«Lo esencial de OpenSSL: trabajando con certificados SSL, Llaves Privadas y CSR» por Mitchell Anicas

Introducción

OpenSSL es una herramienta versátil para ser usada en la línea de comandos y que puede utilizarse para una gran variedad de tareas relacionadas con la infraestructura de clave pública (PKI) y HTTPS (HTTP sobre TLS). Este tutorial, escrito en estilo de *hoja de trucos* y convertido en *guía*, proporciona una referencia rápida a los comandos de **OpenSSL** que son útiles en escenarios comunes y cotidianos. Esto incluye ejemplos de OpenSSL de generación de claves privadas, solicitudes de firma de certificados y conversión de formato de certificado. No cubre todos los usos de OpenSSL.

Cómo usar esta guía

- Si usted no está familiarizado con las Solicitudes de Firmado de Certificados (CSR) comience usted por leer la sección a continuación de esta sección.
- Aparte de la primera sección, esta guía se encuentra en un formato de hoja de trucos simple, con fragmentos de línea de comandos autocontenidos.
- Vaya usted a cualquier sección que sea relevante para la tarea que esté realizando (consejo: utilice la tabla de contenido al principio de esta entrada o bien use la función "**Buscar**" de su navegador web).
- Muchos de los comandos acá explicados son muy largos, buscando la claridad didáctica los hemos plasmado en varias lineas por medio del conector \ al final de cada una de ellas.
- *Nota del traductor*: para instalar el software OpenSSL en Debian y derivados aconsejamos utilizar la siguiente instrucción que actualiza los repositorios locales, instala OpenSSL en sí mismo además del comando **rand** y también crea un fichero necesario para OpenSSL.

```
sudo apt update && sudo apt install openssl rand -y && touch ~/.rnd
```

Acerca de las Solicitudes de Firmado de Certificados (CSR)

Si usted desea obtener un certificado SSL de una autoridad de certificación (CA), debe generar una Solicitud de Firma de Certificado (CSR). Una CSR consiste principalmente en la clave pública de un par de claves y alguna información adicional. Ambos componentes se insertan en el certificado cuando se firma.

Siempre que genere una CSR, se le pedirá que proporcione información sobre el certificado. Esta información se conoce como un Nombre Distinguido (DN). Un campo importante en el DN es el Nombre Común (CN), que debe ser exactamente el Nombre de Dominio Completamente

Cualificado (FQDN) del anfitrión con el que desea utilizar el certificado. También es posible omitir las indicaciones interactivas al crear una CSR pasando la información a través de la línea de comandos o desde un archivo, sin embargo las CA siempre exigirán este requisito.

Los otros campos en un DN proporcionan información adicional sobre su organización o empresa. Si está adquiriendo un certificado SSL de una autoridad de certificación, a menudo es necesario que estos campos adicionales, como "Organización", reflejen con precisión los detalles de su negocio o entidad.

He aquí un ejemplo de una CSR:

```
Country Name (2 letter code) [AU]:VE
 State or Province Name (full name) [Some-State]:Carabobo
 Locality Name (eg, city) []:Valencia
 Organization
 Name (eg, company) [Inter
net Widgits Pty Ltd]:Ejemplo de una Compañía Valenciana de Libros
 Organizational Unit Name (eg, section) []:División de Tecnología
 Common Name (e.g. server
 FQDN or YOUR name) []:ejemplovalencianadelibros.com.ve
 Email Address []:webmaster@ejemplovalencianadelibros.com.ve
```

Si usted no quiere contestar de manera interactiva la CSR, usted bien puede agregar toda esa información mediante el parámetro **-subj** para cualquiera de las solicitudes de comando que llamen a **OpenSSL**. Un ejemplo sería el siguiente, basado en la información anterior suministrada:

```
-subj "/C=VE/ST=Carabobo/L=Valencia/O=
Ejemplo de una Compañía Vale
nciana de Libros /CN=ejemplovalencianadelibros.com.ve"
```

Ahora que entiende las CSR, siéntase libre de ir a cualquier sección de esta guía que cubra sus necesidades sobre **OpenSSL**.

Creando las CSR

Esta sección cubre los comandos **OpenSSL** que están relacionados con la generación de las CSR (y sus claves privadas respectivas si no existen). Las CSR se pueden utilizar para solicitar certificados SSL a una autoridad de certificación.

Tenga en cuenta que puede agregar la información de las CSR de manera no interactiva con la parámetro **-subj**, mencionado en la sección anterior.

Creando una Llave Privada con una CSR

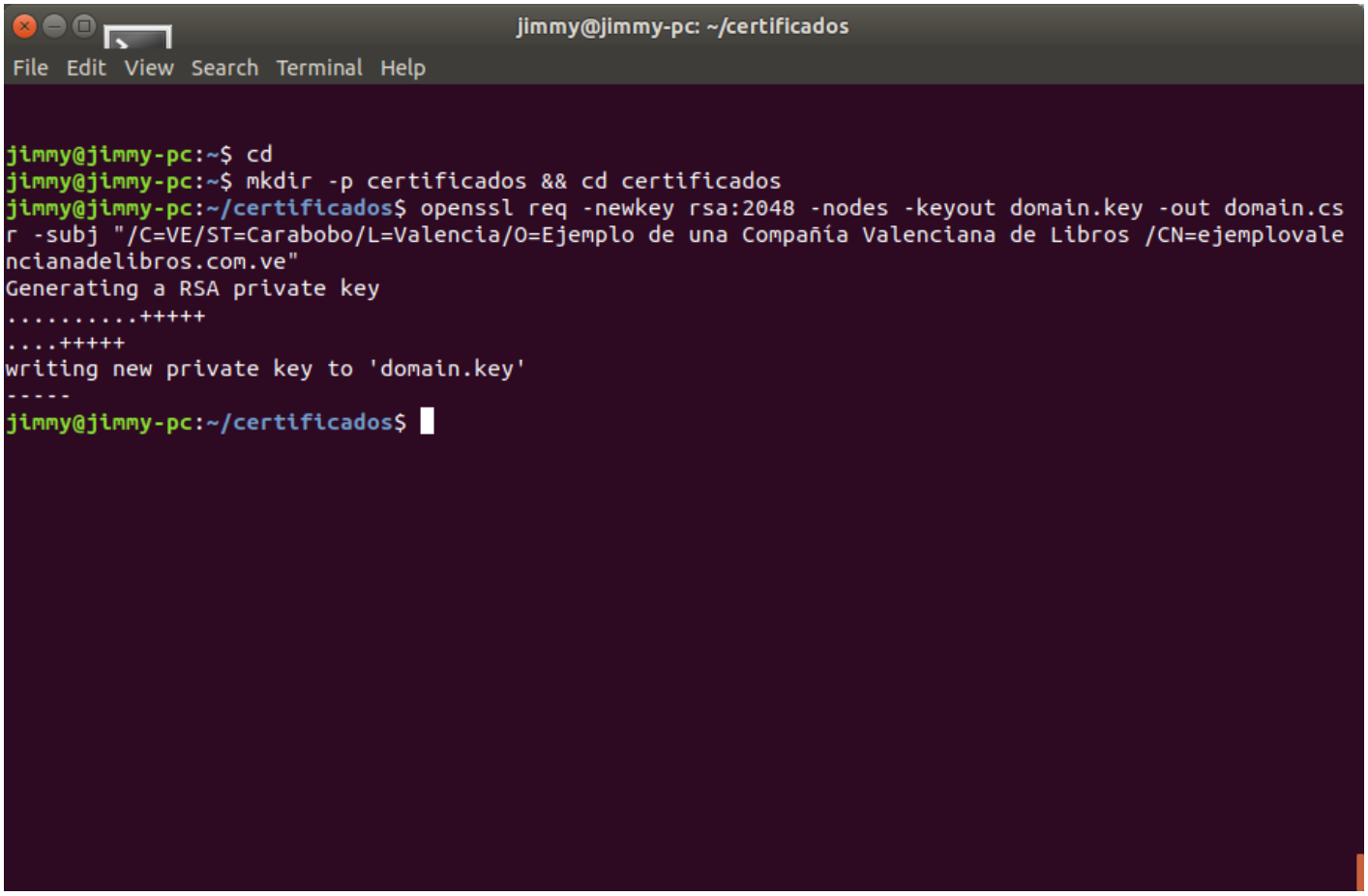
Utilice este método si desea utilizar HTTPS (HTTP over TLS) para cifrar las comunicaciones de su servidor web Apache o nginx y quiere usar una Autoridad de Certificación (CA) para expedir el certificado SSL. La CSR que resulte generada siguiendo estas instrucciones puede ser enviada a la CA para solicitar el otorgamiento del certificado SSL firmado digitalmente por la CA en cuestión. Si la CA admite cifrado SHA-2 entonces agregue el parámetro **-SHA256** para firmar localmente la CSR con cifrado SHA-2.

Este comando a continuación creará una Llave Privada (**dominio.key**) y una CSR (**dominio.csr**) desde cero:

```
openssl req \
           -newkey rsa:2048 -nodes -keyout dominio.key
           -out dominio.csr
```

Conteste las preguntas necesarias para completar el proceso de crear una CSR.

El parámetro **-newkey rsa:2048** especifica que la Llave debe tener una longitud de 2048 bits usando el algoritmo RSA. El parámetro **-nodes** especifica que la Llave Privada *no* debe estar encriptada con una *frase de contraseña*. El parámetro **-new**, la cual no está incluida aquí pero está implícita, señala que una CSR es generada en ese momento.



Creando una Llave Privada con una CSR

Creando una CSR a partir de una Llave Privada existente

Use este método si usted ya tiene una Llave Privada que desea usar para solicitar un certificado de una CA.

Este comando creará una nueva CSR (llamada **dominio.csr**) basado en una Llave Privada existente (llamada **dominio.key**):

```
openssl req \
          -key dominio.key \
          -new -out dominio.csr
```

Responda las preguntas necesarias para completar el proceso de crear una CSR.

El parámetro **-key** especifica la Llave Privada existente (**dominio.key**) que será usada para generar la nueva CSR. El parámetro **-new** señala que una nueva CSR será generada.

Creando una CSR a partir de un certificado y Llave Privada, ambos existentes.

Emplee este método si desea renovar un certificado existente pero usted o su CA, por cualquier motivo, no tienen el CSR original. Básicamente ahorra dificultades de reingresar la información del CSR, la cual es extraída del certificado existente.

Esta orden crea un nuevo CSR (llamado **dominio.csr**) basado en un certificado existente (llamado **dominio.crt** -el cual se explicará cómo crear uno en la próxima sección *N. del T.)* y una Llave Privada (llamada **dominio.key**):

```
openssl x509\
       -in dominio.crt \
       -signkey dominio.key \
       -x506toreq -out dominio.csr
```

El parámetro -x509toreq especifica que usted está usando un certificado tipo X509 para realizar un CSR.

Creando Certificados SSL

Si usted quisiera utilizar un Certificado SSL para ofrecer un servicio web seguro pero no necesita un certificado frimado digitalmente por una CA, una manera válida (y sin costo) es firmar usted mismo sus propios certificados.

Un tipo común de certificado que puede otorgarse usted mismo es un *certificado digital*

autofirmado. Un certificado autofirmado es aquel que esta firmado con su propia Llave Privada. Los Certificados autofirmados bien pueden ser aprovechados para cifrar datos exactamente de la misma manera de los certificados firmados por una CA, *pero sus usuarios les será mostrada una advertencia indicando que el certificado no está autenticado por la computadora de ellos o ellas y/o sus navegadores web.* Por lo tanto los certificados autofirmados deberían ser usados unicamente caundo no necesite probar la identidad del servicio web que usted ofrezca (como es el caso de los entornos de *no están en producción* -ambientes de desarrollo- o en servidores *no abiertos al público*).

Esta sección se ocupa de los comandos de OpenSSL relacionados con la generación de certificados autofirmados.

Creando un certificado autofirmado

Utilice este método si desea utilizar HTTPS (HTTP over TLS) para cifrar las comunicaciones de su servidor web nginx o Apache y usted **no** quiere usar un certificado firmado por una CA.

Estas instrucciones crearán una Llave Privada (llamada **dominio.key**) con una longitud de 2048 bits y un certificado autofirmado (llamado **dominio.crt**) desde cero:

```
openssl req \
       -newkey rsa:2018 -nodes -keyout dominio.key \
       -x509 -days 365 -out domain.crt
```

Complete las preguntas necesarias que haga el programa para culminar el proceso.

El parámetro **-x509** indica a la opción **req** a crear un certificado autofirmado. El parámetro **-days 365** especifica que el certificado será válido por 365 días. Una CSR temporal es creada para recabar información asociada al certificado.

Creando un certificado autofirmado de una Llave Privada existente

Si usted ya tiene una Llave Privada, puede emplear esta manera si quiere crear un certificado

autofirmado a partir de dicha llave.

Las siguientes líneas crean un certificado autofrimado (**dominio.crt**) a partir de una Llave Privada existente (llamada **dominio.key**):

```
openssl req \
       -key dominio.key \
       -new \
       -x509 -days 365 -out dominio.crt
```

Conteste las preguntas necesarias para el proceso.

El parámetro **-x509** indica a la opción **req** a crear un certificado autofirmado. El parámetro **-days 365** especifica que el certificado será válido por 365 días. El parámetro **-new** habilita para que pregunte la información necesaria para la CSR.

Creando un certificado autofirmado a partis de una Llave Privada y una CSR, ambos existentes

Este método le permitirá, teniendo una Llave Privada y una CSR la generación de un certificado autofirmado a partir de ambos datos.

Este coomando creará un certificado autofirmado (llamado **dominio.crt**) desde una existente Llave Privada (llamada **dominio.key**) y una CSR existente (llamada **dominio.csr**):

```
openssl x509 \
       -signkey dominio.key \
       -in dominio.csr \
       -req -days 365 -out dominio3.crt
```

El parámetro **-days 365** especifica que el certificado será válido por 365 días.

Visualización de certificados

Los certificados y las CSR están codificados en un formato llamado PEM, el cual no es legible para los seres humanos.

Esta sección trata acerca de los parámetros para que OpenSSL extraiga los campos contenidos en los ficheros PEM.

Visualizar entradas CSR almacenadas

Este código permite visualizar y verificar los contenidos de una CSR en texto plano (en este ejemplo almacenada en un fichero llamado **dominio.csr**):

```
openssl req -text -noout -verify -in dominio.csr
```

Visualizando entradas de certificado

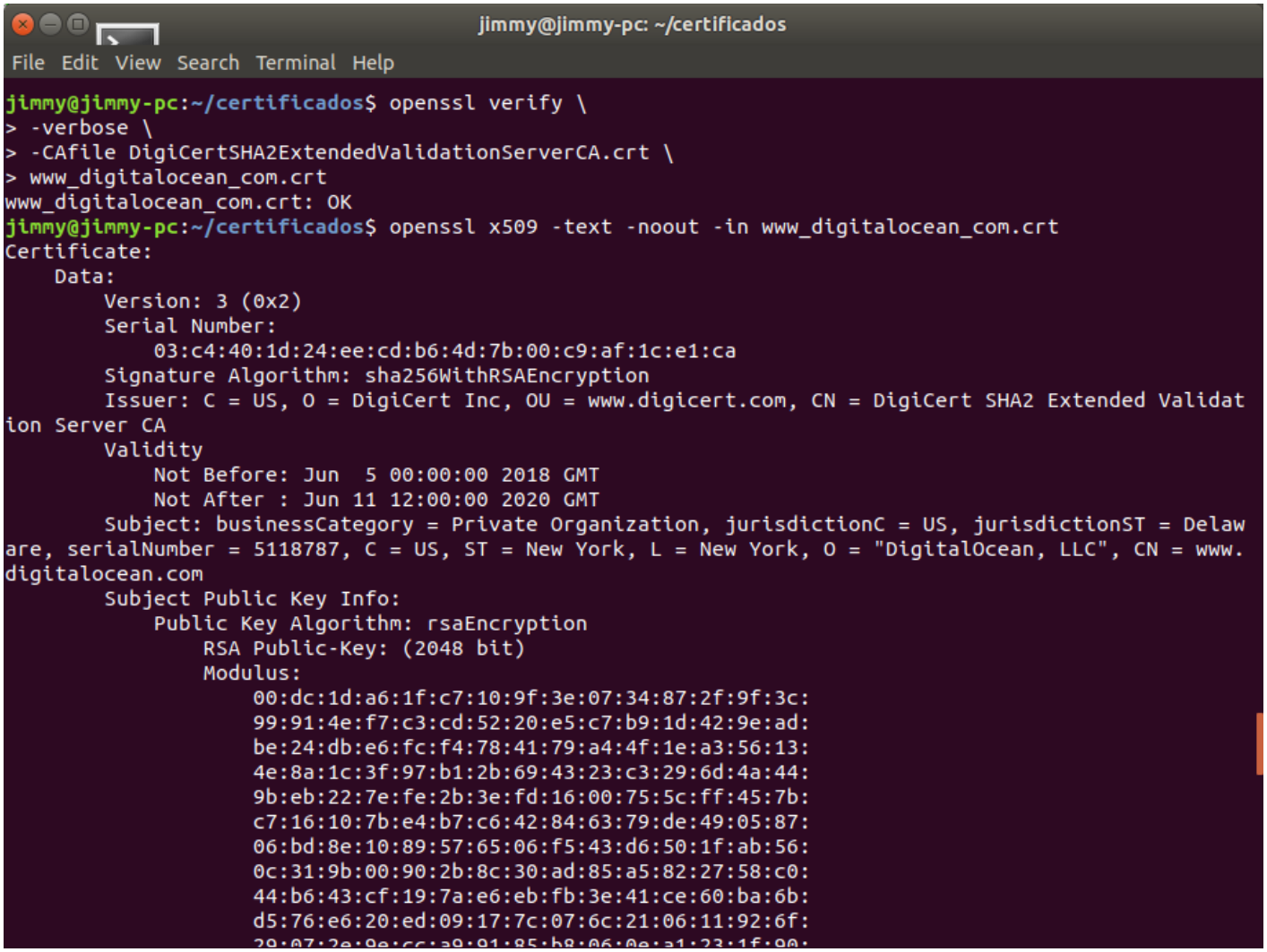
Facilita la exposición del contenido de un certificado en texto plano (en este caso desde un fichero llamado **dominio.crt**):

```
openssl x509 -text -noout -in dominio.crt
```

Verificando un certificado que fue firmado por una CA

Puede usar este comando para verificar un certificado (en este caso llamado **dominio.crt**) el cual fue firmado con el certicifacado de un CA específica (en este caso llamado **ca.crt**)

```
openssl verify -verbose -CAfile ca.crt domain.crt
```



openssl verify -verbose (DigiCert avala certificado de DigitalOcean)

Fuentes adicionales consultadas

Artículos en idioma castellano

Artículos en idioma inglés

- Mensaje "**Can't load ./.rnd into RNG**"en repositorio de OpenSSL en GitHub (reporte #7754); solución opcional: **sudo apt install rand && mkdir -p ~/.rnd** . Véase también reporte #898470 a Debian: apentemente la solución sería **touch ~/.rnd** , es decir, crear un fichero vacio como futuro depósito de cálculos.
- Parámetro **RANDFILE** en archivo de configuración de OpenSSL (en Ubuntu 18 localizado en */etc/ssl/openssl.cnf*): «How to Generate a Self-Signed SSL Certificate on Linux»