Article
Steve Pisani · Nov 23, 2021 4m read

Mutual TLS setup

Hi,

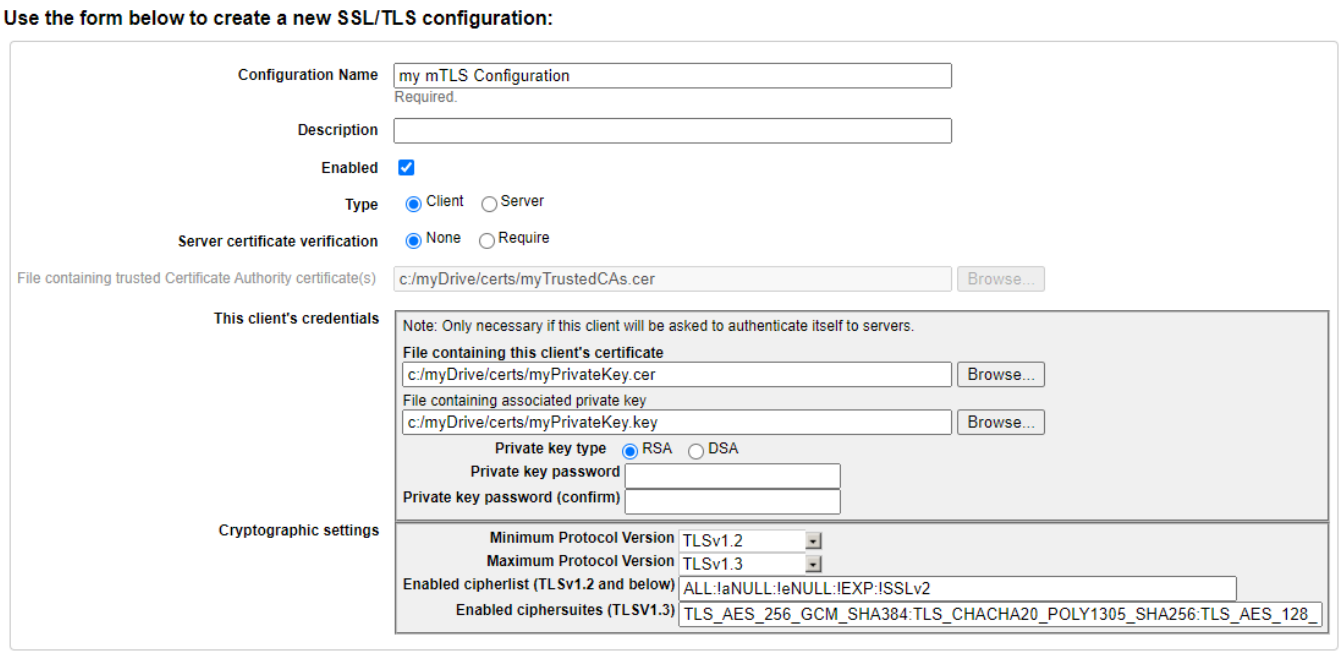
I recently needed to setup an SSL/TLS configuration in IRIS that supported mutual authentication (where the server IRIS is establish a connection to is verified, and, where IRIS is in turn verified by the remote host). After a bit of research and getting it done, I thought it worthwhile to just go over the process I went through in order to potential help others, and save you some time .

I had in hand, Certificate and Private key for my IRIS instance, that was initiating the connection, and needed to ensure that this was setup correctly. I did not have any information about the remote host, which I will basically refer to here as simply 'RemoteHost'.

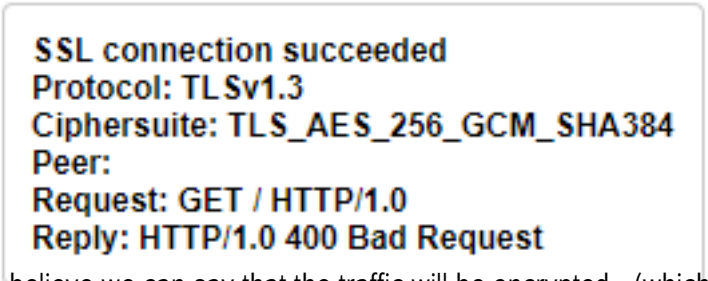
This is what I did, and how I got it implemented :

- Create an SSL/TLS configuration in IRIS via the Security menus
- For 'This client's credentials', in "File containing the client's certificate' you have to supply the path to the file that holds the Certificate for this IRIS instance which you have in hand - this must be in PEM format (if not, use **openssl tools** to convert it to the this format)
- For 'File containing associated private key', you have to supply the path to the file that holds the Private key for this IRIS instance which you have in hand - this too must be in PEM format (again, if not use **openssl** to convert it to the this format)

for Example:



- At this point, it is worth TESTing the connection with the TEST button, which if nothing else, verifies network connectivity. - A succesfull test reveals something like the following: (spoiler... 'Peer' is empty).



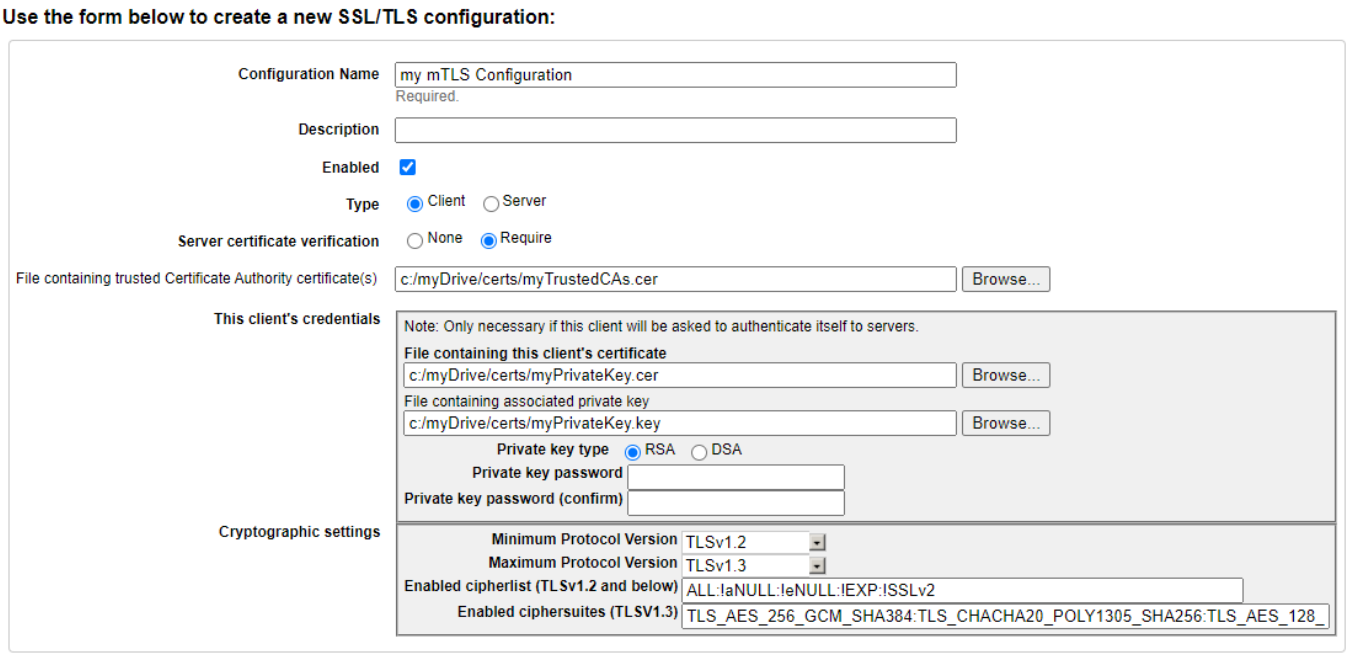
I believe we can say that the traffic will be encrypted - (which it does with the public key received from 'RemoteHost' during the SSL handshake), but, we have not verified that Remote Host is really who they say they are. Mutual authentication is **not** setup yet at this point.

Next Steps...

In the same configuration for '**Server Certificate Verification**' choose '**Require'.**

- You now need to supply a file which has the certificate (in PEM format) of the Certificate Authority (CA) responsible for issuing the IRIS key pair.
  - Note #1 There are often *intermediate* certificate authorities - and that is: multiple CA's in what is often referred to a 'chain' - for example - a Root CA issues a certificate to an Intermediary CA which in turn issues the final certificate. When you have multiple CAs, ***all*** the PEM formatted certificates need to go into this file. In my case this is called '*myTrustedCAs.cer*' as you can see in the screen shot below.
  - Note #2 The IRIS Certificate I had to work with actually contained 2 certs - the Intermediary CA's certificate, and, the IRIS certificate. I removed the intermediary CA's certificate and placed it into the myTrustedCAs.cer file, leaving only the IRIS certificate in *myPrivateKey.cer*
  - I needed to find all the CA's in the chain that issued the Intermediary's certificate, and add them too. Using **openssl** I was able to find out more about IRIS's certificate information (openssl x509 c:/myDrive/certs/myPrivateKey.cer -noout -text) - but even if you do not want to do that - opening the IRIS certificate file in Windows will allow you to see information about it, and, the chain of CA's that issued the certificate. After identifying that there was just just one other - the Root CA, and this was a specific DigiCert Root Certificate authority, I headed off to DigiCert's website where Root and Intermediary CA Certificates can be downloaded (conveniently in PEM format too). I added the Root CA's certificate to my file '*myTrustedCAs.cer*'

- So, with all the trusted Certificate Authority certificates (the whole chain) in myTrustedCAs.cer, and, my IRIS certificate and private key also supplied, my configuration looked like this:



- Time to TEST again. This time, the output from the test shows verified Subject details from the remote server

Peer: /C=AU/L=Australia/O=*Organisation-Name*/CN=*remoteHost.com.au*/DN=*remoteHost.com.au*

This clearly shows that the field '**Peer**' is no longer empty as per the previous test, and now has the remote host identified.

I found that this completed the mutual TLS setup for me. I was able to move on to consume REST Services from my endpoint successfully, having satisfied this requirement.

One final note about the "File containing Trusted Certificate Authority certificate(s)' - in my case "myTrusstedCAs.cer". You can specify the path of the file as either an absolute path (as I have) or as a path relative to the <install-dir>/mgr/ directory. In addition - on Windows and macOS, you can specify that the configuration uses the list of trusted CA certificates that the local operating system provides. To do so, specify the string %OSCertificateStore as the value of this field.

- Steve

#Security #SSL #InterSystems IRIS