Running server-side simulation with SSL

This chapter describes two ways to run server-side simulation with a self-signed certificate.

> **Prerequisites**
>
> - The self-signed certificate used for server-side simulation must be generated for the IP (or hostname) of the machine where Web Application Platform is located. This is necessary because Web Application Platform (and not Teamwork Cloud) is called when connecting to the server. Therefore, if Teamwork Cloud is deployed on a different server, do not use the same certificate for both Teamwork Cloud and Web Application Platform.
> - The certificate file must be in the *.pem* format.

To run server-side simulation with SSL

- Do one of the following:
    - Use the following request to connect to Web Application Platform and provide the path to the certificate file:

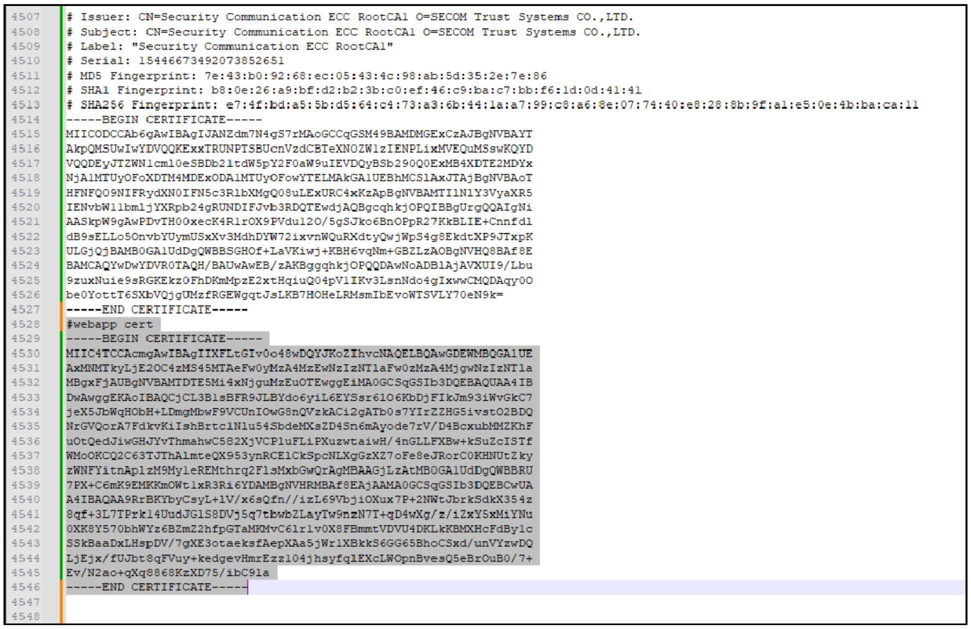
        > In the request, make sure to provide the IP or hostname for which the certificate was generated.

        ```
        client = SimulationWebClient('<WebApp_server_IP_or_hostname>', '<user_name>', '<password>',
        verifySSL='<path_to_certificate_file>')
        ```

    - Add the self-signed certificate to the Python trusted certificates list by completing the following steps:
        1. In Jupyter Notebook, execute the following commands to find the path, where the *cacert.pem* file of trusted certificates is located:

            ```
            import certifi
            print(certifi.where())
            ```

        2. Open the certificate file with a text editor and copy all information inside the file.
        3. Open the *cacert.pem* file with a text editor and paste the copied information at the end of the file, as shown below.

```
# Issuer: CN=Security Communication ECC RootCA1 O=SECOM Trust Systems CO.,LTD.
# Subject: CN=Security Communication ECC RootCA1 O=SECOM Trust Systems CO.,LTD.
# Label: "Security Communication ECC RootCA1"
# Serial: 15446673492073852651
# MD5 Fingerprint: 7e:43:b0:92:68:ec:05:43:4c:98:ab:5d:35:2e:7e:86
# SHA1 Fingerprint: b8:0e:26:a9:bf:d2:b2:3b:c0:ef:46:c9:ba:c7:bb:f6:1d:0d:41:41
# SHA256 Fingerprint: e7:4f:bd:a5:5b:d5:64:c4:73:a3:6b:44:1a:a7:99:c8:a6:8e:07:74:40:e8:28:8b:9f:a1:e5:0e:4b:ba:ca:11
-----BEGIN CERTIFICATE-----
MIICODCCAb6gAwIBAgIJANZdm7N4gS7rMAoGCCqGSM49BAMDMGExCzAJBgNVBAYT
AkpQMSUwIwYDVQQKExxTRUNPTSBUcnVzdCBTeXN0ZW1zIENPLixMVEQuMSswKQYD
VQQDEyJTZWN1cml0eSBDb21tdW5pY2F0aW9uIEVDQyBSb290Q0ExMB4XDTE2MDYx
NjA1MTUyOFoXDTM4MDExODA1MTUyOFowYTELMAkGA1UEBhMCSlAxJTAjBgNVBAoT
HFNFQ09NIFRydXN0IFN5c3RlbXMgQ08uLExURC4xKzApBgNVBAMTIlNlY3VyaXR5
IENvbW11bmljYXRpb24gRUNDIFJvb3RDQTEwdjAQBgcqhkjOPQIBBgUrgQQAIgNi
AASkpW9gAwPDvTH00xecK4R1rOX9PVdu12O/5gSJko6BnOPpR27KkBLIE+Cnnfdl
dB9sELLo5OnvbYUymUSxXv3MdhDYW72ixvnWQuRXdtyQwjWpS4g8EkdtXP9JTxpK
ULGjQjBAMB0GA1UdDgQWBBSGHOf+LaVKiwj+KBH6vqNm+GBZLzAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAwNoADBlAjAVXUI9/Lbu
9zuxNuie9sRGKEkz0FhDKmMpzE2xtHqiuQ04pV1IKv3LsnNdo4gIxwwCMQDAqy0O
be0YottT6SXbVQjgUMzfRGEWgqtJsLKB7HOHeLRMsmIbEvoWTSVLY70eN9k=
-----END CERTIFICATE-----
#webapp cert
-----BEGIN CERTIFICATE-----
MIIC4TCCAcmgAwIBAgIIXFLtGIv0o48wDQYJKoZIhvcNAQELBQAwGDEWMBQGA1UE
AxMNMTkyLjE2OC4zMS45MTAeFw0yMzA4MzEwNzIzNTlaFw0zMzA4MjgwNzIzNTla
MBgxFjAUBgNVBAMTDTE5Mi4xNjguMzEuOTEwggEiMA0GCSqGSIb3DQEBAQUAA4IB
DwAwggEKAoIBAQCjCL3BlsBFR9JLBYdo6yiL6EYSsr61O6KbDjFIkJm93iWvGkC7
jeX5JbWqHObH+LDmgMbwF9VCUnIOwG8nQVzkACi2gATb0s7YIrZZHG5ivstO2BDQ
NrGVQorA7FdkvKiIshBrtc1Nlu54SbdeMXsZD4Sn6mAyode7rV/D4BcxubMMZKhF
uOtQedJiwGHJYvThmahwC582XjVCPluFLiPXuzwtaiwH/4nGLLFXBw+kSuZcISTf
WMoOKCQ2C63TJThAlmteQX953ynRCE1CkSpcNLXgGzXZ7oFe8eJRorC0KHNUtZky
zWNFYitnAplzM9MyleREMthrq2FlsMxbGwQrAgMBAAGjLzAtMB0GA1UdDgQWBBRU
7PX+C6mK9EMKKmOWtlxR3Ri6YDAMBgNVHRMBAf8EAjAAMA0GCSqGSIb3DQEBCwUA
A4IBAQAA9RrBKYbyCsyL+lV/x6sQfn//izL69VbjiOXux7P+2NWtJbrkSdkX354z
8qf+3L7TPrkl4UudJGlS8DVj5q7tbwbZLayTw9nzN7T+qD4wXg/z/iZxY5xMiYNu
0XK8Y570bhWYz6BZmZ2hfpGTaMKMvC6lrlv0X8FBmmtVDVU4DKLkKBMXHcFdBylc
SSkBaaDxLHspDV/7gXE3otaeksfAepXAa5jWrlXBkkS6GG65BhoCSxd/unVYzwDQ
LjEjx/fUJbt8qFVuy+kedgevHmrEzzl04jhsyfqlEXcLWOpnBvesQ5eBrOuB0/7+
Ev/N2ao+qXq8868KzXD75/ibC9la
-----END CERTIFICATE-----
```

        4. Use the following request to connect to Web Application Platform:

            > In the request, make sure to provide the IP or hostname for which the certificate was generated.

```
client = SimulationWebClient('WebApp_server_IP_or_hostname>', '<user_name>', '<password>',
verifySSL=True)
```