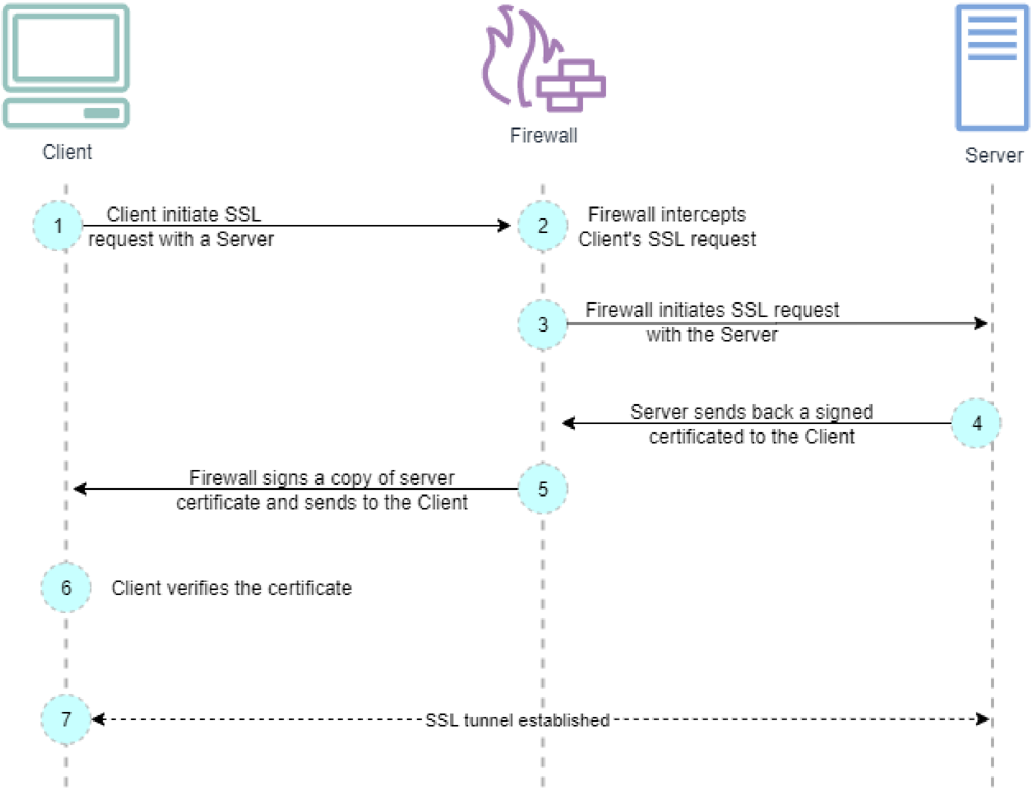
Raymond

Docker with WSL 2, SSL Proxy and SSL Certificate Problem

2022-06-20 1 13,518 0


Client
Firewall
Server
1
Client initiate SSL request with a Server
2
Firewall intercepts Client's SSL request
3
Firewall initiates SSL request with the Server
4
Server sends back a signed certificated to the Client
5
Firewall signs a copy of server certificate and sends to the Client
6
Client verifies the certificate
7
SSL tunnel established

In many companies, proxy including MITM (man-in-the-middle) SSL forward proxy are added to enhance network security. This can cause problems when you use Docker Desktop with WSL 2 base engine. For example, when you need to connect to internet to download packages for your applications, the https may not work due to error - **SSL certificate problem: unable to get local issuer certificate**.

This article provides some learnings I had in the past few days and the resolutions to fix these problems.

Environment

- Host OS: Windows 10
- WSL 2 distro: Ubuntu 20.04
- Docker version: 4.2.0. WSL 2 is used as base engine.
- Proxy is enforced in the network (i.e. SSL forward proxy occurs).

These issues can apply to Docker in Ubuntu or other WSL Linux distros, Linux distros, or Docker in Hyper-V virtual machines, etc.

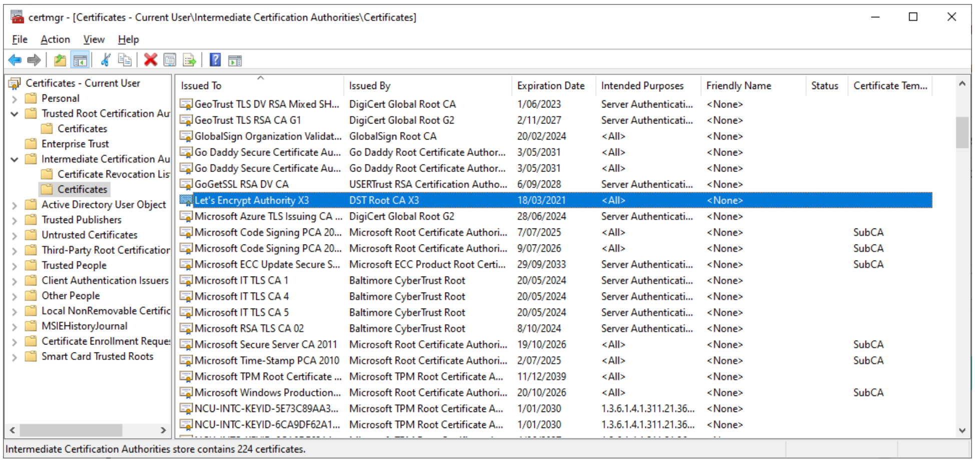
Prerequisites

The high-level approach to fix this problem is to add the proxy certificate for example, Zscaler's root certificate to your docker images as part of docker build. If you don't know the specific certificate, check with your IT network security team.

If you know it, you can export the certificate directly by following these steps:

> ⚠ The following step is for demo purpose only and the actual certificate name and path in your company can be different.

- Type **certmgr.msc** using Windows search button.
- In the window, find out the certificate of the proxy:



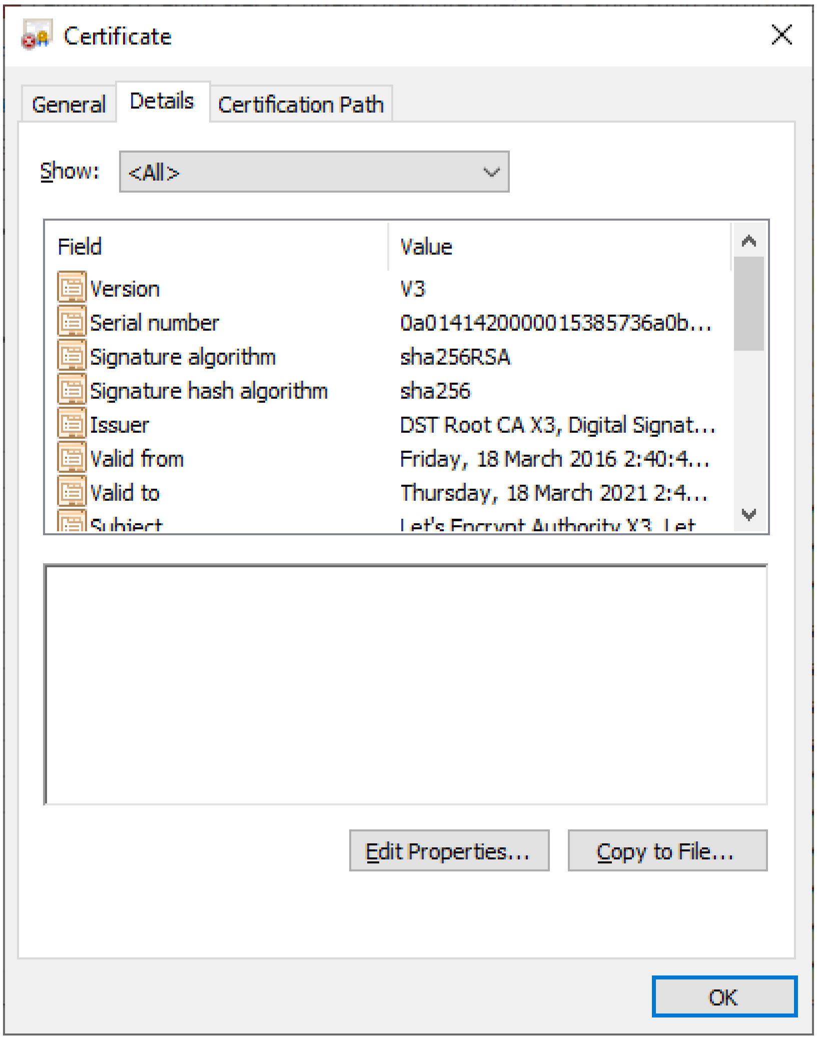
certmgr - [Certificates - Current User\Intermediate Certification Authorities\Certificates]

File
Action
View
Help
Certificates - Current User
Personal
Trusted Root Certification Au
Certificates
Enterprise Trust
Intermediate Certification Au
Certificate Revocation Lis
Certificates
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification
Trusted People
Client Authentication Issuers
Other People
Local NonRemovable Certific
MSIEHistoryJournal
Certificate Enrollment Reque
Smart Card Trusted Roots

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Status	Certificate Tem...
GeoTrust TLS DV RSA Mixed SH...	DigiCert Global Root CA	1/06/2023	Server Authenticati...	<None>		
GeoTrust TLS RSA CA G1	DigiCert Global Root G2	2/11/2027	Server Authenticati...	<None>		
GlobalSign Organization Validat...	GlobalSign Root CA	20/02/2024	<All>	<None>		
Go Daddy Secure Certificate Au...	Go Daddy Root Certificate Author...	3/05/2031	<All>	<None>		
Go Daddy Secure Certificate Au...	Go Daddy Root Certificate Author...	3/05/2031	<All>	<None>		
GoGetSSL RSA DV CA	USERTrust RSA Certification Autho...	6/09/2028	Server Authenticati...	<None>		
Let's Encrypt Authority X3	DST Root CA X3	18/03/2021	<All>	<None>		
Microsoft Azure TLS Issuing CA ...	DigiCert Global Root G2	28/06/2024	Server Authenticati...	<None>		
Microsoft Code Signing PCA 20...	Microsoft Root Certificate Authori...	7/07/2025	<All>	<None>		SubCA
Microsoft Code Signing PCA 20...	Microsoft Root Certificate Authori...	9/07/2026	<All>	<None>		SubCA
Microsoft ECC Update Secure S...	Microsoft ECC Product Root Certi...	29/09/2033	Server Authenticati...	<None>		SubCA
Microsoft IT TLS CA 1	Baltimore CyberTrust Root	20/05/2024	Server Authenticati...	<None>		
Microsoft IT TLS CA 4	Baltimore CyberTrust Root	20/05/2024	Server Authenticati...	<None>		
Microsoft IT TLS CA 5	Baltimore CyberTrust Root	20/05/2024	Server Authenticati...	<None>		
Microsoft RSA TLS CA 02	Baltimore CyberTrust Root	8/10/2024	Server Authenticati...	<None>		
Microsoft Secure Server CA 2011	Microsoft Root Certificate Authori...	19/10/2026	<All>	<None>		SubCA
Microsoft Time-Stamp PCA 2010	Microsoft Root Certificate Authori...	2/07/2025	<All>	<None>		SubCA
Microsoft TPM Root Certificate ...	Microsoft TPM Root Certificate A...	11/12/2039	<All>	<None>		
Microsoft Windows Production...	Microsoft Root Certificate Authori...	20/10/2026	<All>	<None>		SubCA
NCU-INTC-KEYID-5E73C89AA3...	Microsoft TPM Root Certificate A...	1/01/2030	1.3.6.1.4.1.311.21.36...	<None>		
NCU-INTC-KEYID-6CA9DF62A1...	Microsoft TPM Root Certificate A...	1/01/2030	1.3.6.1.4.1.311.21.36...	<None>		

Intermediate Certification Authorities store contains 224 certificates.

- Double click the certificate row and go to **Details** tab:

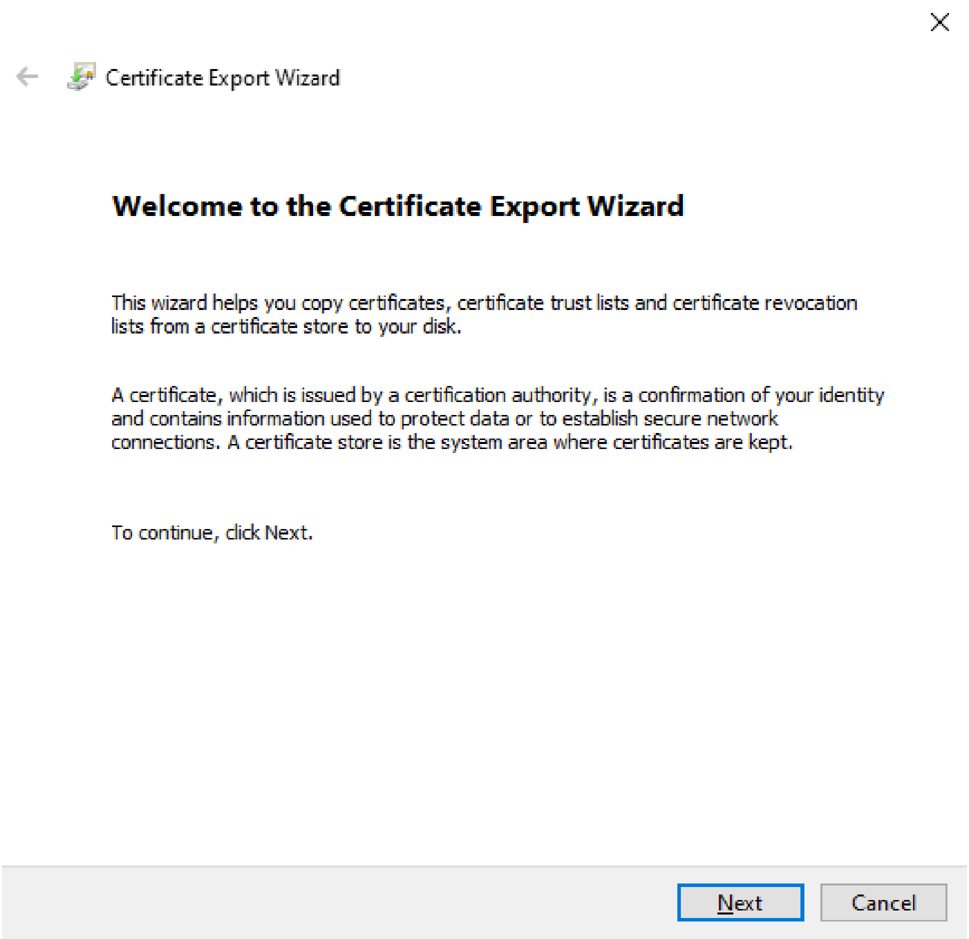


Show: <All>

Field	Value
Version	V3
Serial number	0a0141420000015385736a0b...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	DST Root CA X3, Digital Signat...
Valid from	Friday, 18 March 2016 2:40:4...
Valid to	Thursday, 18 March 2021 2:4...
Subject	Let's Encrypt Authority X3, Let...

- Click button **Copy to File**.

- Click **Next** button:



Certificate Export Wizard

Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

- Select **Base-64 encoded X.509 (.CER)**
- Choose the path and finish the export. For this article, we will call this certificate as **myproxy.cer**.
- Copy this certification file into your WSL distro.
- Install **openssl** package if not installed:

```
sudo apt-get install openssl
```

- Use the following command line to convert

```
openssl x509 -inform PEM -in <filepath>/myproxy.cer -out myproxy.crt
```

- We will use **myproxy.crt** in the following sections.

Double-confirm the issue

Before we dive into the solutions, let's double confirm you have this issue.

Scenario 1 - openssl connection

Logon to your Ubuntu WSL 2 distro and use **openssl** to verify.

```
openssl s_client -connect https://the-domain-you-are-connecting-
```

```
to.com -showcerts
```

If your company's proxy cert is not trusted in the distro, the following error can show:

> **Verify return code: 20 (unable to get local issuer certificate)**

Scenario 2 - node:gallium-alpine

Create a **Dockerfile** file in a project folder. Add the following content:

```
FROM node:gallium-alpine

WORKDIR /app

RUN apk update && apk add bash netcat-openbsd

RUN ["echo", "Hello, Kontext!"]
```

Run docker command under the project folder context:

```
docker build .
```

The following error will show up:

```
fetch https://dl-
cdn.alpinelinux.org/alpine/v3.13/main/x86_64/APKINDEX.tar.gz
140124873489224:error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify
failed:ssl/statem/statem_clnt.c:1913:
ERROR: https://dl-cdn.alpinelinux.org/alpine/v3.13/main:
Permission denied
WARNING: Ignoring https://dl-
cdn.alpinelinux.org/alpine/v3.13/main: No such file or directory
fetch https://dl-
cdn.alpinelinux.org/alpine/v3.13/community/x86_64/APKINDEX.tar.g
z
140124873489224:error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify
failed:ssl/statem/statem_clnt.c:1913:
ERROR: https://dl-cdn.alpinelinux.org/alpine/v3.13/community:
Permission denied
WARNING: Ignoring https://dl-
cdn.alpinelinux.org/alpine/v3.13/community: No such file or
directory
```

Scenario 3 - amazon/aws-cli:2.1.35

Create another project folder with docker file Dockerfile:

```
FROM amazon/aws-cli:2.1.35

WORKDIR /app

RUN yes y | yum install jq

RUN ["echo", "Hello, Kontext!"]
```

Run docker command under the project folder context:

```
docker build .
```

The following error will show up:

```
failure: repodata/repomd.xml from amzn2-core: [Errno 256] Mo
more mirrors to try.
https://cdn.amazonlinux.com/2/core/2.0/x86_64/****** [Errno 14]
curl#60 - "SSL certificate problem: unable to get local issuer
certificate"
```

So all the three scenarios complain about local issuer certificate issue. Thus the resolution is simple - adding the proxy certificate myproxy.crt to the trusted store in the distro or container.

Solutions to fix SSL local issuer certificate issue

Now let's add the solutions for each of the three scenarios listed above.

Scenario 1 - openssl connection

To fix this issue, we just need to trust the certificate using the following command:

```
$ sudo apt-get install -y ca-certificates
$ sudo cp myproxy.crt /usr/local/share/ca-certificates
$ sudo update-ca-certificates
```

Then run the OpenSSL connection command again, it should work like a charm.

Scenario 2 - node:gallium-alpine

AS the image name suggests, it is based on Alpine Linux. To update the certificates in Alpine, we need to connect to internet thus we cannot add it directly (Correct me if I am wrong as I am not a Linux expert). One workaround is to replace Alpine repositories file (`/etc/apk/repositories`) to change https to http.

This can be done by adding a line into docker file:

```
FROM node:gallium-alpine

WORKDIR /app

RUN sed -i 's/https/http/' /etc/apk/repositories

RUN apk update && apk add bash netcat-openbsd

RUN ["echo", "Hello, Kontext!"]
```

Now it should also work.

> ⚠ By changing from https to http, it can add security concerns. Thus please use it cautiously. Another approach is to change your base image to use other Linux/UNIX distros.

Scenario 3 - amazon/aws-cli:2.1.35

Similar as scenario 2, we just need to either avoid SSL or trust the proxy cert **myproxy.crt**.

Add **myproxy.crt** to your project folder and copy it to Docker image and then trust it there:

```
FROM amazon/aws-cli:2.1.35

WORKDIR /app

COPY myproxy.crt /etc/pki/ca-trust/source/anchors/
RUN update-ca-trust extract

RUN yes y | yum install jq

RUN ["echo", "Hello, Kontext!"]
```

This now also works. **The highlighted part can be different from one Linux distro to another, please change them accordingly to apply to your own environments.** The AWS CLI docker image is based on Amazon Linux 2, thus I am using the above two command lines.

I hope this article helps you to get your container application build or other build work under company proxy environment. Let me if you find other issues and I will try my best to help.

References

Installing a root CA certificate in the trust store | Ubuntu

More from Kontext

- SSL Forward Proxy
- Danger: Bypass SSL Interception Proxy Server Certificate Validation in .NE...
- # ssl
- How to install and setup Let's Encrypt on Azure Web Apps
- How to upgrade from WSL 1 to WSL 2
- HTTPS Link Generation for ASP.NET Core App hosted in Azure Container ...

docker wsl ubuntu

Comments

No comments yet.

Please log in or register to comment.

Log in Register

Log in with external accounts

Log in with Microsoft account

Log in with Google account

Home / Columns / Tools

Kontext

Made with ♡ in Melbourne.

Products & features

Columns

Diagrams

Tags

Search

Resources

Subscribe RSS

Help centre

About

Cookie

Privacy

Terms

Contact us

Subscription

Subscribe to Kontext newsletter to get updates about data analytics, programming and cloud related articles.

Email

➢ Subscribe