TUTORIAL

Cómo instalar y configurar una entidad de certificación (CA) en Ubuntu 20.04

Ubuntu Security VPN Ubuntu 20.04

By Jamon Camisso

Published on May 21, 2020 15.6k

Introducción

Las entidades de certificación (CA) son responsables de emitir certificados digitales para verificar identidades en Internet. Las CA públicas son una opción popular para verificar la identidad de sitios web y otros servicios que se proporcionan al público en general, y las CA privadas suelen usarse para grupos cerrados y servicios privados.

Compilar una entidad de certificación privada le permitirá configurar, probar y ejecutar programas que requieren conexiones cifradas entre un cliente y un servidor. Con una CA privada, puede emitir certificados para usuarios, servidores o programas y servicios individuales dentro de su infraestructura.

Algunos ejemplos de programas de Linux que utilizan su propia CA privada son OpenVPN y Puppet. También puede configurar su servidor web para que use certificados emitidos por una CA privada a fin de que los entornos de desarrollo y ensayo se adapten a los servidores de producción que utilizan TLS para cifrar conexiones.

En esta guía, aprenderá a instalar una entidad de certificación privada en un servidor de Ubuntu 20.04 y a generar y firmar un certificado de prueba con su CA nueva. También aprenderá a importar el certificado público del servidor de CA al almacén de certificados de su sistema operativo para poder verificar la cadena de confianza entre los usuarios o servidores remotos y la CA. Por último, aprenderá a revocar certificados y

SCROLL TO TOP

lista de revocación de certificados para asegurarse de que solo usuarios y sistemas autorizados puedan usar los servicios que se basan en su CA.

Requisitos previos

Para completar este tutorial, necesitará acceso a un servidor Ubuntu 20.04 a fin de alojar su servidor de CA. Antes de comenzar a seguir los pasos de esta guía, deberá configurar un non-**root** user con privilegios `sudo`. Puede seguir nuestra guía de configuración inicial de servidores para Ubuntu 20.04 para configurar un usuario con los permisos adecuados. A través del tutorial del enlace también se podrá configurar un **firewall**, que para esta guía se supone que está instalado.

En este tutorial, emplearemos el término **servidor de CA** para hacer referencia a este servidor.

Asegúrese de que el sistema del servidor de CA sea independiente. Se usará únicamente para importar, firmar y revocar solicitudes de certificados. No debe ejecutar ningún otro servicio y lo ideal es que se desconecte o desactive por completo cuando usted no esté trabajando activamente con su CA.

> **Nota:** La última sección de este tutorial es opcional si desea aprender a firmar y revocar certificados. Si decide completar esos pasos de prueba, necesitará un segundo servidor de Ubuntu 20.04 o, de forma alternativa, puede usar su propia computadora local de Linux con Debian o Ubuntu, o distribuciones derivadas de cualquiera de ellos.

Paso 1: Instalar Easy-RSA

La primera tarea de este tutorial es instalar el conjunto de secuencias de comandos `easy-rsa` en su servidor de CA. `easy-rsa` es una herramienta de gestión de entidades de certificación que utilizará para generar una clave privada y un certificado root público que, luego, usará para firmar las solicitudes de los clientes y servidores que se basarán en su CA.

Inicie sesión en su servidor de CA con el non-root sudo user que creó en los pasos de configuración iniciales y ejecute lo siguiente:

```
$ sudo apt update
$ sudo apt install easy-rsa
```

Se le solicitará descargar e instalar el paquete. Presione `y` para confirmar que desea instalarlo.

En este punto, tiene todo lo que necesita para usar Easy-RSA. En el siguiente paso, creará una infraestructura de clave pública y, luego, empezará a crear su entidad de certificación.

Paso 2: Preparar un directorio para la infraestructura de clave pública

Ahora que instaló `easy-rsa`, es el momento de crear una infraestructura de clave pública (PKI) de esqueleto en el servidor de CA. Verifique que siga conectado con su non-root user y cree un directorio `easy-rsa`. Asegúrese de **no utilizar sudo** para ejecutar ninguno de los siguientes comandos, dado que su usuario normal debe administrar la CA e interactuar con ella sin privilegios elevados.

```
$ mkdir ~/easy-rsa
```

Con esto se creará un directorio nuevo llamado `easy-rsa` en su carpeta de inicio. Usaremos este directorio para crear enlaces simbólicos que apunten a los archivos del paquete `easy-rsa` que instalamos en el paso anterior. Estos archivos se encuentran en la carpeta `/usr/share/easy-rsa` en el servidor de CA.

Cree los enlaces simbólicos con el comando `ln`:

```
$ ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

Nota: Aunque en otras guías se le indique copiar los archivos del paquete `easy-rsa` a su directorio de la PKI, en este tutorial, usaremos enlaces simbólicos. Como resultado, toda actualización del paquete `easy-rsa` se reflejará automáticamente en las secuencias de comandos de su PKI.

Para restringir el acceso a su nuevo directorio de la PKI, asegúrese de que solo el propietario pueda acceder a él usando el comando `chmod`:

```
$ chmod 700 /home/sammy/easy-rsa
```

Por último, inicie la PKI dentro del directorio `easy-rsa`:

```
$ cd ~/easy-rsa
$ ./easyrsa init-pki
```

```
Output
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/sammy/easy-rsa/pki
```

Después de completar esta sección, tendrá un directorio con todos los archivos necesarios para crear una entidad de certificación. En la siguiente sección, creará la clave privada y el certificado público para su CA.

Paso 3: Crear una entidad de certificación

Para poder crear la clave privada y el certificado de su CA, debe crear y completar un archivo llamado `vars` con algunos valores predeterminados. Primero, usará `cd` para ingresar al directorio `easy-rsa` y, luego, creará y editará el archivo `vars` con `nano` o el editor de texto que prefiera:

```
$ cd ~/easy-rsa
$ nano vars
```

Una vez que se abra el archivo, pegue las siguientes líneas y sustituya cada valor resaltado por la información de su propia organización. Lo importante aquí es asegurarse de no dejar ninguno de los valores en blanco:

```
~/easy-rsa/vars
set_var EASYRSA_REQ_COUNTRY    "US"
set_var EASYRSA_REQ_PROVINCE   "NewYork"
set_var EASYRSA_REQ_CITY       "New York City"
set_var EASYRSA_REQ_ORG        "DigitalOcean"
set_var EASYRSA_REQ_EMAIL      "admin@example.com"
set_var EASYRSA_REQ_OU         "Community"
set_var EASYRSA_ALGO           "ec"
set_var EASYRSA_DIGEST         "sha512"
```

Cuando termine, guarde y cierre el archivo. Si utiliza `nano`, puede hacerlo pulsando `CTRL+X`, `Y` y `ENTER` para confirmar. Con esto, estará listo para crear su CA.

Para crear el certificado root público y el par de claves privadas para su entidad de certificación, vuelva a ejecutar el comando `./easy-rsa`, aunque esta vez con la opción `build-ca`:

```
$ ./easyrsa build-ca
```

En el resultado, verá algunas líneas sobre la versión de OpenSSL y se le solicitará ingresar una frase de contraseña para su par de claves. Asegúrese de elegir una frase de contraseña segura y anótela en un lugar resguardado. Deberá ingresar la frase de contraseña siempre que deba interactuar con su CA, por ejemplo, para firmar o revocar un certificado.

También se le solicitará confirmar el nombre común (CN) de su CA. El nombre común es el que se usa para hacer referencia a esta máquina en el contexto de la entidad de certificación. Puede ingresar cualquier secuencia de caracteres para el nombre común de la CA; sin embargo, para hacerlo más simple presione ENTER para aceptar el nombre predeterminado.

```
Output
. . .
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
. . .
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/sammy/easy-rsa/pki/ca.crt
```

Nota: Si no desea que se le solicite una contraseña cada vez que interactúe con su CA, puede ejecutar el comando `build-ca` con la opción `nopass`, de la siguiente forma:

```
$ ./easyrsa build-ca nopass
```

Ahora, tiene dos archivos importantes, `~/easy-rsa/pki/ca.crt` y `~/easy-rsa/pki/private/ca.key`, que conforman los componentes públicos y privados de una entidad de certificación.

- `ca.crt` es el archivo del certificado público de la CA. Los usuarios, los servidores y los clientes utilizarán este certificado para verificar que sean parte de la misma red de confianza. Todos los usuarios y los servidores que usen su CA deberán tener una copia de este archivo. Todas las partes se basarán en el certificado público para asegurarse de que nadie suplante un sistema y realice un ataque con intermediario.
- `ca.key` es la clave privada que usa la CA para firmar certificados para servidores y clientes. Si un atacante obtiene acceso a su CA y, a la vez, a su archivo `ca.key`, deberá destruir la CA. Esta es la razón por la cual su archivo `ca.key` deberá estar **únicamente** en su máquina de CA. A su vez, lo ideal sería que su máquina de CA estuviera desconectada cuando no firme solicitudes de certificados como medida de seguridad adicional.

Con esto, estableció su CA y esta se encuentra lista para emplearse en la firma de solicitudes de certificados y revocar certificados.

Paso 4: Distribuir el certificado público de su entidad de certificación

Ahora, su CA está configurada y lista para funcionar como root de confianza para cualquier sistema que desee que la use. Puede agregar el certificado de la CA a sus servidores de OpenVPN, web y de correo, entre otros. Cualquier usuario o servidor que necesite verificar la identidad de otro usuario o servidor de su red debe contar con una copia del archivo `ca.crt` importada en el almacén de certificados de su sistema operativo.

Para importar el certificado público de la CA a un segundo sistema de Linux, como otro servidor o una computadora local, primero debe obtener una copia del archivo `ca.crt` de su servidor de CA. Puede usar el comando `cat` para ver el resultado en una terminal y, luego, copiarlo y pegarlo en un archivo en la segunda computadora en la que se importe el certificado. También puede usar herramientas como `scp` y `rsync` para transferir el archivo entre sistemas. Sin embargo, usaremos el método de copiar y pegar con `nano` en este paso, ya que funciona en todos los sistemas.

Como non-root user en el servidor de CA, ejecute el siguiente comando:

```
$ cat ~/easy-rsa/pki/ca.crt
```

El resultado en su terminal será similar al siguiente:

```
Output
-----BEGIN CERTIFICATE-----
MIIDSzCCAjOgAwIBAgIUcR9Crsv3FBEujrPZnZnU4nSb5TMwDQYJKoZIhvcNAQEL
BQAwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwHhcNMjAwMzE4MDMxNjI2WhcNMzAw
. . .
. . .
-----END CERTIFICATE-----
```

Copie todo, incluso las líneas `-----BEGIN CERTIFICATE-----` y `-----END CERTIFICATE-----` y los guiones.

En su segundo sistema de Linux, use `nano` o el editor de texto que prefiera para abrir un archivo llamado `/tmp/ca.crt`:

```
$ nano /tmp/ca.crt
```

Pegue lo que acaba de copiar del servidor de CA en el editor. Cuando termine, guarde y cierre el archivo. Si utiliza `nano`, puede hacerlo pulsando `CTRL+X` y, luego, `Y` y `ENTER` para confirmar.

Ahora que tiene una copia del archivo `ca.crt` en su segundo sistema de Linux, es momento de importar el certificado al almacén de certificados de su sistema operativo.

En los sistemas basados en Ubuntu y Debian, ejecute los siguientes comandos como non-root user para importar el certificado:

Ubuntu and Debian derived distributions

```
$ sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/
$ sudo update-ca-certificates
```

Para importar el certificado del servidor de CA en un sistema basado en CentOS, Fedora o RedHat, copie el contenido del archivo y péguelo en el sistema, como en e

en un archivo denominado `/tmp/ca.crt`. A continuación, copie el certificado en `/etc/pki/ca-trust/source/anchors/` y luego ejecute el comando `update-ca-trust`.

CentOS, Fedora, RedHat distributions

```
$ sudo cp /tmp/ca.crt /etc/pki/ca-trust/source/anchors/
$ sudo update-ca-trust
```

Ahora, su segundo sistema de Linux confiará en cualquier certificado firmado por el servidor de CA.

Nota: Si utiliza su CA con servidores web y usa Firefox como navegador, deberá importar el certificado público `ca.crt` directamente a Firefox. Firefox no usa el almacén de certificados del sistema operativo local. Para obtener información sobre cómo agregar el certificado de su CA a Firefox, consulte el artículo de asistencia de Mozilla sobre configuración de entidades de certificación (CA) en Firefox.

Si usa su CA para la integración con un entorno de Windows o computadoras de escritorio, consulte la documentación sobre cómo usar `certutil.exe` para instalar un certificado de CA.

Si completa este tutorial como requisito previo para otro o está familiarizado con la forma de firmar y revocar certificados, puede detenerse aquí. Si desea obtener más información sobre cómo firmar y revocar certificados, en la siguiente sección opcional, se explicará cada proceso en detalle.

(Opcional): Crear solicitudes de firma de certificados y revocar certificados

Las siguientes secciones del tutorial son opcionales. Si completó todos los pasos anteriores, dispondrá de una entidad de certificación completamente configurada y funcional que podrá utilizar como requisito previo para otros tutoriales. Puede importar el archivo `ca.crt` de su CA y verificar los certificados de su red firmados por su CA.

Si desea practicar y obtener más información sobre cómo firmar solicitudes de certificados y cómo revocarlos, en estas secciones opcionales se explicará cómo funcionan estos dos procesos.

(Opcional): Crear y firmar una solicitud de certificado de prueba

Ahora que tiene una CA lista para usar, puede practicar generar una clave privada y una solicitud de certificado para familiarizarse con el proceso de firma y distribución.

Las solicitudes de firma de certificados (CSR) constan de tres partes: una clave pública, información de identificación sobre el sistema solicitante y una firma de la solicitud misma, que se crea utilizando la clave privada de la parte solicitante. La clave privada se mantendrá en secreto y se usará para cifrar información que luego podrá descifrar quien tenga el certificado público firmado.

Los siguientes pasos se ejecutarán en su segundo sistema de Debian o Ubuntu, o en una distribución derivada de cualquiera de ellos. Puede ser otro servidor remoto o una máquina local de Linux, como una computadora portátil o de escritorio. Debido a que `easy-rsa` no está disponible por defecto en todos los sistemas, usaremos la herramienta `openssl` para crear una clave privada y un certificado de prueba.

`openssl` suele instalarse por defecto en la mayoría de las distribuciones de Linux. Sin embargo, para estar seguro, ejecute lo siguiente en su sistema:

```
$ sudo apt update
$ sudo apt install openssl
```

Cuando se le solicite instalar `openssl` ingrese `y` para proceder con los pasos de la instalación. Ahora, está listo para crear una CSR de prueba con `openssl`.

El primer paso para crear una CSR es generar una clave privada. Para crear una clave privada usando `openssl`, cree un directorio `practice-csr` y luego genere una clave dentro de este. En vez de crear un certificado para identificar usuarios u otras CA, crearemos esta solicitud para un servidor ficticio llamado `sammy-server`.

```
$ mkdir ~/practice-csr
$ cd ~/practice-csr
$ openssl genrsa -out sammy-server.key
```

```
Output
Generating RSA private key, 2048 bit long modulus (2 primes)
. . .
```

SCROLL TO TOP

```
. . .
e is 65537 (0x010001)
```

Ahora que tiene una clave privada, puede crear una CSR correspondiente y volver a usar la utilidad `openssl`. Se le solicitará completar varios campos, como Country, State y City. Puede ingresar `.` si desea dejar un campo en blanco, pero tenga en cuenta que, si se tratara de una CSR real, sería mejor usar los valores correctos de su ubicación y su organización:

```
$ openssl req -new -key sammy-server.key -out sammy-server.req
```

```
Output
. . .
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) [Default City]:New York City
Organization Name (eg, company) [Default Company Ltd]:DigitalOcean
Organizational Unit Name (eg, section) []:Community
Common Name (eg, your name or your server's hostname) []:sammy-server
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Si desea agregar esos valores automáticamente como parte de la invocación de `openssl` en lugar de hacerlo por medio de la solicitud interactiva, puede pasar el argumento `-subj` a OpenSSL. Asegúrese de editar los valores resaltados para que coincidan con la ubicación, la organización y el nombre del servidor de prueba:

```
$ openssl req -new -key sammy-server.key -out server.req -subj \
$ /C=US/ST=New\ York/L=New\ York\ City/O=DigitalOcean/OU=Community/CN=sammy-server
```

Para verificar el contenido de una CSR, puede leer el archivo de la solicitud con `openssl` y examinar los campos que contiene:

```
$ openssl req -in sammy-server.req -noout -subject
```

```
Output
subject=C = US, ST = New York, L = New York City, O = DigitalOcean, OU = Community, CN = sammy-server
```

Una vez que esté conforme con el asunto de la solicitud de certificado de prueba, copie el archivo `sammy-server.req` a su servidor de CA utilizando `scp`:

```
$ scp sammy-server.req sammy@your_ca_server_ip:/tmp/sammy-server.req
```

En este paso, generó una solicitud de firma de certificado para un servidor ficticio llamado `sammy-server`. En una situación real, la solicitud podría provenir, por ejemplo, de un servidor web de un entorno de ensayo o desarrollo que requiera un certificado TLS para pruebas o de un servidor de OpenVPN que solicite un certificado para que los usuarios puedan conectarse a una VPN. En el siguiente paso, procederemos a firmar la solicitud de firma de certificado usando la clave privada del servidor de CA.

(Opcional): Firmar una CSR

En el paso anterior, creó una solicitud de certificado y una clave de prueba para un servidor ficticio. La copió al directorio `/tmp` de su servidor de CA, simulando el proceso que seguiría si clientes o servidores reales le enviaran solicitudes de firma de certificados.

Continuando con la situación ficticia, ahora, el servidor de CA debe importar el certificado de prueba y firmarlo. Una vez que la CA valida una solicitud de certificado y la vuelve a enviar a un servidor, los clientes que confían en la entidad de certificación también podrán confiar en el nuevo certificado emitido.

Debido a que operaremos dentro de la PKI de la CA, donde está disponible la utilidad `easy-rsa`, utilizaremos esta utilidad `easy-rsa` en los pasos de firma para facilitar la tarea, en vez de usar `openssl` directamente como hicimos en el ejemplo anterior.

El primer paso para firmar la CSR ficticia es importar la solicitud de certificado usando la secuencia de comandos `easy-rsa`:


```
$ cd ~/easy-rsa
```

```
$ ./easyrsa import-req /tmp/sammy-server.req sammy-server
```

```
Output
. . .
The request has been successfully imported with a short name of: sammy-server
You may now use this name to perform signing operations on this request.
```

Ahora, puede firmar la solicitud ejecutando la secuencia de comandos `easyrsa` con la opción `sign-req`, seguida del tipo de solicitud y el nombre común incluido en la CSR. El tipo de solicitud puede ser `client`, `server` o `ca`. Debido a que estamos practicando con un certificado para un servidor ficticio, asegúrese de utilizar el tipo de solicitud `server`:

```
$ ./easyrsa sign-req server sammy-server
```

En el resultado, se le solicitará verificar que la solicitud provenga de una fuente de confianza. Escriba `yes` y luego presione `ENTER` para confirmarlo:

```
Output
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 3650 days:

subject=
    commonName                = sammy-server


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
. . .
Certificate created at: /home/sammy/easy-rsa/pki/issued/sammy-server.crt
```

Si cifró su clave de la CA, se le solicitará ingresar la contraseña en este punto.

Al completar esos pasos, firmó la CSR de `sammy-server.req` usando la clave servidor de CA en `/home/sammy/easy-rsa/pki/private/ca.key`. El archivo `sammy-server.crt`

resultante contiene la clave de cifrado pública del servidor de prueba, así como una nueva firma del servidor de CA. El objetivo de la firma es indicar a quienes confían en la CA que también pueden confiar en el certificado de `sammy-server`.

Si esta solicitud fuera para un servidor real, como un servidor web o VPN, el último paso en el servidor de CA sería distribuir los nuevos archivos `sammy-server.crt` y `ca.crt` del servidor de CA al servidor remoto que realizó la solicitud de firma de certificado:

```
$ scp pki/issued/sammy-server.crt sammy@your_server_ip:/tmp
$ scp pki/ca.crt sammy@your_server_ip:/tmp
```

En este punto, podría utilizar el certificado emitido con, por ejemplo, un servidor web, una VPN, una herramienta de administración de configuración o un sistema de base de datos, o para la autenticación de clientes.

(Opcional): Revocar un certificado

De tanto en tanto, es posible que deba revocar un certificado para evitar que un usuario o un servidor lo use, por ejemplo, si roban la computadora portátil de alguien, si se compromete un servidor web o si un empleado o contratista deja de trabajar con su organización.

Estos son los pasos del proceso general para revocar un certificado:

1. Revoque el certificado con el comando `./easyrsa revoke client-name`.
2. Genere una nueva CRL con el comando `./easyrsa gen-crl`.
3. Transfiera el archivo `crl.pem` actualizado al servidor o los servidores que se basan en su CA y, en esos sistemas, cópielo al directorio o los directorios necesarios de los programas que hagan referencia a él.
4. Reinicie todos los servicios que utilicen su CA y el archivo CRL.

Puede utilizar este proceso para revocar cualquier certificado que haya emitido anteriormente en cualquier momento. Analizaremos cada paso en detalle en las siguientes secciones, comenzando por el comando `revoke`.

Revocar un certificado

Para revocar un certificado, diríjase al directorio `easy-rsa` en su servidor de CA:

```
$ cd ~/easy-rsa
```

Luego, ejecute la secuencia de comandos `easyrsa` con la opción `revoke` seguida del nombre del cliente que desee revocar: Siguiendo el ejemplo práctico anterior, el nombre común del certificado es `sammy-server`:

```
$ ./easyrsa revoke sammy-server
```

Con esto, se solicitará que confirme el rechazo ingresando `yes`:

```
Output
Please confirm you wish to revoke the certificate with the following subject:

subject=
    commonName                = sammy-server


Type the word 'yes' to continue, or any other input to abort.
  Continue with revocation: yes
. . .
Revoking Certificate 8348B3F146A765581946040D5C4D590A
. . .
```

Observe el valor resaltado en la línea `Revoking Certificate`. Este valor es el número de serie único del certificado que se revocará. Necesitará este valor si desea examinar la lista de revocación en el último paso de esta sección para verificar que el certificado se encuentre en ella.

Una vez que se confirme la acción, la CA revocará el certificado. Sin embargo, los sistemas remotos que se basan en la CA no tienen forma de verificar si se revocó un certificado. Los usuarios y los servidores podrán seguir utilizando el certificado hasta que la lista de revocación de certificados (CRL) se distribuya a todos los sistemas que se basan en la CA.

En el siguiente paso, generará una CRL o actualizará un archivo `crl.pem` ex

Generar una lista de revocación de certificados

Ahora que revocó un certificado, es importante actualizar la lista de certificados revocados en su servidor de CA. Una vez que tenga una lista de revocación actualizada, podrá indicar los usuarios y sistemas que tienen certificados válidos en su CA.

Para generar una CRL, ejecute el comando `easy-rsa` con la opción `gen-crl` mientras se encuentre en el directorio `~/easy-rsa`:

```
$ ./easyrsa gen-crl
```

Si usó una frase de contraseña cuando creó su archivo `ca.key`, se le solicitará ingresarla. El comando `gen-crl` generará un archivo llamado `crl.pem`, que contiene la lista actualizada de los certificados revocados de esa CA.

A continuación, deberá transferir el archivo `crl.pem` actualizado a todos los servidores y clientes que se basan en esta CA cada vez que ejecute el comando `gen-crl`. De lo contrario, los clientes y sistemas podrán seguir accediendo a los servicios y sistemas que utilizan su CA, dado que esos servicios tienen que conocer el estado de revocación del certificado.

Transferir una lista de revocación de certificados

Ahora que generó una CRL en su servidor de CA, deberá transferirla a los sistemas remotos que se basan en su CA. Para transferir este archivo a sus servidores, puede usar el comando `scp`.

> **Nota:** En este tutorial, se explica la forma de generar y distribuir una CRL manualmente. Si bien hay métodos más sólidos y automatizados para distribuir y verificar las listas de revocación, como OCSP-Stapling, la configuración de esos métodos está fuera del alcance de este artículo.

Asegúrese de estar conectado a su servidor de CA como non-root user y ejecute lo siguiente; sustituya `your_server_ip` por el IP de su servidor o su nombre de DNS:

```
$ scp ~/easy-rsa/pki/crl.pem sammy@your_server_ip:/tmp
```

Ahora que el archivo se encuentra en el sistema remoto, el último paso es actualizar todos los servicios con la nueva copia de la lista de revocación.

Actualizar servicios que admiten una CRL

En este tutorial, no se enumeran los pasos necesarios para actualizar los servicios que utilizan el archivo `crl.pem`. En general, deberá copiar el archivo `crl.pem` a la ubicación que el servicio espera y, luego, reiniciarlo mediante `systemctl`.

Una vez que actualice sus servicios con el nuevo archivo `crl.pem`, podrán rechazar las conexiones de clientes o servidores que usen un certificado revocado.

Examinar y verificar los componentes de una CRL

Si desea examinar un archivo CRL, por ejemplo, para confirmar una lista de certificados revocados, utilice el siguiente comando `openssl` desde el directorio `easy-rsa` de su servidor de CA:

```
$ cd ~/easy-rsa
$ openssl crl -in pki/crl.pem -noout -text
```

También puede ejecutar este comando en cualquier servidor o sistema que tenga la herramienta `openssl` instalada con una copia del archivo `crl.pem`. Por ejemplo, si transfirió el archivo `crl.pem` a su segundo sistema y desea verificar que el certificado de `sammy-server` se haya revocado, puede usar un comando `openssl`, como el siguiente, sustituyendo el número de serie resaltado aquí por el que observó anteriormente cuando revocó el certificado:

```
$ openssl crl -in /tmp/crl.pem -noout -text |grep -A 1 8348B3F146A765581946040D5C4D590A
```

```
Output
    Serial Number: 8348B3F146A765581946040D5C4D590A
        Revocation Date: Apr  1 20:48:02 2020 GMT
```

Observe que se usa el comando `grep` para verificar el número de serie único que observó en el paso de revocación. Ahora, puede verificar el contenido de su lista de revocación de certificados en cualquier sistema que se base en ella para restringir el acceso a usuarios y servicios.

Conclusión

En este tutorial, creó una entidad de certificación privada usando el paquete Easy-RSA en un servidor independiente de Ubuntu 20.04. Aprendió sobre el funcionamiento del modelo de confianza entre partes que se basan en la CA. También creó y firmó una solicitud de firma de certificado (CSR) para un servidor de prueba y, luego, aprendió a revocar certificados. Por último, aprendió a generar y distribuir una lista de revocación de certificados (CRL) para cualquier sistema que se base en su CA con el fin de garantizar que los usuarios o servidores que no deban tener acceso a los servicios no puedan tenerlo.

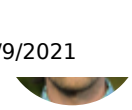
Ahora, puede emitir certificados para usuarios y usarlos con servicios como OpenVPN. También puede usar su CA para configurar servidores web de desarrollo o ensayo con certificados para proteger sus entornos de no producción. El uso de una CA con certificados TLS durante el desarrollo puede ayudarlo a asegurarse de que su código y sus entornos se adapten en la mayor medida posible a su entorno de producción.

Si desea saber más sobre cómo usar OpenSSL, nuestro tutorial Conceptos básicos de OpenSSL: Trabajar con certificados SSL, claves privadas y CSR contiene mucha información adicional para ayudarlo a familiarizarse con los fundamentos de OpenSSL.

About the authors

Jamon Camisso

has authored 42 tutorials



has authored 42 tutorials.

Still looking for an answer?

Ask a question

Search for more help

RELATED

Cómo instalar MongoDB desde los repositorios APT predeterminados de Ubuntu 20.04

Tutorial

Cómo editar el archivo sudoers

Tutorial

Comments

0 Comments

Leave a comment...

Sign In to Comment

GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

SCROLL TO TOP

Featured on Community Kubernetes Course Learn Python 3 Machine Learning in Python Getting started with Go Intro to Kubernetes

DigitalOcean Products Virtual Machines Managed Databases Managed Kubernetes Block Storage Object Storage Marketplace VPC Load Balancers

Company

About
Leadership
Blog
Careers
Partners
Referral Program
Press
Legal
Security & Trust Center

Products

Pricing
Products Overview
Droplets
Kubernetes
Managed Databases
Spaces
Marketplace
Load Balancers
Block Storage
API Documentation
Documentation
Release Notes

Community

Tutorials
Q&A
Tools and Integrations
Tags
Product Ideas
Write for DigitalOcean
Presentation Grants
Hatch Startup Program
Shop Swag
Research Program
Open Source
Code of Conduct

Contact

Get Support
Trouble Signing In?
Sales
Report Abuse
System Status

SCROLL TO TOP