Troubleshooting TLS errors

In this article

If you run into TLS issues with your appliance, you can take actions to resolve them.

Removing the passphrase from your key file

If you have a Linux machine with OpenSSL installed, you can remove your passphrase.

1. Rename your original key file.

   ```
   mv yourdomain.key yourdomain.key.orig
   ```

2. Generate a new key without a passphrase.

   ```
   openssl rsa -in yourdomain.key.orig -out yourdomain.key
   ```

You'll be prompted for the key's passphrase when you run this command.

For more information about OpenSSL, see OpenSSL's documentation.

Converting your TLS certificate or key into PEM format

If you have OpenSSL installed, you can convert your key into PEM format by using the `openssl` command. For example, you can convert a key from DER format into PEM format.

```
openssl rsa -in yourdomain.der -inform DER -out yourdomain.key -outform PEM
```

Otherwise, you can use the SSL Converter tool to convert your certificate into the PEM format. For more information, see the SSL Converter tool's documentation.

Unresponsive installation after uploading a key

If your GitHub Enterprise Server instance is unresponsive after uploading an TLS key, please contact GitHub Enterprise Support with specific details, including a copy of your TLS certificate. Ensure that your private key **is not** included.

Certificate validity errors

Clients such as web browsers and command-line Git will display an error message if they cannot verify the validity of an TLS certificate. This often occurs with self-signed certificates as well as "chained root" certificates issued from an intermediate root certificate that is not recognized by the client.

If you are using a certificate signed by a certificate authority (CA), the certificate file that you upload to GitHub Enterprise Server must include a certificate chain with that CA's root certificate. To create such a file, concatenate your entire certificate chain (or "certificate bundle") onto the end of your certificate, ensuring that the principal certificate with your hostname comes first. On most systems you can do this with a command similar to:

```
cat yourdomain.com.crt bundle-certificates.crt > yourdomain.combined.crt
```

You should be able to download a certificate bundle (for example, `bundle-certificates.crt`) from your certificate authority or TLS vendor.

Installing self-signed or untrusted certificate authority (CA) root certificates

If your GitHub Enterprise Server appliance interacts with other machines on your network that use a self-signed or untrusted certificate, you will need to import the signing CA's root certificate into the system-wide certificate store in order to access those systems over HTTPS.

1. Obtain the CA's root certificate from your local certificate authority and ensure it is in PEM format.
2. Copy the file to your GitHub Enterprise Server appliance over SSH as the "admin" user on port 122.

   ```
   scp -P 122 rootCA.crt admin@HOSTNAME:/home/admin
   ```

3. Connect to the GitHub Enterprise Server administrative shell over SSH as the "admin" user on port 122.

   ```
   ssh -p 122 admin@HOSTNAME
   ```

4. Import the certificate into the system-wide certificate store.

   ```
   ghe-ssl-ca-certificate-install -c rootCA.crt
   ```

Updating a TLS certificate

You can generate a new self-signed certificate or update an existing TLS certificate for your GitHub Enterprise Server instance with the `ghe-ssl-certificate-setup` command

line utility. For more information, see "Command-line utilities."