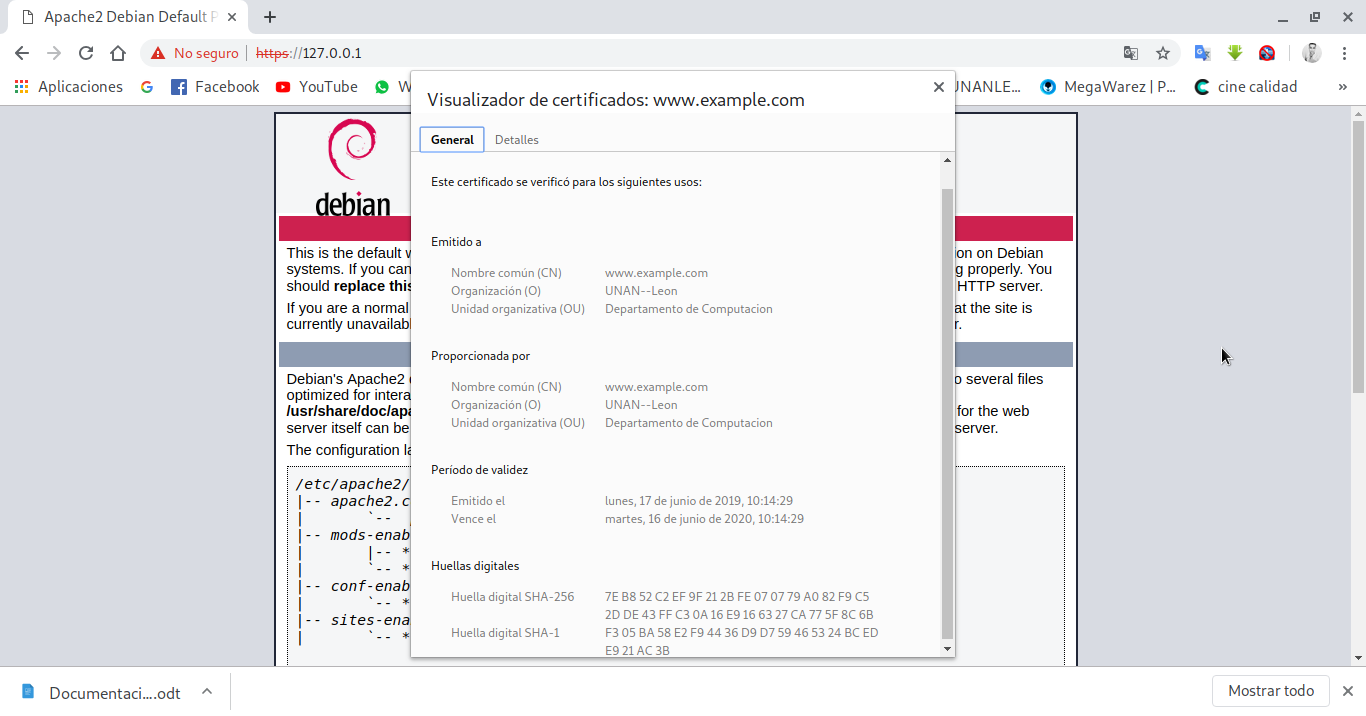
**2.3 Configuración**



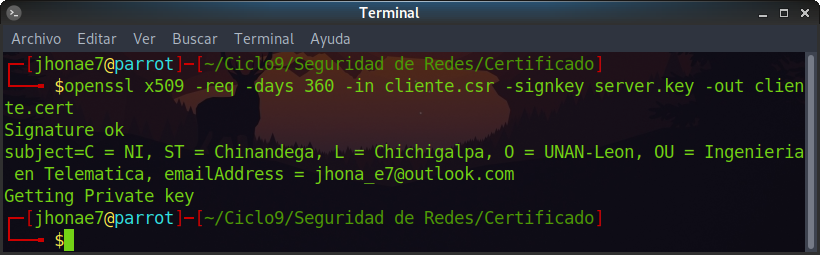
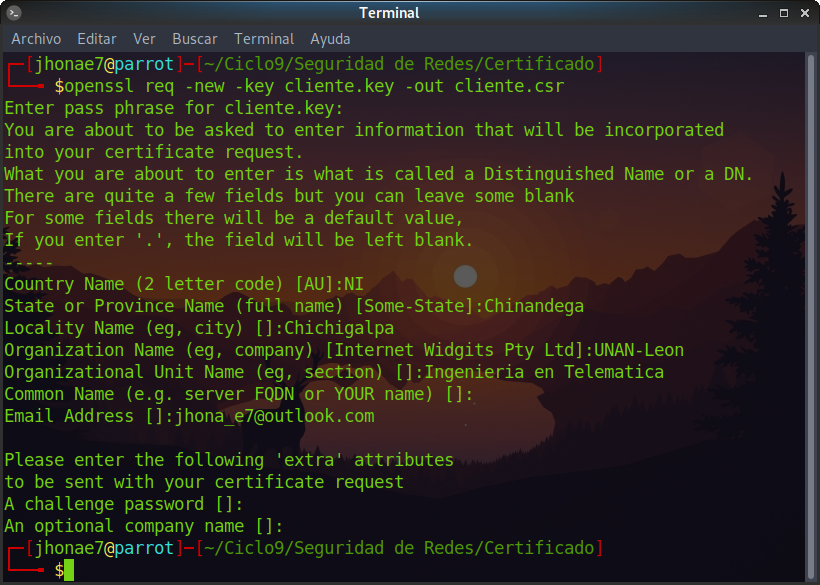
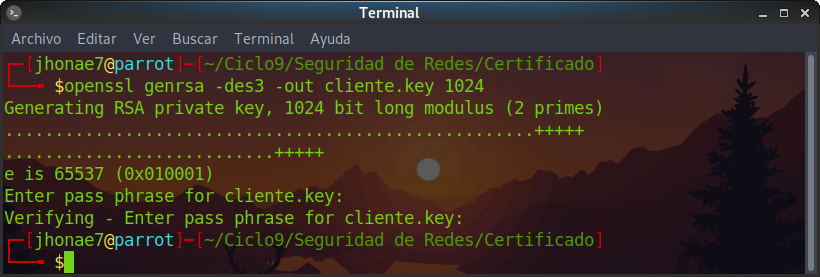


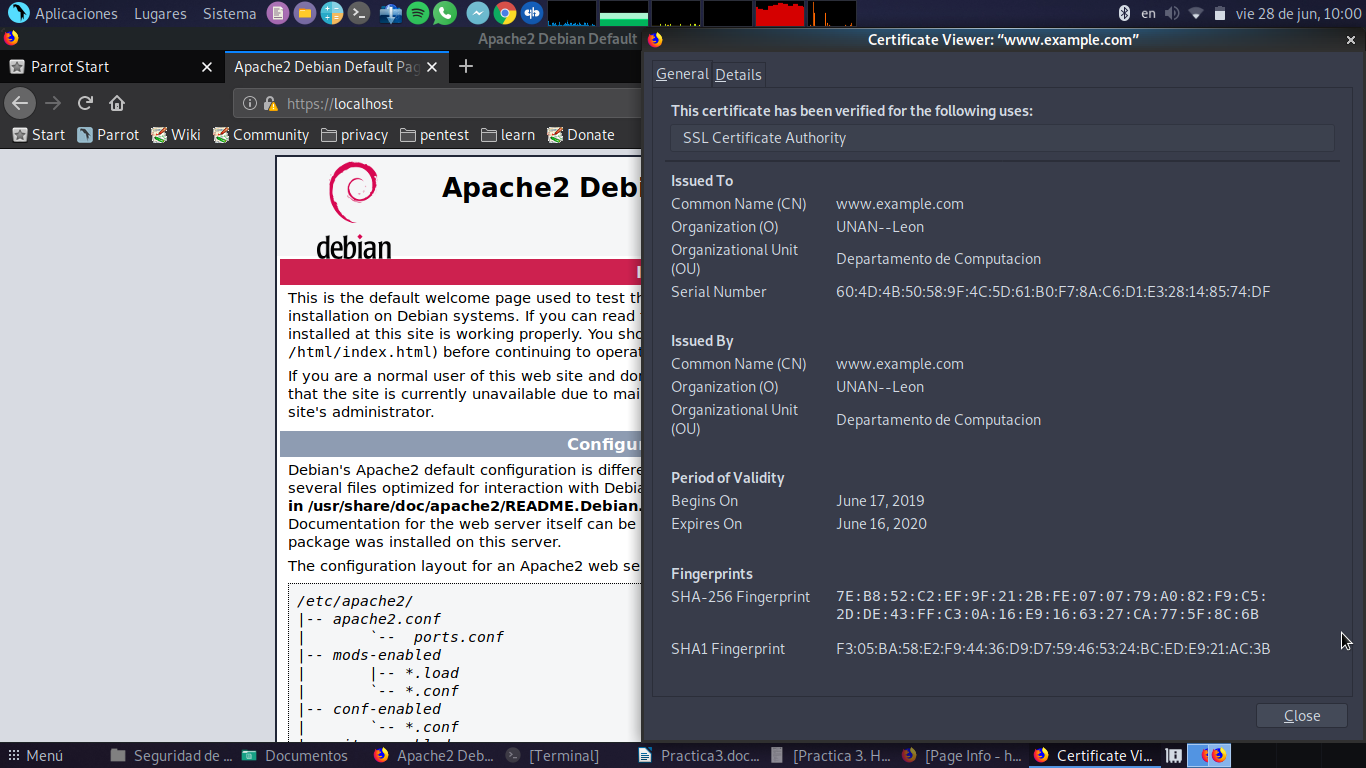
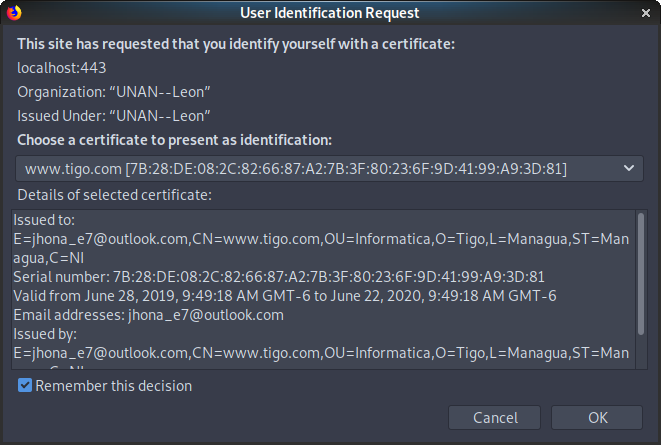
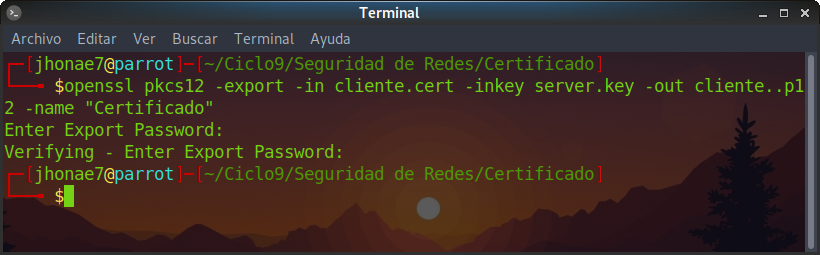
**3. Pruebas**

****

****

**4. Acceso utilizando el navegador WEB**

****

****