**Tutoría de Despliegue de Aplicaciones Web 27/10/2022:**

**Creación de virtualhosts:**

**Creamos un directorio en /var/www/miseguridad.com**:

Mkdir miseguridad.com

Cd miseguridad.com/

Mkdir public\_html

**Para crear usuarios y la autenticación de una página**: Sudo apt install apache2-utils

**Para crear un fichero donde estarán nuestros usuarios**:

Sudo htpasswd -c/var/www/miseguridad.com/.usuarios paco : se crea este fichero, en el que se almacenan los usuarios que tendrán acceso a nuestra página web. Paco es el nombre de usuario

Nos pide una contraseña para el usuario, y se la damos

Si ahora hacemos cat .usuarios, veremos el usuario y un hash de la contraseña, no la contraseña original.

**Creamos una entrada virtualhost para miseguridad.com:**

Cd /etc

Sudo gedit hosts

127.0.0.1 miseguridad.com

Cd /etc/apache2/sites-available

Ls -la

cp 000-default.conf miseguridad.com.conf

sudo gedit miseguridad.com.conf

**Editamos el ServerName**, estableciéndolo a miseguridad.com

**Editamos el DocumentRoot** a /var/www/miseguridad.com/public\_html

Añadir una **directiva llamada Directory**. Esta permite, a ciertas carpetas, darle ciertos permisos.

<Directory “/var/www/miseguridad.com/public\_html”>

AuthType Basic

AuthName “autenticación con usuario y contraseña”

AuthBasicProvider file

AuthUserFile “/var/www/miseguridad.com/.usuarios

Require user paco

</Directory>

Guardamos el fichero.

**Falta activar el virtualhost**: a2ensite miseguridad.com

Y reiniciar Apache: sudo service apache2 restart

Ahora, para probar si funciona, creamos una página básica en /var/www/miseguridad.com/public\_html/index.html

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Abrimos el navegador web e intentamos acceder a miseguridad.com

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Esto va dentro del propio protocolo HTTP.

**Si queremos crear más de un usuario para un mismo virtualhost**:

Sudo htpasswd /var/www/miseguridad.com/.usuarios Antonio

Le ponemos contraseña, y listo.

**Si visualizamos el fichero de usuarios**:

cd /var/www/miseguridad.com

cat .usuarios

Texto

Descripción generada automáticamente

**Añadimos el usuario al fichero de virtualhost**:

Sudo gedit miseguridad.com.conf

Y añadimos Antonio a la última línea:

Texto

Descripción generada automáticamente

**Reiniciamos Apache**: sudo service apache2 restart

Y ahora si entramos a la página, nos debería dejar entrar con el usuario Antonio.

**Podemos crear grupos** (a modo de roles), para agrupar a distintos usuarios sin tener que añadirlos manualmente:

**Activamos el módulo**: sudo a2enmod authz\_groupfile

Si nos vamos a la lista de módulos cd mods-enabled/, vemos los dos, el authz\_groupfile y el authz\_user

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Creamos el fichero grupos dentro de /var/www/miseguridad.com**: sudo gedit .grupos



Y ahora, **creamos el usuario Carlos**:



**Modificamos el fichero virtualhosts**:



Y en dicho fichero, **añadimos la directiva AuthGroupFile** “/var/www/miseguridad.com/.grupos”

Imagen que contiene interior, tabla, foto

Descripción generada automáticamente

**Reiniciamos Apache2**: sudo service apache2 restart

Pero al viajar la información por HTTP, la información viaja en claro, no está cifrado el tráfico. Por tanto, pueden desencriptar nuestras contraseñas.

**Cifrado HTTPS:**

Nos permite autenticar el servidor con un certificado digital (que van firmados por autoridad certificadora) diciendo y confirmando que él es quien dice ser, en lugar de un atacante que está en medio.

La criptografía simétrica es un texto al cual se le aplica una contraseña para encriptar, y quien lo recibe, puede desencriptarlo con la misma contraseña.

La criptografía asimétrica (mecanismo que usa SSL) está absada en dos contraseñas: cada usuario que está involucrado tiene dos contraseñas: una pública y una privada. La contraseña pública la divulga a la gente. Y él se guarda en secreto su llave privada.

Cualquier mensaje que cifra uno con una contraseña, se puede descrifrar con la otra.

Yo, con mi contraseña privada, cifro un documento. De tal manera, que puedo hacer publico ese documento. Cualquier persona que coja la contraseña publica que yo divulgué, puede descifrar y confirmar que yo fui quien cifró ese documento.

Luego tiene la otra vertiente: cualquier persona que quiera mandarme un mensaje privado, usa mi contraseña publica para cifrarlo, me lo envía, y yo puedo descrifrarlo con mi clave privada.

**Vamos a crear un virtualhost que vaya con HTTPS:**

**Habilitamos el módulo de SSL de Apache**: sudo a2enmod ssl

**Comprobamos que el puerto 443 está abierto**: cat ports.conf

Texto

Descripción generada automáticamente

**Nos vamos a los sites-available**:

**Para habilitar el virtualhost que trabaja en el puerto 443**:



Sudo a2ensite default-ssl.conf

**Reiniciamos**:

Sudo service apache2 restart

**Y ahora si aparecen habilitados**:



Si entramos a <https://localhost>:

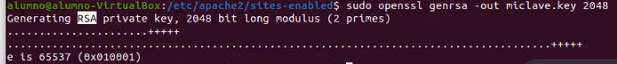
Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**También podemos crear un certificado nuestro propio** (aunque sigue siendo sin ser oficial, ya que como digo, solo reconocen los navegadores como seguros los emtiidos por entidades acreditadoras):

**Generar juego de claves de usuario y contraseña (clave publica y clave privada)**: Sudo openssl genrsa -out miclave.key 2048

Vemos que usa como algoritmo de cifrado el RSA:



**Creamos la petición**: Sudo openssl req -new -key miclave.key -out mipeticion.csr:

Texto

Descripción generada automáticamente

**Generamos el certificado**:

Sudo openssl x509 -req -days 365 -in mipeticion.csr -signkey



Movemos los ficheros a: /etc/ssl/private

Sudo mv miclave.key /etc/ssl/private

Sudo mv micertificado.crt y /etc/ssl/certs

Editamos el virtualhost para añadir este certificado: Sudo gedit default-ssl.conf

Y añadimos lo siguiente (y comentamos las que vienen por defecto):

Texto

Descripción generada automáticamente

Reiniciamos Apache2: sudo service apache2 restart

Entramos a <https://localhost>:

Y aparece ya nuestro certificado:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente