TP TLS - OpenSSL - Certificados Digitales X.509

Fecha de Entrega: Chivilcoy 14/11/2022 - Luján 21/11/2022.

Objetivo: Conocer un caso de aplicación de varios tipos de cifrados. Reconocer mecanismos de intercambio de clave y familiarizarse con las tecnologías que permiten operar una Infraestructura de Clave Pública (PKI). Entender el concepto de certificado en el contexto de TLS / HTTPS y su funcionamiento. Conocer herramientas y mecanismos para generar y administrar certificados ITU-T X.509, los procedimientos de solicitud, firma e implementación. Categorías ISO: FCAP**S**.

Bibliografía

- STALLINGS, W. 2014. Capítulo 14: Key Management and Distribution en *Cryptography and Network Security - Principles and Practice (6th ed)*. Pearson Education Inc.
- STALLINGS, W. 2014. Capítulo 17: Transport-Level Security en *Cryptography and Network Security - Principles and Practice (6th ed)*. Pearson Education Inc.
- GORALSKI, W. 2017. Capítulo 27: "Securing Sockets with SSL" en *The Illustrated Network: How TCP/IP Works in a Modern Network (2nd ed)*. Morgan Kaufmann. https://www.sciencedirect.com/science/book/9780128110270
- DRISCOLL, M. 2018. The Illustrated TLS Connection - Every byte of a TLS connection explained and reproduced - https://tls.ulfheim.net/
- DRISCOLL, M. 2018. Server Certificate Detail - https://tls.ulfheim.net/certificate.html
- COOPER, D., et al. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280.

Experiencia de laboratorio

1. Acceda a los siguientes sitios: https://webmail.unlu.edu.ar/, https://gitlab.com/help, https://www.google.com.ar/. A través de las herramientas de desarrollador (en Chrome y Firefox) obtenga los certificados y analice la jerarquía de certificados (Certificate Hierarchy).

   a) ¿Que entidades emitieron tales certificados? ¿Cuál es el orden de jerarquía? ¿Hay alguna coincidencia en la jerarquía de los certificados de los sitios visitados?

   b) Para el certificado de la web de la universidad (*.unlu.edu.ar) detalle: el algoritmo de firma utilizado, el período de validez del certificado, el sujeto (*subject*), el emisor (*issuer*) y la **copia de la clave pública** del servidor.

   c) ¿Qué significa la sección "Nombre alternativo del sujeto del certificado"? ¿Para qué puede utilizarse? ¿Qué valores poseen los certificados de UNLu y de Google?

2. Ingrese al sitio web https://badssl.com/ donde se recopila una serie de situaciones que pueden suceder en el contexto de HTTPS y los certificados provistos. Con el acompañamiento docente, examine los diferentes sitios que están linkeados allí y determine los errores que subyacen a las fallas presentadas en el sitio.

3. Genere un certificado auto-firmado para un sitio web alojado en su propia dirección IP. Configure un servidor web Apache que lo utilice (ver anexo). Acceda mediante un navegador a su sitio web utilizando HTTPS. ¿Qué significa el mensaje de error que presenta el navegador?

4. En lugar de utilizar el certificado auto-firmado, solicite al equipo docente que procese su solicitud de firma de certificado (CSR). Instale el nuevo certificado en el servidor web y repita la consulta. ¿Qué cambió? ¿Es posible evitar que el navegador presente la advertencia?

5. Utilice el comando `openssl` para obtener el certificado desde su servidor web y almacenarlo en un archivo denominado *cert_servidor_laboratorio.txt*

   ```
   $ openssl s_client -connect IP_SERVIDOR:PUERTO
   ```

6. Realice una captura al momento de realizar una consulta a un servidor web en el laboratorio. Guarde esta captura bajo el nombre *cap-ejer-ssl.pcap*

Trabajo Práctico

1. Acceda a https://www2.mincyt.gob.ar/ y tome nota del error. ¿Por qué el navegador dice *no confiar* en el contenido de esa web? (ayuda: haga clic en "Avanzadas").
   Repita los pasos para el sitio web https://equifaxdigital.com.ar/ e indique también el motivo en este caso. ¿Qué validación no se cumple en este caso?

2. ¿Cuántas Autoridades de Certificación (CA) son reconocidas por su navegador web? ¿Qué problemas puede ocasionar la adición de nueva autoridad de certificación falsa? ¿Qué problemas puede ocasionar la eliminación de una o más autoridades de la lista?

3. ¿A que corresponden las extensiones de archivos "**.crt**", "**.key**" y "**.csr**" en el contexto de los certificados?

4. ¿Qué modificaciones debe realizar en un servidor web apache2 para proveer servicio de HTTPS? ¿Cómo se agrega, en este contexto, un certificado a un sitio web? ¿Por qué son necesarios tanto el certificado como la clave privada?

5. ¿En qué situación los certificados que son firmados por un tercero pueden aún considerarse no seguros para un navegador? ¿Cómo se puede lograr que un navegador confíe en el certificado para esta situación?

6. ¿En qué escenarios pueden resultar útiles los certificados autofirmados?

7. Analice el archivo *cert_servidor_laboratorio* ¿Qué información, que considere útil, puede recuperar de allí?

8. Realice un análisis de la captura *cap-ejer-ssl.pcap* y donde:

   a. Identifique las distintas etapas del protocolo TLS.
   b. Identifique opciones intercambiadas respecto a Cipher Suite y Extensiones soportadas.
   c. Identifique la información del los certificados y valídela contra lo generado en los pasos previos. Indique si el certificado es válido para el dominio/ip accedido y si aún es vigente.

Guia de lectura

1. ¿En qué consiste un certificado X.509?
2. ¿Por qué son necesarias las Autoridades de Certificación (CA)? ¿Cómo distribuyen sus claves públicas a todos los usuarios?
3. ¿Qué diferencia tecnológica existe entre un certificado autofirmado y un certificado firmado por una CA?
4. Nombre al menos cuatro causas por las que un certificado HTTPS puede ser inválido.
5. ¿Cómo establece TLS la conexión segura entre un navegador en el equipo de un cliente y un servidor web que implementa HTTPS?
6. Los certificados X.509, ¿se utilizan en otro contexto más allá de TLS?
7. ¿Por qué motivo se utilizan claves simétricas en una conexión TLS?

Anexo

Creación de un certificado autofirmado (sólo para proveer Confidencialidad)

1. Generar la clave privada (Private Key) del servidor, que será almacenada en el archivo `server.key` .

```
openssl genrsa -out server.key 4096
```

2. Generar la solicitud de firma de certificado (Certificate Signing Request), que será almacenada en el archivo `server.csr` . Completar los campos solicitados según el formulario de solicitud. Por ejemplo:

```
openssl req -new -sha256 -key server.key -out server.csr
-----
Country Name (2 letter code): AR
State or Province Name (full name): Buenos Aires
Locality Name (eg, city): Lujan
Organization Name (eg, company): Organización Example S.A.
Organizational Unit Name (eg, section): Gerencia de Sistemas
Common Name (eg, YOUR name): SU-DIRECCION-IP
Email Address: SU-DIRECCION-DE-CORREO

Please enter the following 'extra' attributes to be sent with your cert req.
A challenge password:
An optional company name:
```

3. Firmar la petición con la propia clave privada como sigue. En este caso se lo denomina “autofirmar”, puesto que estamos firmando la clave pública con la misma clave privada que le corresponde. En los sitios web que operan con TLS, quien firma la petición es una tercera entidad (una Autoridad de Certificación) en la que “todos” confían.

```
openssl x509 -req -days 365 -sha256 -in server.csr -signkey server.key
             -out server.crt
```

4. Para visualizar el contenido del certificado digital, ejecutar:

```
openssl x509 -text -in server.crt
```

Pasos a seguir para configurar e instalar los certificados en el servidor web

1. Instalar el servidor web Apache 2 o superior.

```
# apt-get install apache2
```

2. Activar los módulos `rewrite` y `ssl` , y el sitio `default-ssl` en Apache.

```
# a2enmod rewrite
# a2enmod ssl
# a2ensite default-ssl
```

3. Crear la ubicación `/etc/apache2/certificados` donde se almacenarán los certificados, copiarlos a la misma y asignar los permisos adecuados según la documentación disponible en `/usr/share/doc/apache2/README.Debian.gz` :

```
#
```

```
# mkdir /etc/apache2/certificados
# cd /etc/apache2/certificados
# mv origen/server.crt .
# mv origen/server.key .
# chown root.root server.crt server.key
# chmod 444 server.crt
# chmod 400 server.key
```

4. Editar el archivo `/etc/apache2/sites-enabled/default-ssl.conf` y reemplazar las líneas `SSLCertificateFile` y `SSLCertificateKeyFile` según sigue:

```
SSLCertificateFile  /etc/apache2/certificados/server.crt
SSLCertificateKeyFile   /etc/apache2/certificados/server.key
```

5. De ser necesario, habilitar el acceso al puerto 443 en el firewall del host y los nodos que correspondan.

6. Reiniciar el servidor apache (no alcanza sólo con recargar la configuración).

```
# service apache2 restart
```