Niagara 4.9 Module Signing

November 12, 2020

Q&A

1. Are the Vykon pro modules going to be signed so they can be added back into Niagara 4.9?

   Signing of the Vykon modules was started with Niagara 4.8 and should be valid for use.

2. Where does a person set module signing level requirements?

   Set the "niagara.moduleVerificationMode" property in system.properties. You can find system.properties in niagara.home/defaults or wherever your Niagara is installed - C:\Niagara\Niagara-4.9.0.198\defaults. For a JACE, use the file transfer client to transfer a modified copy of /opt/niagara/defaults/system.properties.

3. Can internal CAs be installed programmatically? If yes, how we can do that?

   CA certs can be installed using provisioning. See Help guide Doc Provisioning -> Provisioning environment -> Installing a certificate

4. Does the expiration of an installed cert on the platform cause problems in a station running a module signed by the now expired cert?

   As long as the code was signed and timestamped while the certificate was valid, the code should continue to run properly. If the code was signed after the certificate was expired, then you'll run into issues.

5. Does 4.9 support certificate file types other than .pem?

   We only support importing certificates in PEM format in 4.9. You can use tools like openssl or Keystore Explorer to help convert.

6. Can this procedure be executed in a browser without Workbench? If not, is there a roadmap for that browser feature set?

   There currently isn't a cert management interface or tools for the web interface. Typically, you would want to sign your modules as part of your CICD pipeline.

7. What about custom java code?

   If you are referring to program modules/your own developed modules, then the procedure demonstrated in the TridiumTalk would work.

8. Is there an API for automating this procedure?

   There are developer tools for automating signing code as part of the build process. See Workbench help guide Doc Developer -> Security -> Code Signing

9. Is there a place to get a list of these procedures?

   There is a step by step guide to module signing located here: https://docs.niagara-community.com/bundle/ModuleSigning/page/index.html

10. I have been sent module's along with a .sig corresponding file, what is done with that?

    The sig file typically corresponds to something that is installed as part of one of the dist files (compared to installing modules via the software manager). Sig files don't typically go with a Niagara module.

11. Can individual program objects be signed, or do they need to be contained within a ProgramModule?

    Program objects can be signed individually. There is some information on how to do this in Workbench's help page "Configuring Workbench to sign program objects"

12. Is there any support for bulk certification of many modules, many N4 instances, other than long Workbench sessions?

    The jar signer tool does not currently support signing multiple jars at a time. There are developer tools for automating signing code as part of the build process.

13. I get warnings in 4.7 that they need to be signed in a future version.

    Yes, the requirements have increased in later versions. 4.7 by default allows unsigned modules. This will not work in 4.9, which does not allow unsigned modules unless you lower the module verification mode (which is not recommended).

14. When you generate your new local certificate for those odd jars, you need to be on the internet to poke at the Cert servers, but after this if you have an "islanded project with no internet access" you should be OK correct until the expiration date for that certificate?

    The only time that you need to have internet access is when the module is being signed. When it generates the timestamp, it reaches out to a timestamp server. That isn't required once it's signed.

15. How does this process differ if you use a public CA cert?

    Once signed, if you have used a commercial CA, that CA certificate should already be in the system trust store so you shouldn't need to install the CA cert yourself. That is one of the benefits of going with a commercial CA.

16. What is the recommended process for development of a module? I currently code-compile-test on my PC and don't really want to have to go through a whole build process while I am in the midst of development.

    For development, you can either install the auto-generated self-signed certificate into your user trust store, or you can set the level to low in system.properties.

17. How will community modules be signed?

    That would really depend on the maintainers of the modules. What you can do, is use the jar signer tool in workbench to sign them if they aren't signed if you trust them.

18. Can OEMs include the Internal CAs so that it gets installed into trust store?

    Unfortunately, there currently isn't a way to do this since the CA would have to be installed in the user trust store.

19. What do I need to purchase from the public CA?

    You need a code signing certificate.

20. I set the level to low now, what about when low disappears?

    It's anticipated that we would continue to support low with dev licenses for just this purpose, or we would come up with another approach.

21. Can we use a self-signed certificate for production? is that safe?

    Yes, self-signed certificates can be used in production. It is safe as long as the private keys are sufficiently protected, and customers only install certificates that they trust.

22. Should/Can we sign that certificate with the Tridium one already installed on the local platform (as seen on the Tools > Certificate Manager)?

    No, the auto generated Tridium certificate is for TLS communication only and is not associated with code signing.

23. If using a certificate generated by an externally recognized CA, is it just a matter of installing it on the local platform to be able to sign our modules (jars) with it using: Tools > Jar Signer Tool?

    Yes, the private key and full certificate chain must be imported into the Workbench user key store in order to be used with the Jar Signer Tool.

24. Must we distribute the certificate to our clients? (public part, that is)?

    If using a self-signed or internal CA signed certificate, yes it must be distributed to customers and installed in the user trust store of all instances of Workbench/platform/JACE where the module will be used. If using a publicly trusted CA, no certificate distribution or installation is necessary.

25. Apart of Workbench Tools > Jar Signer Tool, can we use the normal signing process using gradle tools/plugins?

    Yes, this is covered in the recording and in the developer documentation available in Workbench.

26. I've read on the WB documentation that the certificate needs to be redelivered to clients once it expires (well, not the old one, but a new renewed one). Do we need to resign the modules and redeliver them too? (we could have released a new, let's say, a week before with the old certificate).

    As long as timestamping is enabled, a module that is signed while the certificate is still valid will continue to verify. When the certificate expires, only newly built modules will need to be signed with the new certificate.