



# Practica 1

ILLIA NECHESA | PABLO OSPINO

Seguridad en Sistemas Informáticos | 4º Curso | 16-10-2019

# Un poco de criptografía

## Enunciado 1

Una vez leídas las indicaciones de la práctica, se nos indica que el enunciado2.oculto contiene un texto en base64. Vamos a proceder a descifrarlo con el siguiente comando:

➤ `openssl base64 -d -in enunciado2.base64 -out enunciado2.txt`

El siguiente paso, es aprender a cifrar y descifrar con cifradores simétricos como DES, AES, IDEA, BLOWFISH

## Enunciado 2

Vamos a proceder a descifrar el siguiente enunciado que esta cifrado en DES con la clave '0123456789012345':

➤ `openssl des-ecb -d -K 0123456789012345 -in enunciado3.oculto -out enunciado3.txt`

## Enunciado 3

Lo que vamos a hacer ahora, será aprender a calcular los hashes criptográficos más habituales, MD5, SHA-1, SHA-2... Para ello, vamos a descifrar "enunciado4.oculto" que se ha cifrado en AES256 en modo cipher block chaining, usando como passphrase MD5('Contraseña'). Para ello, los siguientes comandos:

- `openssl aes-256-cbc -d -a -in enunciado4.oculto -out enunciado4.txt`
- Enter aes-256-cbc decryption password:
  - Aquí es donde introducimos la contraseña en formato md5. La obtenemos mediante el comando → `openssl dgst -md5` → Introducimos la contraseña "Contraseña" → Presionamos Ctrl + D

## Enunciado 4

Ahora, deberíamos ser capaces de crear una clave privada y extraer de ella la clave publica, así como leer las claves privadas o publicas y extraer sus parámetros. La siguiente parte de las instrucciones se ha cifrado con un cifrador BLOWFISH y la clave usada se ha cifrado con la clave pública correspondiente a la clave privada 'unclaveprivada.pem'. Para ello, vamos a hacer uso de los siguientes comandos:

- `cat claveparael5.txt | base64 -d > claveparael5rawfile.txt`
- `openssl rsautl -decrypt -inkey unclaveprivada.pem -in claveparael5rawfile.txt -out claveparael5des.txt`
- `openssl bf-cbc -d -a -in enunciado5.oculto -out enunciado5.txt`  
enter bf-cbc decryption password: esto es la clave

## Enunciado 5

En esta parte, vamos a aprender a hacer certificados. Con una clave privada, podemos generar un certificado que contenga la clave publica y algunos datos mas identificando un servidor seguro. Vamos a aprender a generar una petición de certificado x509 y a firmarlo con clave privada.

Primero generamos una clave privada:

- `ssh-keygen -f key5 (passphrase ssi29)`

Seguido, generamos un certificado x509 a partir de la clave privada:

- `openssl req -new -x509 -key key5.pem -out cert5.pem`
  - Enter pass phrase for key5.pem:  
You are about to be asked to enter information that will be incorporated into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) []:ES  
State or Province Name (full name) []:Navarra  
Locality Name (eg, city) []:Pamplona  
Organization Name (eg, company) []:UpNa  
Organizational Unit Name (eg, section) []:SSI  
Common Name (eg, fully qualified host name) []:SSI  
Email Address []:ospino.p@gmail.com

Una vez hecho esto vamos a proceder a configurar el servidor apache en nuestro ordenador personal.

- `sudo apt-get update`
- `sudo apt-get install apache2`
- `sudo nano /etc/apache2/apache2.conf`
- Añadir línea "ServerName localhost:80" → Nombre del servidor
- `sudo nano /etc/apache2/sites-available/default-ssl.conf` # Deshabilitar el listado de directorios
- `systemctl restart apache2` | `service apache2 restart` → Podemos utilizar cualquiera de los dos

**Primer paso** → Activamos el modulo SSL y reiniciamos el servidor.

- `sudo a2enmod ssl`
- `service apache2 restart`

**Segundo paso** → Creamos un subdirectorio dentro de la carpeta de configuración de apache para colocar los certs, y creamos la llave y el certificado en una misma instrucción.

- `mkdir certs`
- `openssl req -new -newkey rsa:2048 -days 365 -nodes -x509 -keyout server.key -out server.crt`

**Tercer paso** → Configuramos el archivo default-ssl.conf que contiene la configuración de SSL predeterminada.

**Cuarto paso** → Habilitamos el SSL una vez configurado el host virtual, y reiniciamos apache con “`service apache2 restart`”

**Quinto paso** → Comprobamos la configuración

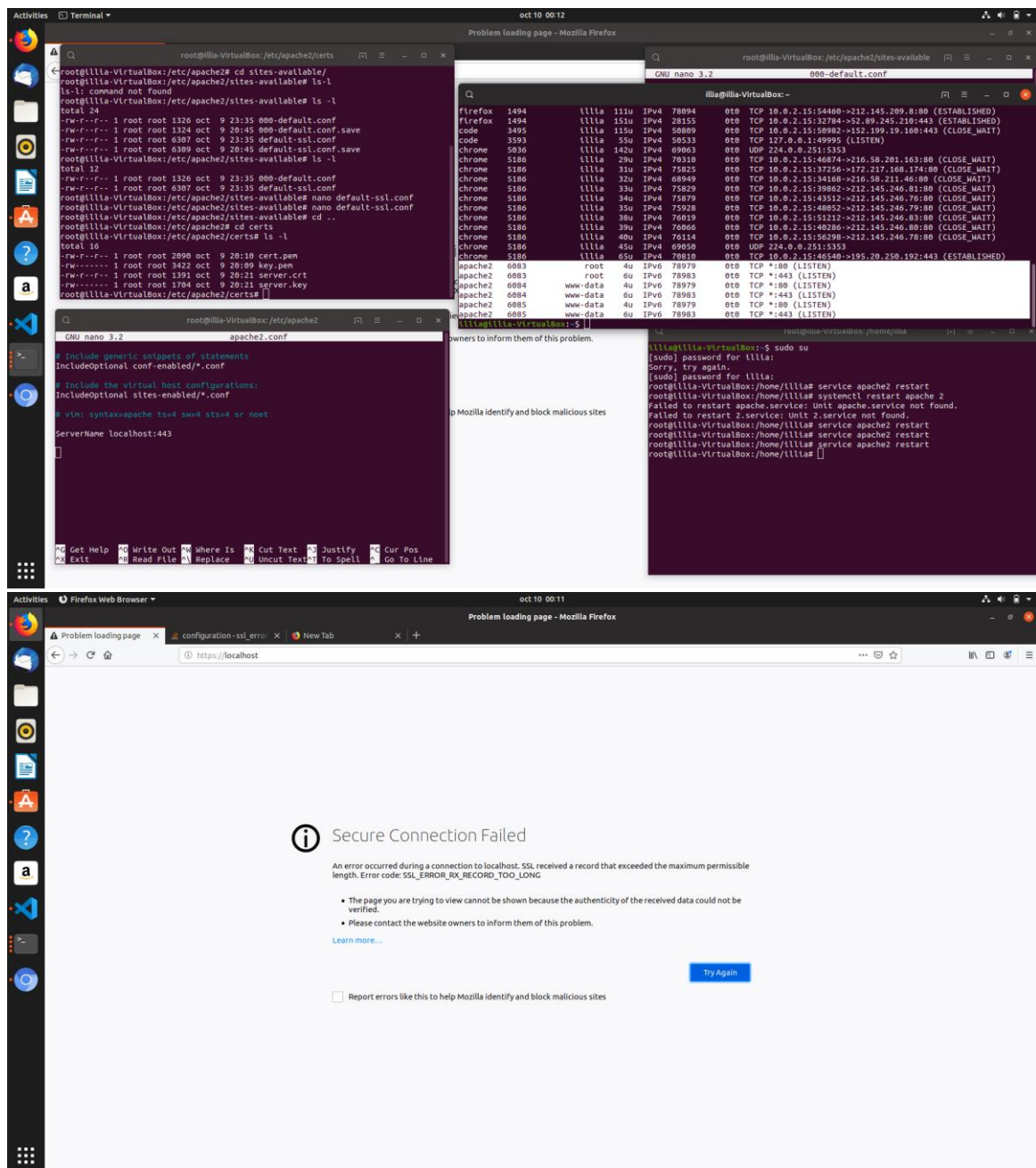
.....

Hemos instalado apache2, y lo hemos lanzado. Hemos creado en un comando tanto el certificado como la clave, que posteriormente los utilizo en el archivo de propiedades "default-ssl.conf". También hemos modificado el archivo 000-default.conf, ya que nos es imposible conectarnos al servidor. En el archivo apache2.conf hemos añadido también el servername. En los screenshots, se pueden ver todos los cambios realizados, y además hemos comprobado que el puerto 443 este escuchando y que está reservado para el apache2. Aquí abajo dejo los screenshots con la configuración :

The screenshots show the following configurations and actions:

- default-ssl.conf:** Configuration for the SSL engine, including the SSLCertificateFile and SSLCertificateKeyFile directives.
- 000-default.conf:** Configuration for the default virtual host, including the ServerName and DocumentRoot directives.
- apache2.conf:** Main configuration file for Apache2, including the IncludeOptional directive and the VirtualHost directive.
- Terminal Output:** Shows the output of the service restart command, indicating that the service is running and listening on port 443.





Comentar que hemos probado cambiar el ServerName por

- ➔ \*:80
- ➔ localhost:80
- ➔ \*:443
- ➔ localhost:443

Y con ninguno de los anteriores nos funciona correctamente.