SELF-HOSTING

Options de Certificat

Options de Certificat

Cet article définit les options de certificat disponibles pour les instances auto-hébergées de Bitwarden. Vous sélectionnerez votre option de certificat pendant l'installation.

> **Note**
> Les informations dans cet article peuvent ne pas s'appliquer aux déploiements unifiés de Bitwarden auto-hébergés.

Générer un certificat avec Let's Encrypt

Let's Encrypt est une autorité de certification (CA) qui délivre des certificats SSL de confiance gratuitement pour tout domaine. Le script d'installation de Bitwarden offre l'option de générer un certificat SSL de confiance pour votre domaine en utilisant Let's Encrypt et Certbot.

Les vérifications de renouvellement de certificat ont lieu chaque fois que Bitwarden est redémarré. L'utilisation de Let's Encrypt vous demandera de saisir une adresse de courriel pour les rappels d'expiration de certificat.

L'utilisation de Let's Encrypt nécessite que les ports 80 et 443 soient ouverts sur votre machine.

Mettez à jour manuellement un certificat Let's Encrypt

Si vous changez le nom de domaine de votre serveur Bitwarden, vous devrez mettre à jour manuellement votre certificat généré. Exécutez les commandes suivantes pour créer une sauvegarde, mettre à jour votre certificat et reconstruire Bitwarden :

Bash

```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
```

Sélectionnez 1, puis suivez les instructions:

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

PowerShell

> **Tip**
> Vous devrez installer une version d'OpenSSL pour 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
```

Utilisez un certificat SSL existant

Vous pouvez également choisir d'utiliser un certificat SSL existant, ce qui vous obligera à avoir les fichiers suivants :

- Un certificat de serveur (`certificate.crt`)
- Une clé privée (`private.key`)
- Un certificat CA (`ca.crt`)

Vous devrez peut-être regrouper votre certificat principal avec des certificats CA intermédiaires pour éviter les erreurs de confiance SSL. Tous les certificats doivent être inclus dans le fichier de certificat du serveur lors de l'utilisation d'un certificat CA. Le premier certificat dans le fichier devrait être votre certificat de serveur, suivi de tout certificat(s) d'autorité de certification intermédiaire, suivi de l'autorité de certification racine.

Dans la configuration par défaut, placez vos fichiers dans `./bwdata/ssl/votre.domaine`. Vous pouvez spécifier un emplacement différent pour vos fichiers de certificat en éditant les valeurs suivantes dans `./bwdata/config.yml` :

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

Note

Les valeurs définies dans `config.yml` représentent des emplacements à l'intérieur du conteneur NGINX. Les répertoires sur l'hôte sont mappés aux répertoires à l'intérieur du conteneur NGINX. Dans la configuration par défaut, les mappages s'alignent comme suit :

Les valeurs suivantes dans `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
```

Cartographiez les fichiers suivants sur l'hôte :

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

Vous ne devriez avoir besoin de travailler qu'avec des fichiers dans `./bwdata/ssl/`. Il n'est pas recommandé de travailler directement avec les fichiers dans le conteneur NGINX.

En utilisant l'échange de clés Diffie-Hellman

Optionnellement, si vous utilisez l'échange de clés Diffie-Hellman pour générer des paramètres éphémères:

- Incluez un fichier `dhparam.pem` dans le même répertoire.
- Définissez la valeur `ssl_diffie_hellman_path:` dans `config.yml`.

Note

Vous pouvez créer votre propre fichier `dhparam.pem` en utilisant OpenSSL avec `openssl dhparam -out ./dhparam.pem 2048`.

Utilisation d'un certificat auto-signé

Vous pouvez également choisir d'utiliser un certificat auto-signé, cependant cela n'est recommandé que pour les tests.

Les certificats auto-signés ne seront pas considérés comme fiables par défaut par les applications client Bitwarden. Vous devrez installer manuellement ce certificat dans le magasin de confiance de chaque appareil que vous prévoyez d'utiliser avec Bitwarden.

Générer un certificat auto-signé:

```
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"
```

Votre certificat auto-signé (`.crt`) et votre clé privée (`private.key`) peuvent être placés dans le répertoire `./bwdata/ssl/self/your.domain` et configurés dans le `./bwdata/config.yml` :

```
Bash

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

Faites confiance à un certificat auto-signé

Fenêtres

Pour faire confiance à un certificat auto-signé sur Windows, exécutez `certmgr.msc` et importez votre certificat dans les Autorités de Certification Racine de Confiance.

Linux

Pour faire confiance à un certificat auto-signé sur Linux, ajoutez votre certificat aux répertoires suivants :

```
Bash

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

Et exécutez les commandes suivantes :

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

Pour notre application de bureau Linux, pour accéder au coffre web en utilisant des navigateurs basés sur Chromium, et l'application de bureau du connecteur de répertoire, vous devez également compléter cette procédure de gestion de certificat Linux.

Pour le Bitwarden CLI et le Directory Connector CLI, votre certificat auto-signé doit être stocké dans un fichier local et référencé par une variable d'environnement `NODE_EXTRA_CA_CERTS=`, par exemple :

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

Android

Pour faire confiance à un certificat auto-signé sur un appareil Android, référez-vous à la documentation de Google sur l'ajout et la suppression de certificats.

> **Note**
>
> Si vous n'êtes **pas auto-hébergé** et rencontrez l'erreur de certificat suivante sur votre appareil Android :
>
> ```Bash
> Exception message: java.security.cert.CertPathValidatorException: Trust anchor for certificati
> on path not found.
> ```
>
> Vous devrez téléverser les certificats de Bitwarden sur votre appareil. Reportez-vous à ce fil de discussion de la communauté pour obtenir de l'aide pour trouver les certificats.

N'utilisez aucun certificat

> **Warning**
>
> Si vous choisissez de ne pas utiliser de certificat, vous **devez mettre en avant votre installation avec un proxy qui sert Bitwarden sur SSL**. C'est parce que Bitwarden nécessite HTTPS ; essayer d'utiliser Bitwarden sans le protocole HTTPS déclenchera des erreurs.