SELF-HOSTING

Opciones de Certificado

Opciones de Certificado

Este artículo define las opciones de certificado disponibles para instancias autoalojadas de Bitwarden. Seleccionará su opción de certificado durante la instalación.

> **Note**
> La información en este artículo puede no aplicarse a las implementaciones autoalojadas unificadas de Bitwarden.

Generar un certificado con Let's Encrypt

Let's Encrypt es una autoridad certificadora (CA) que emite certificados SSL confiables de forma gratuita para cualquier dominio. El script de instalación de Bitwarden ofrece la opción de generar un certificado SSL confiable para su dominio utilizando Let's Encrypt y Certbot.

Las verificaciones de renovación de certificados ocurren cada vez que Bitwarden se reinicia. Usar Let's Encrypt requerirá que ingreses una dirección de correo electrónico para recordatorios de vencimiento de certificados.

El uso de Let's Encrypt requiere que los puertos 80 y 443 estén abiertos en tu máquina.

Actualiza manualmente un certificado de Let's Encrypt

Si cambia el nombre de dominio de su servidor Bitwarden, necesitará actualizar manualmente su certificado generado. Ejecute los siguientes comandos para crear una copia de seguridad, actualizar su certificado y reconstruir Bitwarden:

Fiesta

```bash
./bitwarden.sh stop

mv ./bwdata/letsencrypt ./bwdata/letsencrypt_backup

mkdir ./bwdata/letsencrypt

chown -R bitwarden:bitwarden ./bwdata/letsencrypt

chmod -R 740 ./bwdata/letsencrypt

docker pull certbot/certbot

docker run -i --rm --name certbot -p 443:443 -p 80:80 -v <Full Path from / >/bwdata/letsencrypt:/et
c/letsencrypt/ certbot/certbot certonly --email <user@email.com> --logs-dir /etc/letsencrypt/logs
```

Seleccione 1, luego siga las instrucciones:

```bash
openssl dhparam -out ./bwdata/letsencrypt/live/<your.domain.com>/dhparam.pem 2048
./bitwarden.sh rebuild
./bitwarden.sh start
```

PowerShell

> **Tip**
> Necesitarás instalar una versión de OpenSSL para Windows.

```bash
.\bitwarden.ps1 -stop
mv .\bwdata\letsencrypt .\bwdata\letsencrypt_backup
mkdir .\bwdata\letsencrypt
docker pull certbot/certbot
docker run -i --rm --name certbot -p 443:443 -p 80:80 -v <Full Path from \ >\bwdata\letsencrypt\:/e
tc/letsencrypt/ certbot/certbot certonly --email <user@email.com> --logs-dir /etc/letsencrypt/logs
Select 1, then follow instructions
<path/to/openssl.exe> dhparam -out .\bwdata\letsencrypt\live\<your.domain.com>\dhparam.pem 2048
.\bitwarden.ps1 -rebuild
.\bitwarden.ps1 -start
```

Usa un certificado SSL existente

También puedes optar por usar un certificado SSL existente, lo cual requerirá que tengas los siguientes archivos:

- Un certificado de servidor (`certificate.crt`)
- Una clave privada (`private.key`)
- Un certificado CA (`ca.crt`)

Es posible que necesite agrupar su certificado principal con certificados de CA intermedios para prevenir errores de confianza SSL. Todos los certificados deben incluirse en el archivo de certificado del servidor cuando se utiliza un certificado de CA. El primer certificado en el archivo debe ser el certificado de su servidor, seguido de cualquier certificado(s) de CA intermedio, seguido de la CA raíz.

Bajo la configuración predeterminada, coloque sus archivos en `./bwdata/ssl/su.dominio`. Puede especificar una ubicación diferente para sus archivos de certificado editando los siguientes valores en `./bwdata/config.yml`:

```
Bash

ssl_certificate_path: <path>
ssl_key_path: <path>
ssl_ca_path: <path>
```

Note

Los valores definidos en `config.yml` representan ubicaciones dentro del contenedor NGINX. Los directorios en el host se mapean a directorios dentro del contenedor NGINX. Bajo la configuración predeterminada, las asignaciones se alinean de la siguiente manera:

Los siguientes valores en `config.yml`:

```
Bash

ssl_certificate_path: /etc/ssl/your.domain/certificate.crt
ssl_key_path: /etc/ssl/your.domain/private.key
ssl_ca_path: /etc/ssl/your.domain/ca.crt
```

Mapear a los siguientes archivos en el anfitrión:

```
Bash

./bwdata/ssl/your.domain/certificate.crt
./bwdata/ssl/your.domain/private.key
./bwdata/ssl/your.domain/ca.crt
```

Solo debería necesitar trabajar con archivos en `./bwdata/ssl/`. No se recomienda trabajar con archivos directamente en el contenedor NGINX.

Usando el intercambio de claves Diffie-Hellman

Opcionalmente, si se utiliza el intercambio de claves Diffie-Hellman para generar parámetros efímeros:

- Incluye un archivo `dhparam.pem` en el mismo directorio.
- Establece el valor de `ssl_diffie_hellman_path:` en `config.yml`.

Note

Puedes crear tu propio archivo `dhparam.pem` usando OpenSSL con `openssl dhparam -out ./dhparam.pem 2048`.

Usando un Certificado autofirmado

También puedes optar por usar un certificado autofirmado, sin embargo, esto solo se recomienda para pruebas.

Los certificados autofirmados no serán confiables por defecto para las aplicaciones cliente de Bitwarden. Se requerirá que instale manualmente este certificado en la tienda de confianza de cada dispositivo que planea usar con Bitwarden.

Generar un certificado autofirmado:

```Bash
mkdir ./bwdata/ssl/bitwarden.example.com
openssl req -x509 -newkey rsa:4096 -sha256 -nodes -days 365 \
  -keyout ./bwdata/ssl/bitwarden.example.com/private.key \
  -out ./bwdata/ssl/bitwarden.example.com/certificate.crt \
  -reqexts SAN -extensions SAN \
  -config <(cat /usr/lib/ssl/openssl.cnf <(printf '[SAN]\nsubjectAltName=DNS:bitwarden.example.com
\nbasicConstraints=CA:true')) \
  -subj "/C=US/ST=New York/L=New York/O=Company Name/OU=Bitwarden/CN=bitwarden.example.com"
```

Su certificado autofirmado (`.crt`) y clave privada (`private.key`) pueden ser colocados en el directorio `./bwdata/ssl/self/your.domain` y configurados en el `./bwdata/config.yml`:

```Bash
 ssl_certificate_path: /etc/ssl/bitwarden.example.com/certificate.crt
 ssl_key_path: /etc/ssl/bitwarden.example.com/private.key
```

Confía en un certificado autofirmado

Ventanas

Para confiar en un certificado autofirmado en Windows, ejecute `certmgr.msc` e importe su certificado en las Autoridades de Certificación Raíz de Confianza.

Linux

Para confiar en un certificado autofirmado en Linux, añade tu certificado a los siguientes directorios:

```Bash
/usr/local/share/ca-certificates/
/usr/share/ca-certificates/
```

Y ejecuta los siguientes comandos:

```bash
sudo dpkg-reconfigure ca-certificates
sudo update-ca-certificates
```

Para nuestra aplicación de escritorio Linux, para acceder a la caja fuerte web utilizando navegadores basados en Chromium, y la aplicación de escritorio del Conector de Directorio, también necesitas completar este procedimiento de gestión de certificados Linux.

Para el Bitwarden ILC y ILC del Conector de Directorio, su certificado autofirmado debe estar almacenado en un archivo local y referenciado por una variable de entorno `NODE_EXTRA_CA_CERTS=`, por ejemplo:

```bash
export NODE_EXTRA_CA_CERTS=~/.config/Bitwarden/certificate.crt
```

Androide

Para confiar en un certificado autofirmado en un dispositivo Android, consulte la documentación de Google sobre Agregar y eliminar certificados.

Note

Si **no está autoalojado** y encuentra el siguiente error de certificado en su dispositivo Android:

```bash
Exception message: java.security.cert.CertPathValidatorException: Trust anchor for certificati
on path not found.
```

Necesitarás subir los certificados de Bitwarden a tu dispositivo. Consulte este hilo de la comunidad para obtener ayuda para encontrar los certificados.

No use certificado

Warning

Si optas por no usar un certificado, **debes respaldar tu instalación con un proxy que sirve Bitwarden sobre SSL**. Esto es porque Bitwarden requiere HTTPS; intentar usar Bitwarden sin el protocolo HTTPS provocará errores.