PRACTICA: WEB SERVERS VIRTUALES (RESUMEN TEMA 2).

Trabajo realizado por: Javier Epifanio López

Actividad Resumen de las prácticas.

1. **Recuerda las ventajas de los servidores web virtuales (¡¡¡¡no confundir con máquinas virtuales tipo VirtualBox!!!!).**
   Más rápido y mejor para archivos estáticos.
   En comparación con **Apache**, ofrece 4 veces más conexiones simultáneas.
   Compatibilidad.
   Apoyo de equilibrio.
   Sitios más rápidos y SEO mejorado.

2. **Indica en al menos 5 pasos (y obviando la instalación de Apache/Nginx) , el procedimiento para crear servidores web virtuales y su configuración. Indica ficheros, directorios y directivas implicadas.**
   a. **Hosts virtuales basados en Nombres.**
      1. Crear las carpetas donde van a estar nuestros sites /var/www/html/site
      2. Asignar los permisos a los archivos user www-data
      3. Modificar el fichero de conf de nuestro site que se encuentra en /etc/nginx/sites-avaliable/site
      4. Configurar el fichero de conf añadiendo otro server en el mismo fichero cambiando el server_name a tu otro site.
      5. Comprobar que la sintaxis es correcta sudo nginx -t. y restart a nginx.service.
      6. Comprobar que funciona en nuestro cliente.

   b. **Hosts virtuales basados en Puertos.**
      1. Activar los puertos que necesites con ufw
      2. Crear o utilizar el site anterior para el puerto 8080
      3. Modificar el fichero conf de nuestro site, añadiendo un server en el que escuche por el puerto 8080 y añadimos el server_name y root
      4. Comprobar la sintaxis del fichero con sudo nginx -t y restart nginx
      5. Comprobar que funciona en el cliente.

   c. **Control de acceso**
      1. Configurar en nuestro site en el fichero de conf donde vamos a guardar los ficheros de errores (error.log y Access.log).
      2. Personalizamos los mensajes de error que queremos que se muestren.
      3. Y lo añadimos en el fichero de nuestro site con error_page 404 y la ruta a partir del root
      4. Podemos añadirlo en cualquier sitio en el location /ruta que quieras.
      5. Comprobar sintaxis y reiniciar servicio.
      6. Comprobar en el cliente que funcione.

d. Autenticación

1. Instalar la paquetería apache2-utils
2. Podemos activar la auth_basic en todo nginx o en cualquier sitio de nuestro site en mi caso lo hago en mi site (dentro del location).
3. Crear el archivo de contraseñas y añadimos los usuarios que queremos.
4. Comprobar sintaxis del fichero y reiniciar el servicio.
5. Comprobar que funcione el cliente.

3. Servidor web seguro.

a. Pasos para crear un servidor web seguro con auto certificado. Indica también cómo se accede ahora a tu servidor web seguro.

1. Instalar OpenSSL -> sudo apt-get install openssl.
2. Modificar el fichero openssl.conf con nuestra información.
3. Crear el directorio donde vamos a guardar .key
4. Generar el .pem y .key
5. En nuestro fichero de conf del site creamos un server y que escuche por el puerto 443 y añadimos la conf de nuestro site.
6. Añadimos en el server que escucha por el puerto 80 un return para que se rediga siempre a https (Web Segura).

b. Diferencia entre lo implementado en clase y Let'sEncript (o CAcert)

La diferencia es lo que hicimos en clase es un certificado autofirmado y el navegador nunca podría reconocerlo, sin embargo el de CACert lo provee una entidad certificadora y sí que lo puede llegar a reconocer si nosotros lo especificamos