

Technical Document

Niagara 4 Hardening Guide

March 7, 2023

niagara⁴

Niagara 4 Hardening Guide

Tridium, Inc.

3951 Westerre Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, and Sedona Framework are registered trademarks, and Workbench are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2023 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

Contents

- About this guide5**
 - Document change log5
 - Related documentation5
- Chapter 1 Password7**
 - Use password strength feature7
 - Change password strength7
 - Stronger passwords8
 - Enable the account lockout feature8
 - Expire passwords9
 - Configuration of password expiration9
 - Password expiration: edit user window..... 10
 - Using password history..... 11
 - Use password reset feature 12
 - Remember credentials..... 13
- Chapter 2 System passphrase.....15**
 - Change default system passphrase 15
 - Use TLS to set the system passphrase 16
 - Choose strong system passphrase 16
 - Protect the system passphrase..... 17
 - Platform owner must know system passphrase 17
- Chapter 3 Platform account management.....19**
 - Use different account for each platform user 19
 - Use unique account names for each project21
 - Platform owner must know platform credentials22
 - Station account management22
 - Use different account for each station user22
 - Use unique service type accounts for each project23
 - Disable known accounts when possible23
 - Set up temporary accounts to expire automatically24
 - Change system type account credentials.....25
 - Disallow concurrent sessions when appropriate25
- Chapter 4 Roles and permission management27**
 - Configure roles with minimum required permissions27
 - Create new categories.....27
 - Assign minimum required roles to users27
 - Use minimum possible number of superusers28
 - Require superuser permissions for program objects.....28
 - Use minimum required permissions for external accounts28
 - Authentication28
 - Use authentication scheme appropriate for account type29
 - Remove unnecessary authentication schemes..... 30
- Chapter 5 TLS & certificate management.....31**

- Enable platform TLS only 31
- Enable Fox TLS only 33
- Enable Web TLS only 34
- Enable TLS on other services 36
- Set up certificates 36
- Module installation 36
- Verify module permissions 36
- Chapter 6 Additional recommendations 39**
 - Require signed program objects and robots 39
 - Disable SSH and SFTP 39
 - Disable unnecessary services 40
 - Configure necessary services securely 41
 - Update Niagara 4 to latest release 41
 - Address needs for dual approval 41
 - Provide proper management of audit logs 41
 - Provide mechanism for generating alarm for audit processing failure 42
 - Allow only authorised management of Niagara installation 42
 - External factors 42
 - Install devices in secure location 42
 - Make sure that stations are behind a VPN 42
 - Cipher suite group settings 42
- A Additional information 45**
 - Creating strong passwords that are actually strong 45
 - Hardening checklist 45
 - Create a blacklist for sensitive files and folders 47
- Index 49**

About this guide

This topic contains information about how to implement best security practices in Niagara 4.

Following practices can be used to make Niagara 4 more secure:

- Passwords
- System paraphrase
- Platform Account Management
- Station Account Management
- Role and Permission management
- Authentication
- TLS and Certificate Management
- Module Installation
- Additional Settings
- External factors

Apart from above mentioned factors many other factors are responsible for security. Many other factors affect security- and vulnerabilities in one area can affect security in another.

Document change log

This topic summarizes the history of this document.

March 7, 2023

Replaced a screen capture in “Enable platform TLS only” and edited the rest of the chapter to promote only secure communication.

Updated screen captures, TLS version recommendations, and certificate password options in “TLS and certificate management” chapter.

September 30, 2022

Added chapter on “Cipher Suite Group settings”.

October 10, 2019

Updated for Niagara 4.7.

Related documentation

Additional information is available in the following documents.

- *Niagara Station Security Guide*
- *Niagara Platform Guide*

Chapter 1 Password

Topics covered in this chapter

- ◆ Use password strength feature
- ◆ Change password strength
- ◆ Stronger passwords
- ◆ Enable the account lockout feature
- ◆ Expire passwords
- ◆ Configuration of password expiration
- ◆ Password expiration: edit user window
- ◆ Using password history
- ◆ Use password reset feature
- ◆ Remember credentials

The Niagara 4 system typically uses passwords to authenticate “users” to a station or platform. It is particularly important to handle passwords correctly. If an attacker acquires a user’s password, they can gain access to the system and have the same permissions as that user. In the worst case, an attacker might gain access to a Super User account or platform account and the entire system could be compromised.

Here are some of the steps that you can take to help secure the passwords in a Niagara 4 system:

- Use the Password Strength Feature
- Enable the Account Lockout Feature
- Expire Passwords
- Use the Password History
- Use the Password Reset Feature
- Leave the Remember These Credentials Box Unchecked

Use password strength feature

Many of the configured authentication schemes in Niagara 4 support the notion of authenticating users with a password, but not all passwords are equally effective. Ensuring that users are choosing good, strong passwords is essential to securing a Niagara 4 system that uses password-based authentication schemes.

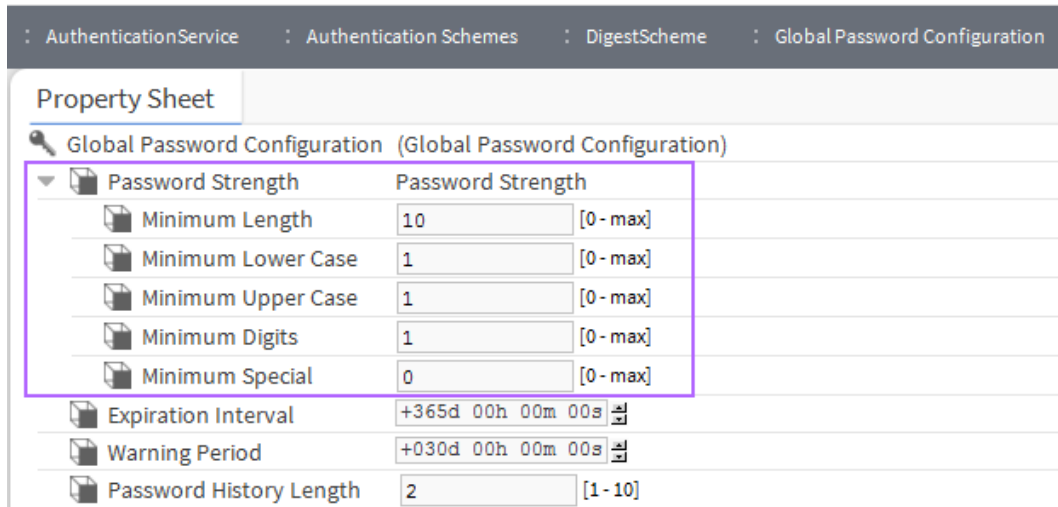
In Niagara 4, password strength is enforced by the **Password Strength** property on the authentication scheme **Global Password Configuration** property and the required password strength can be customized to meet the needs of each system. By default, passwords are required to be at least 10 characters in length, and contain at least 1 digit, 1 uppercase and 1 lowercase character. This is the recommended industry standard for most applications. However, systems with higher security requirements can configure the **Password Strength** property to require a password strength that meets their needs.

NOTE: **Password Strength** can be increased, it should not be reduced.

Change password strength

To change the required password strength, follow the steps described below:

- Step 1 Double click station's **Config** → **Services** → **Authentication Services**.
- Step 2 Expand the **Authentication schemes** folder and expand the **Authentication Scheme** that you want to change.
- Step 3 Go to **Global Password Configuration** property, expand the **Password Strength** property and edit the fields as appropriate.



Step 4 **Save** the changes.

NOTE: This does not force a user whose password no longer meets a password strength requirement to change their passwords. If that user changes their password after the password strength requirements are modified, their new password will have to meet the new requirements.

Stronger passwords

Even with good password strength requirements, there are some passwords that are stronger than others. It is important to educate users on password strength. Password strength requirements are not enough to ensure that actually strong passwords are used. For example, "Password10" satisfies all the requirements, but is actually a weak, easily hackable password. When creating a password follow the guidelines in [Creating strong passwords that are actually strong, page 45](#) to help you generate stronger passwords.

Enable the account lockout feature

The user lockout feature allows the **User Services** to lockout a user after a specified number of failed login attempts. That user is not able to log back into the station until lockout is removed. this helps protect Niagara 4 system against attackers trying to guess or brute force user's passwords.

Account Lock Out is enabled by default, but if it is not currently enabled, you can enable it as described below:

- Step 1 Expand station menu **Config**→**Services**→**UserService** property sheet.
- Step 2 Set the **Lock Out Enabled** property as `true`.

Station (serenity) : Config : Services : UserService

UserService

Display Name	Value
Lock Out Enabled	<input checked="" type="checkbox"/> true
Lock Out Period	+ 0 h 0 m 10 s
Max Bad Logins Before Lock Out	5
Lock Out Window	+ 0 h 0 m 30 s
Guest	guest
User Prototypes	User Prototypes
admin	admin

Step 3 Adjust the other lockout properties as necessary.

- **Lock Out Period:** This determines how long the user is **Lock Out** for. Even short periods (for example, 10 seconds) can be quite effective at blocking “brute force” attacks without inconveniencing users. However, more sensitive systems may warrant a longer lockout period.
- **Max Bad Logins Before Lock Out:** This determines how many login failures are required before locking out the user.
- **Lock Out Window:** The user is only locked out if the specified number of login failures occurs within the time set in the Lock Out Window. This helps separate suspicious activity (for example, 10 login failures in a few seconds) from normal usage (for example, 10 login failures over a year).

Step 4 Save the changes.

Expire passwords

In Niagara 4, user passwords can be set to expire after a specified amount of time, or on a set date. This ensures that old passwords are not kept around indefinitely. If an attacker acquires a password, it is only useful to them until the password is changed. Expiration settings are configured on authentication schemes **Global Password Configuration** property sheets as well as on individual user properties.

Configuration of password expiration

This topic describes how to configure property sheet for password expiration. Following are the steps to configure password expiration settings:

Step 1 Expand station menu **Config**→**Services**→**Authentication Service** property sheet.

Step 2 Go to the **Authentication Scheme** property sheet and find the authentication scheme which needs to be modified.

Step 3 Expand the **Global Password Configuration** property and configure the expiration settings as necessary:

- **Expiration Level:** property setting determines how long a password is used before it needs to be changed. The default is 365 days. You should change this to a lower value; ninety days is standard for many situations

NOTE: You must also set individual user password expiration dates (See **Password Expiration : Edit User** window).

- **Warning Period:** Users are notified when their password is about to expire. The Warning Period specifies how far in advance the user is notified. Fifteen days generally gives the user enough time to change their password.

Display Name	Value
Global Password Configuration	Global Password Configuration
Password Strength	Password Strength
Expiration Interval	+ 90 d 0 h 0 m 0 s
Warning Period	+ 15 d 0 h 0 m 0 s
Password History Length	2

Step 4 Save the changes.

Password expiration: edit user window

Password expiration may also be enabled on each user. If enabled on a user, the setting on the user takes precedence over the authentication scheme password expiration configuration. Once the password expires, the configuration on the user's authentication scheme is applied. This property can be configured from **userservice** window also. Following is the procedure to configure the property from **User manager view**:

- Step 1 Select the **User Manager** view from **Station**→**Config**→**Services**→**UserServices**.
- Step 2 In the **User Manager** view, select one or more users and click **Edit** to open the **Edit** window.

Name	Full Name	Enabled	Expiration	Lock Out	Roles	Allow Concurrent Sessi
rtam	River Tam	true	Never	false	Passenger	true
mreynolds	Malcolm Reynolds	true	Never	false	Captain	true

Name	rtam
Full Name	<input type="text" value="River Tam"/>
Enabled	<input checked="" type="checkbox"/> true
Expiration	<input checked="" type="radio"/> Never Expires <input type="radio"/> Expires On <input type="text" value="25-Jul-16"/> <input type="text" value="11:59"/> PM
Roles	<input type="checkbox"/> admin <input checked="" type="checkbox"/> Passenger <input type="checkbox"/> Captain <input type="checkbox"/> Crew
Allow Concurrent Sessions	<input checked="" type="checkbox"/> true
Network User	<input type="checkbox"/> false
Prototype Name	<input type="text"/>
Language	<input type="text"/>
Authentication Scheme Name	DigestScheme
Authenticator	Password Authenticator
Password	<input type="text" value="Password"/> <input type="text" value="Confirm"/>
Password Config	User Password Configuration
Password History	
Force Reset At Next Login	<input type="checkbox"/> false
Expiration	<input type="radio"/> Never Expires <input checked="" type="radio"/> Expires On <input type="text" value="31-Aug-16"/> <input type="text" value="11:59"/> PM

Step 3 Choose **Expires On** for the **Password Expiration** and set the expiration date at least 15 days into the future or perhaps to what equal to what you set for the **Password Configuration Warning Period** property.

NOTE: The default user **Password Expiration** property value is **Never Expires**. To create new users with expiring passwords enabled, set the **Password Configuration** expiration property to **Expiry On** under **Default Prototype** but be sure to set the **Expires On** date for each user.

NOTE: You could set the **Expires On** date to an arbitrary date for enough into the future that user will likely have logged into the system before expiring and also set the **Force Set at Next Login** to **true** so the user is forced to change their password at first login. This would get the their expiration in sync.

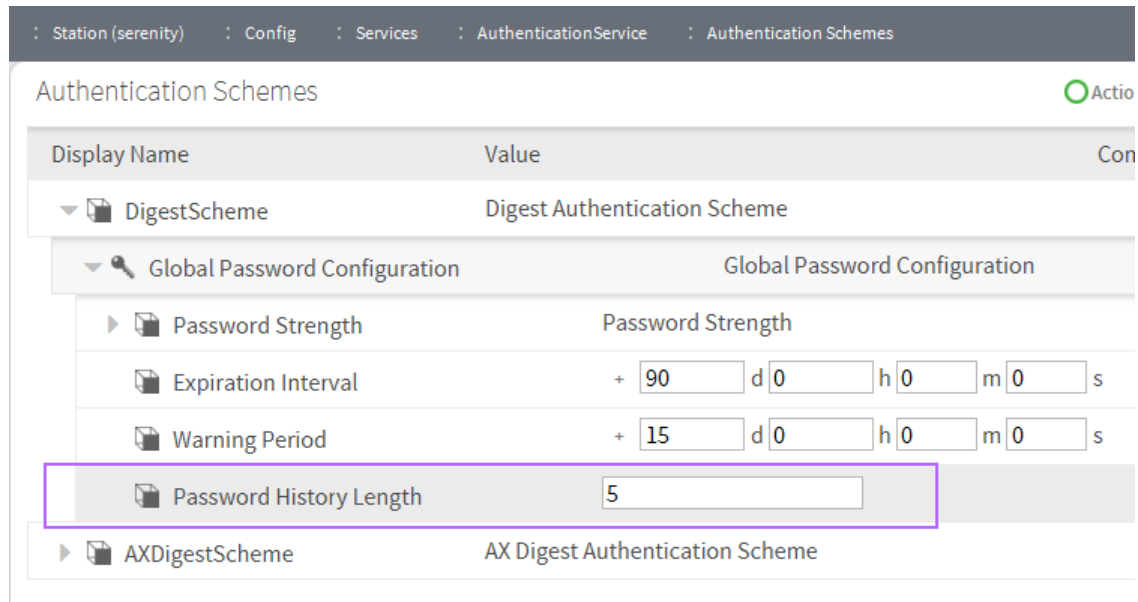
Step 4 **Save** the changes. The next time the user changes their password, the expiration date is automatically updated to the **User services Expiration Interval** added to current date and time.

Using password history

In Niagara 4, **Authentication Schemes** can be configured to remember users' previously used passwords. This password history is used to ensure that when a user changes his password, he or she does not choose a previously used password. Much like the password expiration feature, the password history helps prevent users from using passwords indefinitely. The default setting of 2 should always be changed to a reasonable number for your system. Password histories are tied to authentication schemes. Therefore, users with more sensitive accounts can have stronger **Authentication Schemes** with longer password histories.

Prerequisites:

- Step 1 Expand station and view **Config > Services > UserService→Services→Authentication Service→Authentication Schemes**.
- Step 2 Got to the Schemes folder whose password history needs to be modified.
- Step 3 Expand the **Global Password Configuration** property.

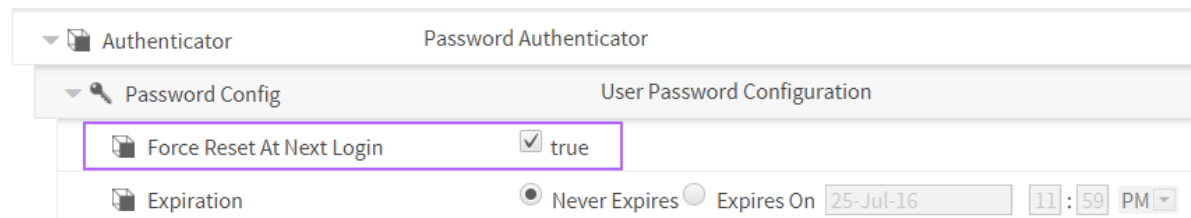


- Step 4 Set the **Password History Length** to nonzero value. This determines how many passwords are remembered. The maximum password history length is 10.

Use password reset feature

In Niagara 4, you can force users to reset their password. This is particularly useful when creating a new user. The first time a user logs in, he or she can create a brand-new password known only to that user. The password reset feature is also useful to ensure that a new password policy is enforced for all users. For example, if a station is changed to require strong passwords, the existing passwords may not conform to the password policy. Forcing users to reset their passwords will ensure that after logging in to the station, their password conforms to the rules. The following steps describe how to force a user to reset their password:

- Step 1 Go to Users property sheet view.
- Step 2 Expand the **Password Configuration** property.
- Step 3 Set the **Force Reset at Next Login** to true.



- Step 4 The next time the user logs in they will be prompted to reset their password, as shown below. The user cannot access the station until resetting the password.



To create new users with the **Force Reset at Next Login** property automatically set to `true` verify that the **Force Reset at Next Login** property is set to `true` on the **Default Prototype**.

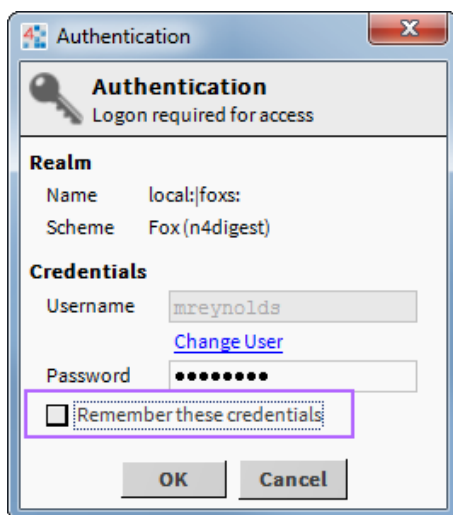
Remember credentials

When using credentials it is strongly recommended not to reuse them. For example, do not use the same credentials for an email server to which you are connected as you do for highly sensitive systems. When logging in to a Niagara 4 system via Workbench, the login dialog includes a check box to **Remember these credentials**.

When checked, Workbench will remember the credentials and use them to automatically fill in the login dialog box the next time the user tries to log in. This option is provided for convenience.

CAUTION: If the box is checked, anyone with access to that Workbench can log in using those credentials. For highly sensitive systems, privileged accounts, or unsecured computers, you should always leave the box unchecked.

In Niagara 4, there is the **Allow User Credential Caching** property on the **General** tab in the Workbench **Tools** → **Options**, which defaults to `true`. If you set that property to `false`, it will prevent a user from being able to even select the **Remember these credentials** check box in the login dialog.



Chapter 2 System passphrase

Topics covered in this chapter

- ◆ Change default system passphrase
- ◆ Use TLS to set the system passphrase
- ◆ Choose strong system passphrase
- ◆ Protect the system passphrase
- ◆ Platform owner must know system passphrase

Niagara 4 uses a system passphrase to help protect the various sensitive data in a Niagara 4 system. This can include user passwords, Kerberos keytab files, backups, etc. To protect them, the data are encrypted using the system passphrase. The system passphrase is not associated with a user; it is used by the system to encrypt files. Because the passphrase is known by a human user, the data can be moved to another unit and decrypted there, provided the new system is provided with the correct system passphrase.

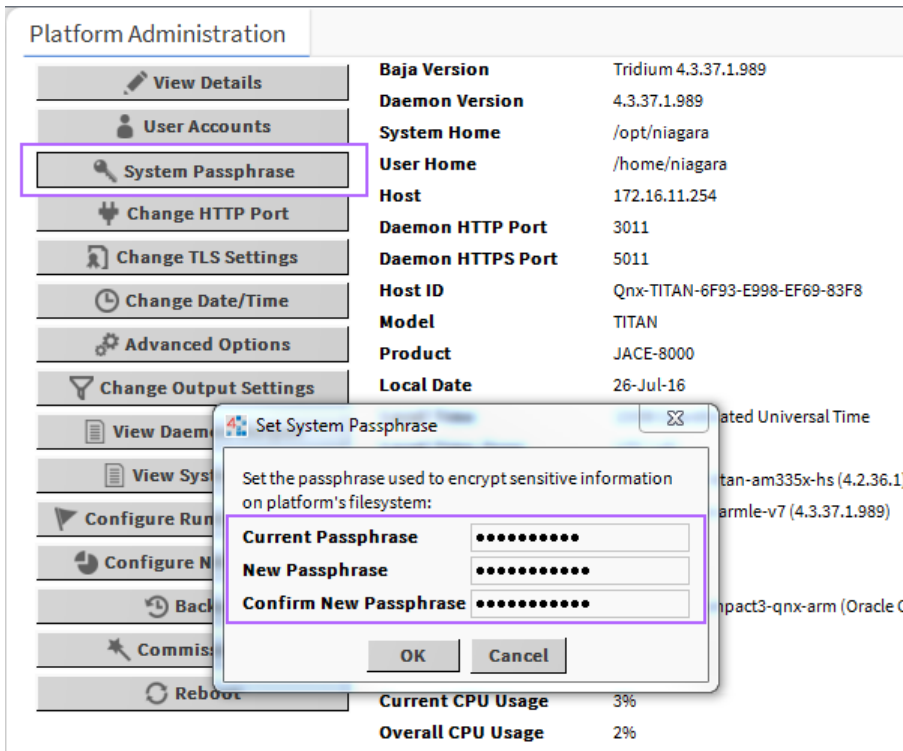
Because it is used to protect sensitive data, the system passphrase is also considered sensitive and should be protected. This section describes the various steps to take to keep your system passphrase safe.

- Change the Default System Passphrase
- Use TLS To Set the System Passphrase
- Choose a Strong System Passphrase
- Protect the System Passphrase
- Ensure Platform Owner Knows the System Passphrase

Change default system passphrase

Each JACE is shipped with a default system passphrase, Niagara. When commissioning a new JACE, you should always change the system passphrase from the default to some new, unique passphrase. Default values are typically well known and leaving the system passphrase at the default value leaves your sensitive data open to attack. Following are the steps to change the system passphrase:

- Step 1 Open the platform connection and go to **Platform Administration** view.
- Step 2 Click on **System Passphrase**.



Step 3 Enter the old system passphrase. Enter the new system passphrase and confirm. The system passphrase must contain at least 10 characters, 1 digit, 1 lower case character and 1 upper case character.

NOTE: You can easily tell if you are still using the default passphrase by going to the **Platform Administration** view. If you are using the default passphrase, a yellow warning box will be displayed in the bottom right indicating the problem.

Use TLS to set the system passphrase

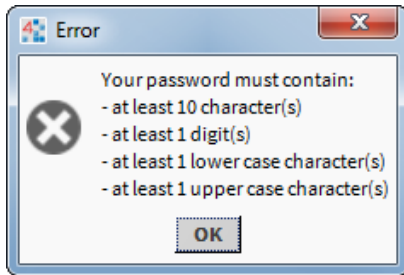
The system passphrase protects sensitive data; it must be protected. One way an attacker can attempt to acquire the system passphrase is by sniffing network traffic: although the password is sent across in encrypted format, it is sent in a clear text wrapper indicating that this is a password reset message. Using TLS adds additional protection by encrypting the whole communication - an attacker would not be able to tell which message a password is reset.

Choose strong system passphrase

The system passphrase is used to protect important data. As a result, a strong passphrase should be selected.

The system enforces the following passphrase requirements (see below):

- At least 10 characters long



- At least 1 digit
- At least 1 lower case character
- At least 1 upper case character

It is important to note that passphrase strength requirements are not sufficient to ensure that actually strong passphrases are used. See [Creating strong passwords that are actually strong, page 45](#) for guidelines on creating strong passwords.

Protect the system passphrase

In addition to picking a strong system passphrase, users should take care to protect the system passphrase. The passphrase should not be written down or placed on a sticky note on the JACE. If forgetting the passphrase is truly a concern, it should be recorded in a proper key management system or written down and locked away in a truly secure location (for example, a safe).

Platform owner must know system passphrase

When installing a Niagara 4 system, it's not uncommon for the installer to be a different person than the owner or user of the platform. For example, many people hire system integrators to set up their Niagara 4 system. In these situations, it is important that once the system integrator is done, they provide the system owner with the system passphrase. The system owner should then change the system passphrase to something known only to them.

This has several advantages:

- If something happens and a JACE can no longer be restored, a backup of the system can be restored to another device, but only if the system password is known. If the original system integrator cannot be brought back in, and the system owner does not know the password, their backups cannot be restored to a new JACE.
- The data protected by the system passphrase belongs to the system owner, and ideally should be protected by something only they know. This improves confidentiality of their data.

Chapter 3 Platform account management

Topics covered in this chapter

- ◆ Use different account for each platform user
- ◆ Use unique account names for each project
- ◆ Platform owner must know platform credentials
- ◆ Station account management
- ◆ Use different account for each station user
- ◆ Use unique service type accounts for each project
- ◆ Disable known accounts when possible
- ◆ Set up temporary accounts to expire automatically
- ◆ Change system type account credentials
- ◆ Disallow concurrent sessions when appropriate

Platform accounts are highly sensitive accounts that can allow a user to modify or bring down the system. These platform accounts must be protected to maintain the confidentiality, integrity and availability of your Niagara 4 system.

This section describes steps that can be taken to secure your platform accounts:

- Use a different account for each platform user.
- Use unique account names for each project.
- Ensure platform owner knows the platform credentials.

Use different account for each platform user

In a Niagara 4 system, multiple platform users can be created for a JACE. Each platform user account should represent a single user. Different people should never share the same account. For example, rather than a general **PlatformAdmin** user that many administrators could use, each administrator should have their own, separate account.

There are many reasons for each platform user to have their own individual account:

- If each user has their own account, audit logs will be more informative. It will be easy to determine exactly which user did what. This can help detect if an account has been compromised. In the example below, it is easy to determine which changes were made by the user JACE, and which were made by the user "TheCaptain".

```

System log for platform on 172.16.11.254
log log1 log2
0 app registry: station registry starting
8 rtc: sync system time rtc <Fri Jun 24 19:13:58 2016>, OS <Fri Jun 24 19:1
8 rtc: sync system time rtc <Tue Jun 28 19:14:03 2016>, OS <Tue Jun 28 19:1
8 rtc: sync system time rtc <Sat Jul 2 19:14:08 2016>, OS <Sat Jul 2 19:1
8 rtc: sync system time rtc <Wed Jul 6 19:14:13 2016>, OS <Wed Jul 6 19:1
8 rtc: sync system time rtc <Sun Jul 10 19:14:18 2016>, OS <Sun Jul 10 19:1
8 rtc: sync system time rtc <Thu Jul 14 19:14:22 2016>, OS <Thu Jul 14 19:1
8 rtc: sync system time rtc <Mon Jul 18 19:14:27 2016>, OS <Mon Jul 18 19:1
8 rtc: sync system time rtc <Fri Jul 22 19:14:32 2016>, OS <Fri Jul 22 19:1
16 usermgr: niagarad added user TheCaptain, rc=0
0 acctmgt: user "jace" added user "localhost/TheCaptain"
16 usermgr: niagarad add user 400 to group 20 rc=0
16 usermgr: niagarad add user 400 to group station_owners rc=0
0 acctmgt: user "jace" added user "TheCaptain" to group "niagarad_admin"
16 usermgr: niagarad changed user 400 password rc=0
0 acctmgt: password change by user "TheCaptain" for user "TheCaptain" succe
16 usermgr: niagarad changed user 400 password rc=0
0 acctmgt: password change by user "TheCaptain" for user "TheCaptain" succe
  
```

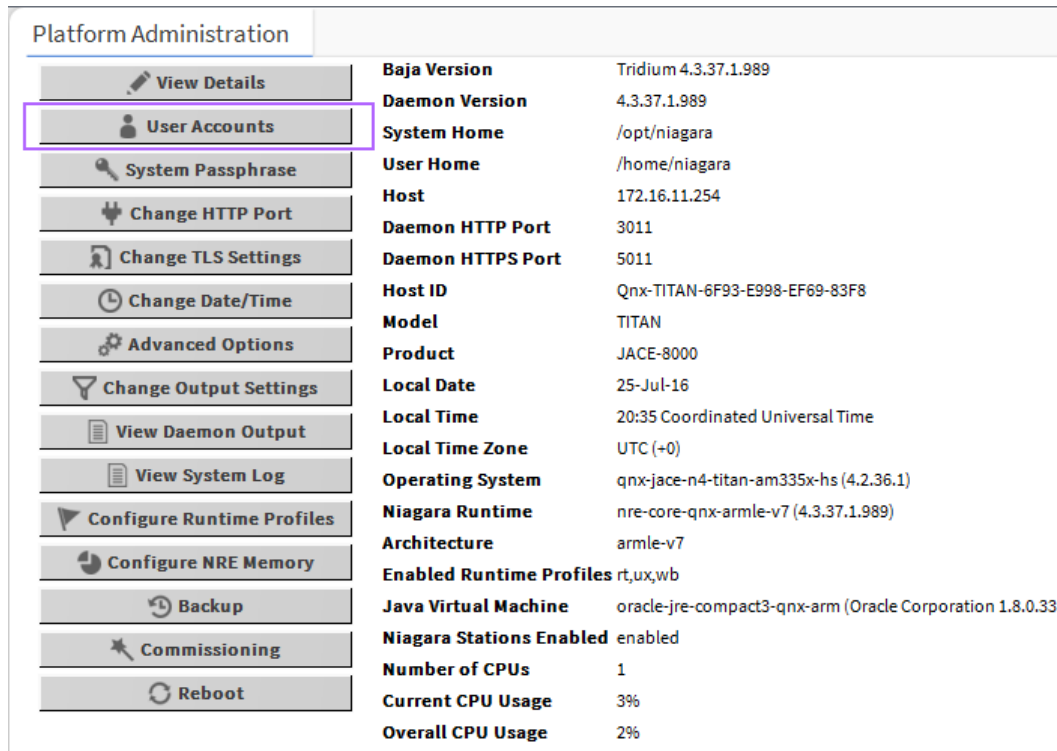
NOTE: Not all platform audit entries include the user who performed the action, but it is still a good idea to have a separate account for each user.

- If an account is removed, it does not inconvenience many users. For example, if a user should no longer have access to a station, deleting their individual account is simple. If it is a shared account, the only options are to change the password and notify all users, or to delete the account and notify all users. Leaving the account as-is is not an option – the goal is to revoke the user’s access.
- A shared account means a shared password. It is an extremely bad security practice to share passwords. It makes it much more likely for the password to be leaked and makes it more difficult to implement certain password best practices, such as password expiration. Each different user should have a unique individual account.

NOTE: Platform accounts are highly sensitive accounts. Malicious access to the platform can completely compromise the confidentiality, integrity, and availability of the platform. Therefore, you should only have a few authorized platform users, each of which should have their own unique account.

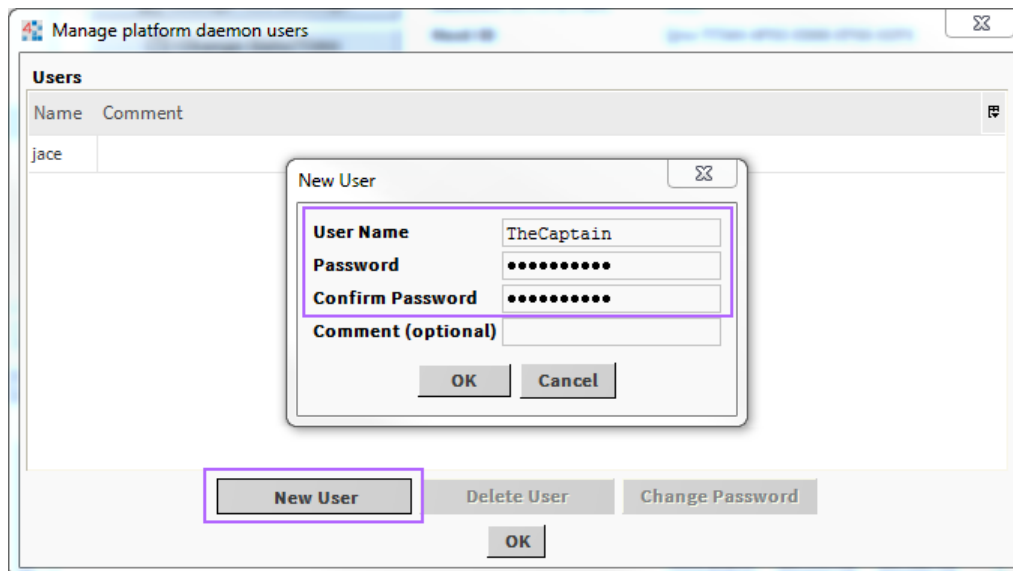
Following are the steps to create new platform account:

Step 1 Open a platform connection to a JACE and click on **User Accounts**.



Property	Value
Baja Version	Tridium 4.3.37.1.989
Daemon Version	4.3.37.1.989
System Home	/opt/niagara
User Home	/home/niagara
Host	172.16.11.254
Daemon HTTP Port	3011
Daemon HTTPS Port	5011
Host ID	Qnx-TITAN-6F93-E998-EF69-83F8
Model	TITAN
Product	JACE-8000
Local Date	25-Jul-16
Local Time	20:35 Coordinated Universal Time
Local Time Zone	UTC (+0)
Operating System	qnx-jace-n4-titan-am335x-hs (4.2.36.1)
Niagara Runtime	nre-core-qnx-armle-v7 (4.3.37.1.989)
Architecture	armle-v7
Enabled Runtime Profiles	rt,ux,wb
Java Virtual Machine	oracle-jre-compact3-qnx-arm (Oracle Corporation 1.8.0.33)
Niagara Stations Enabled	enabled
Number of CPUs	1
Current CPU Usage	3%
Overall CPU Usage	2%

Step 2 Select **New User** in the window that pops up, enter the new user's username and password. You can optionally provide a comment that will be shown in clear text in the platform user management window.



The screenshot shows the 'Manage platform daemon users' window with a 'New User' dialog box. The dialog box contains the following fields:

- User Name: TheCaptain
- Password: [Redacted]
- Confirm Password: [Redacted]
- Comment (optional): [Empty]

The 'New User' button in the main window is highlighted with a purple box.

Step 3 Click Ok.

Use unique account names for each project

It is a common (bad) practice that some system integrators often use the exact same platform credentials on every project they install. If one system is compromised, the attacker could potentially have credentials for access to many other projects installed by the same contractor.

Platform owner must know platform credentials

When installing a Niagara 4 system, it's not uncommon for the installer to be a different person than the owner or user of the platform. For example, many people hire system integrators to set up their Niagara 4 system. In these situations, it is important that once the system integrator is done, they provide the system owner with the platform credentials. The system owner should then change the platform credentials to something known only to them.

This has several advantages:

- If a platform connection is required (for example, for an update), but the original system integrator cannot be brought back in, the system owner can still perform the update, either themselves or using a new system integrator.
- The Niagara system and its data typically belong to the system owner, and ideally should be protected by something only they know. This improves confidentiality of their data.

Station account management

A Niagara 4 station has accounts, represented by users in the `UserService`. It is important that these accounts are properly managed. Failure to do so can make it easier for an attacker to penetrate the system or make it more difficult to detect that an attack has occurred.

Some steps to help correctly manage user accounts are listed below:

- Use a different account for each station user.
- Use unique service type accounts for each project.
- Disable known accounts when possible.
- Set up temporary accounts to expire automatically.
- Change system type account credentials.
- Disallow concurrent sessions when appropriate.

Use different account for each station user

Each user account in the `UserService` should represent a single user. Different people should never share the same account. For example, rather than a general "managers" user that many managers could use, each manager should have their own, separate account.

There are many reasons for each user to have their own individual account:

- If each user has their own account, audit logs will be more informative. It will be easy to determine exactly which user did what. This can help detect if an account has been compromised. In the example below, it is easy to determine which changes were made by the user "admin", and which were made by the user "mreynolds".

Station (serenity) : History : serenity : AuditHistory

Time Range 25-Jul-16 10:43 AM EDT to ?

serenity/AuditHistory

Timestamp	Operation	Target	Slot Name	Old Value	Value	User Name
25-Jul-16 10:43:07 AM EDT	Login	/Services/FoxService/se	127.0.0.1	Workbench 4.3.37.1	true	admin
25-Jul-16 11:57:32 AM EDT	Changed	/Services/Authentication	expirationInterval	365days	90days	admin
25-Jul-16 11:57:32 AM EDT	Changed	/Services/Authentication	warningPeriod	30days	15days	admin
25-Jul-16 12:48:23 PM EDT	Added	/Services/UserService/r	rtam		rtam	admin
25-Jul-16 12:48:23 PM EDT	Changed	/Services/UserService/r	password	--password--	--password--	admin
25-Jul-16 12:48:49 PM EDT	Added	/Services/RoleService	Passenger		Role	admin
25-Jul-16 12:49:07 PM EDT	Added	/Services/RoleService	Captain		Role	admin
25-Jul-16 12:49:18 PM EDT	Added	/Services/RoleService	Crew		Role	admin
25-Jul-16 12:49:31 PM EDT	Changed	/Services/UserService/r	roles		Passenger	admin
25-Jul-16 12:50:13 PM EDT	Added	/Services/UserService	mreynolds		mreynolds	admin
25-Jul-16 12:50:13 PM EDT	Changed	/Services/UserService/r	password	--password--	--password--	admin
25-Jul-16 1:03:12 PM EDT	Changed	/Services/Authentication	passwordHistoryLength	2	5	admin
25-Jul-16 1:34:42 PM EDT	Changed	/Services/UserService/r	forceResetAtNextLogin	false	true	admin
25-Jul-16 1:39:29 PM EDT	Logout (Timeout)	/Services/WebService	127.0.0.1			admin
25-Jul-16 1:39:29 PM EDT	Logout	/Services/FoxService/se	127.0.0.1	Workbench 4.3.37.1		admin
25-Jul-16 1:39:38 PM EDT	Login	/Services/FoxService/se	127.0.0.1	Workbench 4.3.37.1	true	mreynolds
25-Jul-16 1:41:35 PM EDT	Logout	/Services/FoxService/se	127.0.0.1	Workbench 4.3.37.1		mreynolds
26-Jul-16 10:46:39 AM EDT	Login	/Services/FoxService/se	127.0.0.1	Workbench 4.3.37.1	true	mreynolds

- If an account is removed, it does not inconvenience many users. For example, if a user should no longer have access to a station, deleting their individual account is simple. If it is a shared account, the only options are to change the password and notify all users, or to delete the account and notify all users. Leaving the account as-is is not an option – the goal is to revoke the user’s access.
- If each user has their own account, it is much easier to tailor permissions to precisely meet their needs. A shared account could result in users having more permissions than they should.
- A shared account means a shared password. It is an extremely bad security practice to share passwords. It makes it much more likely for the password to be leaked and makes it more difficult to implement certain password best practices, such as password expiration.

Each different user should have a unique individual account. Similarly, users should never use accounts intended for station-to-station connections. Station-to-station connections should have their own accounts.

Use unique service type accounts for each project

It is a common (bad) practice that some system integrators often use the exact same system (station to station) credentials on every project they install. If one system is compromised, the attacker could potentially have credentials for access to many other projects installed by the same contractor.

Disable known accounts when possible

In Niagara 4, it is possible to disable the default admin account. The admin account is a known account name in a Niagara 4 system. If the admin or any other known account name is enabled a potential hacker need only guess the user’s password. Note that you will not be able to disable the admin user account until you have created another super user account.

Set up temporary accounts to expire automatically

In some cases, you may need to set up an account for a user who only temporarily needs access. For example, an auditor may need an account to inspect the system. In these situations, a new account should be created and set up to expire automatically when it is no longer needed, using the **Expiration** property. This ensures that no accounts are accidentally left enabled.

Following are the steps to set up an account to expire:

- Step 1** Expand station and go to **Config**→**Services**→**User service** and create new user.
- Step 2** In **User creation** window set the **Expiration** property to the date the user will no longer require access.

Name	Full Name	Enabled	Expiration	Lock Out	Roles	Allow Concurrent Sessions
dbook	Derrial Book	true	Never	false	Passenger	true

Name	dbook
Full Name	Derrial Book
Enabled	<input checked="" type="checkbox"/> true
Expiration	<input type="radio"/> Never Expires <input checked="" type="radio"/> Expires On <input type="text" value="31-Aug-16"/> <input type="text" value="11:59"/> <input type="text" value="PM"/>
Roles	<input type="checkbox"/> admin <input checked="" type="checkbox"/> Passenger <input type="checkbox"/> Captain <input type="checkbox"/> Crew
Allow Concurrent Sessions	<input checked="" type="checkbox"/> true
Network User	<input type="checkbox"/> false
Prototype Name	

- Step 3** If the user is already created, **Edit** the **Expiration** property to the date the user will no longer require access.

Station (serenity) : Config : Services : UserService : dbook

dbook Actions & Topics

Display Name	Value
Full Name	Derrial Book
Enabled	<input checked="" type="checkbox"/> true
Expiration	<input type="radio"/> Never Expires <input checked="" type="radio"/> Expires On <input type="text" value="31-Aug-16"/> <input type="text" value="11:59"/> <input type="text" value="PM"/>
Lock Out	<input type="checkbox"/> false
Language	
Email	
Authenticator	Password Authenticator

Change system type account credentials

It may be necessary to periodically change the system type account credentials (station to station, station to rdbms, and so on). For example, if an employee who is knowledgeable of the system type credentials is terminated, you may want to change those credentials. Also, in most cases, it is better to configure a system type account with non-expiring passwords, so that those passwords expiring silently do not affect system operation.

Disallow concurrent sessions when appropriate

In Niagara 4, users can, by default, log in from multiple clients at the same time. For example, a user could be logged from two different workstations, or from two different browsers on the same workstation. However, certain accounts may be more sensitive and may require extra protection. If you know that a user will only ever be logged in from one client at a time, you can disable the ability to run concurrent sessions. If a user is logged in, and the same user logs in from a different workstation, the original session will be disconnected with a message informing the user why.

This has several advantages:

- It helps prevent sessions being left open unattended. If a user goes home and forgets to end their session at the office, it will automatically be terminated if they log in from home.
- It notifies the user of suspicious activity. If a user's session is disconnected unexpectedly, this can indicate that an unauthorized person has accessed their account. The user can quickly change their password or alert the system administrator to disable their account.

Following are the steps to disallow the concurrent session:

Step 1 In **User Manager** view double click on the user for which you wish to disallow concurrent sessions.

Step 2 The popup window opens. Set the **Allow Concurrent Sessions** property to `false`.

Name	Full Name	Enabled	Expiration	Lock Out	Roles	Allow Concurrent Session
mreynolds	Malcolm Reynolds	true	Never	false	Captain	false

Name	<input type="text" value="mreynolds"/>
Full Name	<input type="text" value="Malcolm Reynolds"/>
Enabled	<input checked="" type="checkbox"/> true
Expiration	<input checked="" type="radio"/> Never Expires <input type="radio"/> Expires On <input type="text" value="26-Jul-16"/> <input type="text" value="11"/> : <input type="text" value="59"/> <input type="text" value="PM"/>
Roles	<input type="checkbox"/> admin <input type="checkbox"/> Passenger <input checked="" type="checkbox"/> Captain <input type="checkbox"/> Crew
Allow Concurrent Sessions	<input type="checkbox"/> false
Network User	<input type="checkbox"/> false

Chapter 4 Roles and permission management

Topics covered in this chapter

- ◆ Configure roles with minimum required permissions
- ◆ Create new categories
- ◆ Assign minimum required roles to users
- ◆ Use minimum possible number of superusers
- ◆ Require superuser permissions for program objects
- ◆ Use minimum required permissions for external accounts
- ◆ Authentication
- ◆ Use authentication scheme appropriate for account type
- ◆ Remove unnecessary authentication schemes

In Niagara 4, user permissions are managed by roles and the `RoleService`. Permissions are assigned to roles, and roles can be assigned to one or more users. It is important to manage roles and permissions properly. Failure to do so can result in users having more permissions than they need, which can result in accidental or malicious security breaches

Some steps to help properly manage roles and permissions are listed below:

- Configure roles with minimum required permissions.
- Assign minimum required roles to users.
- Use the minimum possible number of superusers.
- Require superuser permissions for program objects.
- Use the minimum required permissions for external accounts.

Configure roles with minimum required permissions

When creating a new role, think about what the users who will be assigned that role needs to do in the station, and then assign the minimum permissions required to do that job. For example, by default only admin users are permitted to write to your file system. Make sure that you do not change this setting for security reasons. In addition, a user who only needs to acknowledge alarms does not need access to the `UserService` or the `WebService`. Giving non-required permissions increases the chance of a security breach. The user might inadvertently (or purposefully) change settings that they should not change. Worse, if the account is hacked, more permissions give the attacker more power.

Create new categories

In the `CategoryService`, you should create categories as needed to ensure that users have access only to what they absolutely need. For more information on setting categories and permissions, refer to the `Authorization Management` section and various subsections in the *Station Security Guide*.

Assign minimum required roles to users

Users can be assigned one or more roles. This allows you to create roles corresponding to discrete tasks (for example, Alarm Manager or Light Technician). Users should only be assigned the roles that they need to complete their required tasks. As does assigning too many permissions to a role, assigning too many roles to a user increases the chance of a security breach.

Use minimum possible number of superusers

Only assign a superuser role when necessary. A superuser is an extremely powerful account – it allows complete access to everything. A compromised superuser account can be disastrous. Only the system administrator should have accessed a superuser account.

It is a good practice for system administrators to have two accounts. One account for normal use, and the other for use in emergency situations.

Although it can be very tempting to take the easy route and create a single superuser role and assign it to each user, doing so puts your system at risk. Instead, create a set of roles that allow you to easily assign the permissions your users require.

Require superuser permissions for program objects

Program objects are special components in a Niagara 4 station that have certain special permissions granted to them (in particular the ability to run external executables).

While Program Objects are restricted to Superusers by default, it is possible to lift this restriction by editing the <niagara_home>\lib\system.properties file. To ensure that the restriction is in place, verify that the line `niagara.program.requireSuperUser=false` is commented out (using the # character) as shown below:

```
# When this line is set to false, the restriction that only
# super users can add/edit program objects and robots in a
# running station will be lifted. The default value is true,
# meaning that only super users can add/edit program objects (and robots).
#niagara.program.requireSuperUser=false
```

NOTE: Although only Superusers should be allowed to edit Program Objects, it can be acceptable for other users to invoke the Program Object's "Execute" action.

Use minimum required permissions for external accounts

Some stations use accounts for external servers – for example, a `RdbmsNetwork` with a `SqlServerDatabase` must specify a username and password for the SQL server. This account is used when connecting to the server to read from or write to the database.

NOTE: References in this section are to permissions on the external server, and not permissions on the Niagara 4 station.

These and any other external accounts should always have the minimum permissions needed for the required functionality. That way, if the station is compromised or an exploit is discovered, the external server is better protected: an attacker gaining control of an SQL administrator user could wreak havoc, reading confidential information or deleting important data; on the other hand, an attacker gaining control of a restricted user has much less power.

When configuring a Niagara 4 station, be sure to understand exactly what tasks the external account needs to be able to perform and create a user with the minimum rights and permissions required to perform those tasks.

Authentication

Niagara 4 stations have a pluggable authentication system that can support many different authentication schemes at once. These schemes determine how a client talks to the station and how the user's credentials are transmitted to the station for proof of identity. Be sure to use the strongest authentication policies to increase protection for user credentials, keeping those accounts safer from attacks.

The following steps help secure the authentication system:

- Use an authentication scheme appropriate for the account type.

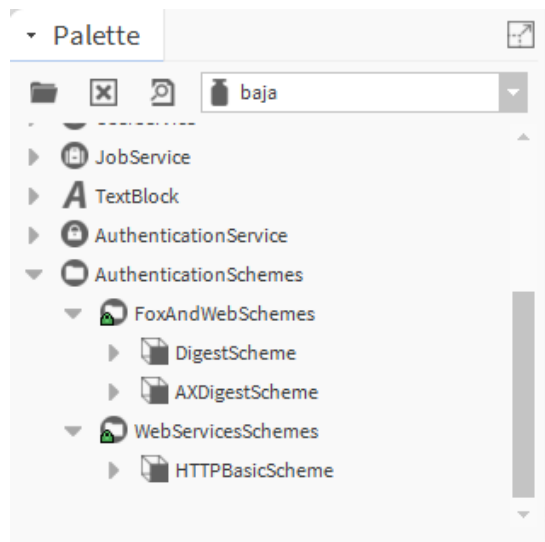
- Remove unnecessary authentication schemes.

Use authentication scheme appropriate for account type

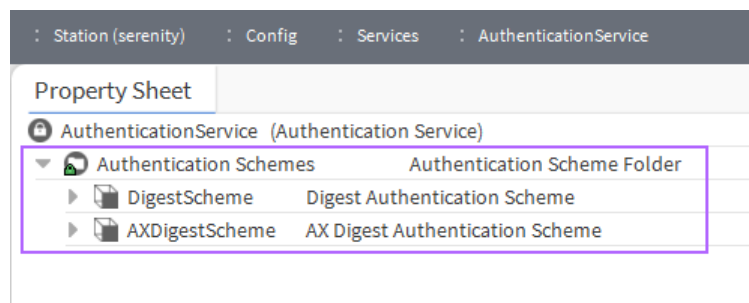
In Niagara 4, the type of authentication used is determined by the user account. Sensitive accounts should use stronger authentication types. Accounts for simple devices that can't do anything else can use less robust authentication schemes but should have roles with as few permissions as possible.

Authentication schemes can be added to the station via the **Authentication Service**, as shown below:

- Step 1 Expand station **Config**→**Services**→**Authentication Service**, and go to the **Authentication Schemes** folder.
- Step 2 Open the palette for the module that contains your authentication scheme. Niagara 4 comes with authentication schemes built in the 'baja' and 'ldap' modules.



- Step 3 Drag the **AuthenticationScheme** you want your station to support to the **Authentication Schemes** folder and configure it as appropriate.



NOTE: You can have multiple instances of the same authentication scheme type, configured differently. For example, you could have multiple **DigestAuthenticationSchemes** configured with different password strength requirements. Or, you could have different **LdapAuthenticationSchemes** pointing to different LDAP servers.

- Step 4 To configure your user account to use a authentication scheme, follow the steps below:
- Select the user you wish to configure in the **UserService UserManager** view.
 - Choose the authentication scheme you want to associate with that user from the "Authentication Scheme Name" property.

Name	Full Name	Enabled	Expiration	Lock Out	Roles	Allow Concurrent Session
mreynolds	Malcolm Reynolds	true	Never	false	Captain	false

Name

Full Name

Enabled true

Expiration Never Expires Expires On

Roles admin Passenger Captain Crew

Allow Concurrent Sessions false

Network User false

Prototype Name

Language

Authentication Scheme Name DigestScheme

Authenticator

Password

Password Config

NOTE: Certain authentication schemes (for example, **LdapAuthenticationScheme**) support the notion of remote users. For these **AuthenticationSchemes**, it is not required to create the user ahead of time. When an unknown user attempts to log in, the scheme will automatically be attempted, and the user will be created and configured on a successful login.

Remove unnecessary authentication schemes

A Niagara 4 station should only support the **Authentication schemes** that it needs. Every new **Authentication scheme** installed increases the station's attack surface: it provides a new point of entry for an attacker to attempt to exploit

For example, every Niagara 4 station comes with the Digest and **AXDigest** authentication schemes installed by default. The **AXDigestScheme** allows NiagaraAX stations to connect to a Niagara 4 station. If your station will not have NiagaraAX stations connecting to it, you should remove the **AXDigestScheme** from the **AuthenticationService**

To delete a scheme, simply delete it from the **AuthenticationSchemes** folder.

Chapter 5 TLS & certificate management

Topics covered in this chapter

- ◆ Enable platform TLS only
- ◆ Enable Fox TLS only
- ◆ Enable Web TLS only
- ◆ Enable TLS on other services
- ◆ Set up certificates
- ◆ Module installation
- ◆ Verify module permissions

Transport Layer Security (TLS) provides communication security over a network by encrypting the communication at a lower level than the actual data being communicated. This allows secure transmission of unencrypted data (for example, the username and password in LDAP authentication) over an encrypted connection. TLS as a protocol replaces its predecessor, Secure Sockets Layer (SSL); however, because TLS originally evolved from the SSL standard, the terms “TLS” and “SSL” are often used interchangeably. Although many people still refer to TLS as “SSL”, it is important to know that the latest version of SSL as a protocol (SSLv3) is not considered secure, and it is important to use the latest version of TLS available.

NOTE: In late 2014, the POODLE vulnerability was discovered in SSLv3. As a result, SSLv3 support was removed from Niagara 4.

Using TLS protects data from anyone who might be eavesdropping and watching network traffic. It also provides proof of identity, so that an attacker cannot impersonate the server to acquire sensitive data. When possible, always use TLS.

Niagara 4 provides several opportunities for using TLS. You should use these options whenever they are feasible. Niagara 4 TLS options are listed below:

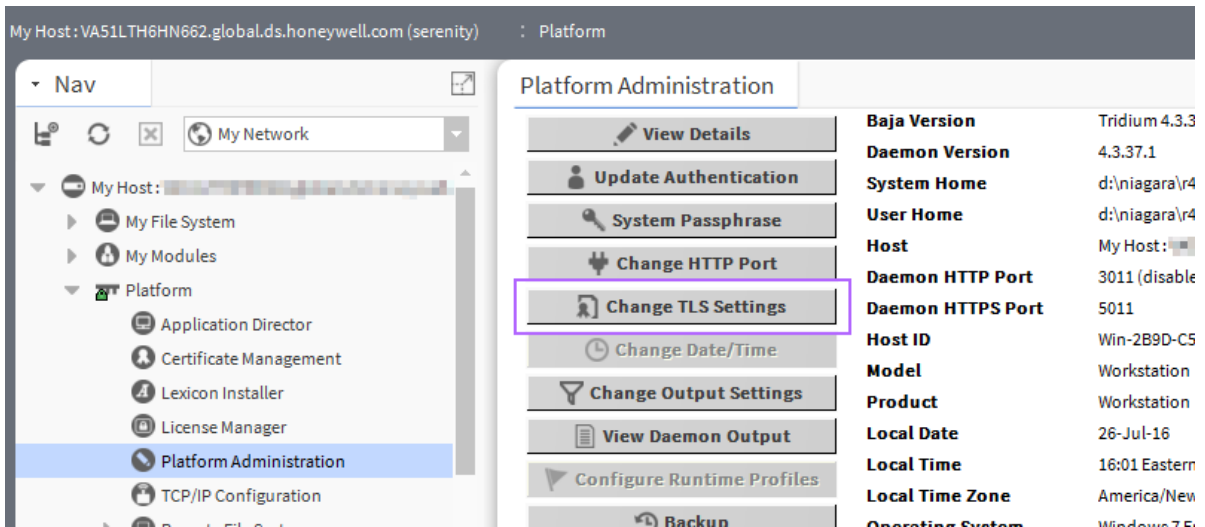
- Enable platform TLS only
- Enable Fox TLS only
- Enable Web TLS Only
- Enable TLS on other services
- Set up certificates

Enable platform TLS only

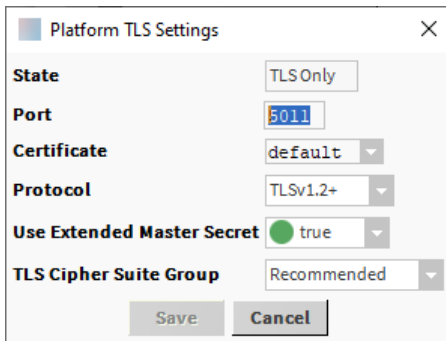
In Niagara 4, TLS can be enabled for platform connections.

Step 1 Open a platform connection.

Step 2 Navigate to the **Platform Administration** view and select **Change TLS Settings**.



Step 3 A Platform TLS Settings window opens.



The window defaults to TLS Only for State.

Step 4 Adjust the other properties if necessary.

- **Port** defaults to 5011. This is generally acceptable but may need to be changed due to IT constraints.
- **Certificate** selects the certificate to use for TLS.

NOTE: The default self-signed “default” certificate only provides encryption and does not verify server identity.

As of Niagara 4.13, for certificate protection you can set a unique private key password or select the global certificate password option. Refer to the *Niagara Station Security Guide* for information on certificates.

- **Protocol** specifies which protocols are allowed.

NOTE:

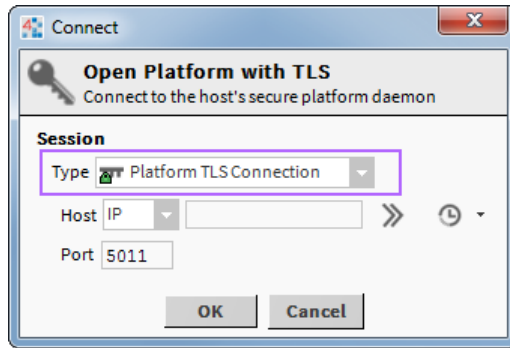
As of Niagara 4.13, TLSv1.0 and TLSv1.1 are still supported for backwards compatibility, but it is recommended to use TLSv1.2 and higher.

IT or contractual constraints may require you to pick a particular setting.

Step 5 Click **Save** and close the platform connection.

Step 6 With TLS enabled, open a platform connection to the station.

The **Open Platform with TLS** window opens.



Step 7 Under **Session**, change the **Type** property to `Platform TLS Connection`.

The system updates the window.

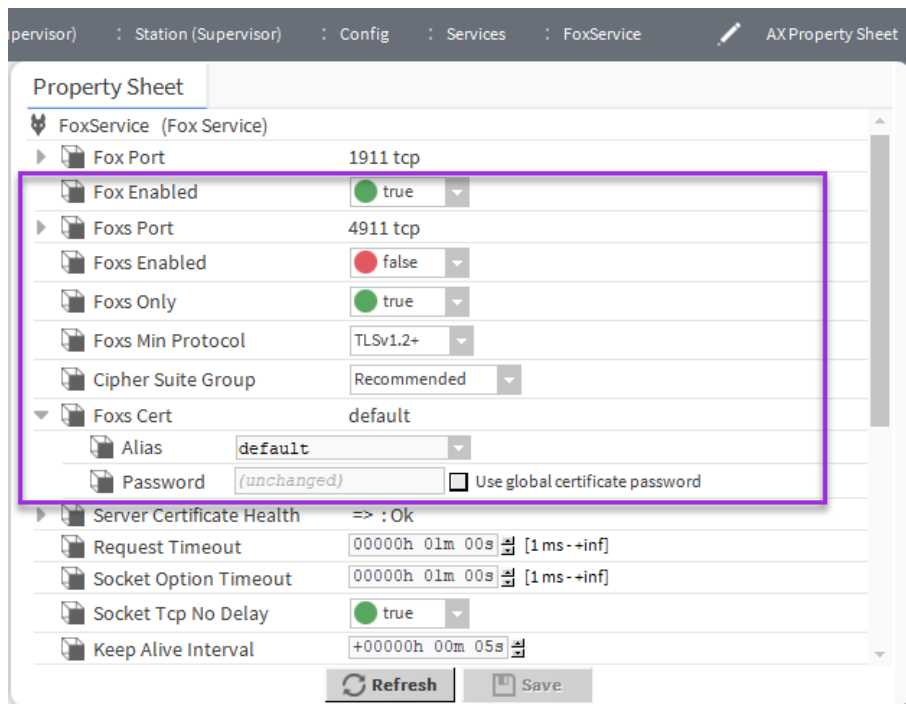
Step 8 Enter the **IP**, **Port** and credentials for the platform and click **OK**.

Enable Fox TLS only

In Niagara 4, you can enable TLS for Fox connections.

Step 1 Open a station connection, expand **Config**→**Services** and double-click **FoxService**.

The FoxService property sheet opens.



Step 2 Confirm that **FoxS Enabled** and **FoxS Only** are set to `true`.

Step 3 Adjust the other Fox settings as necessary:

- **Foxs port** defaults to 4911. This setting is generally acceptable but may need to be changed due to IT constraints.
- **Foxs Min Protocol** determines the minimum acceptable TLS version to use.

NOTE:

As of Niagara 4.13, TLSv1.0 and TLSv1.1 are still supported for backwards compatibility, but it is recommended to use TLSv1.2 and higher.

- **Foxx Cert** selects the certificate to use for TLS. As of Niagara 4.13, for certificate protection you can set a unique private key password or select the global certificate password option. Refer to the *Niagara Station Security Guide* for more information.

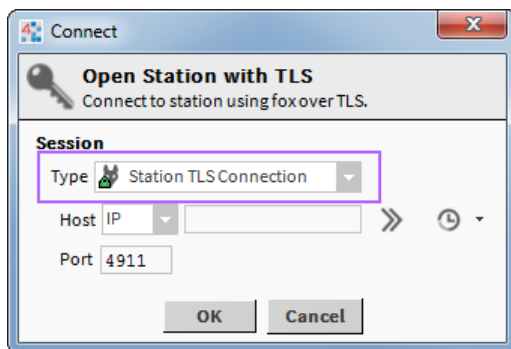
Step 4 To save your settings, click **Save** and close the station connection.

NOTE: If **Foxx only** is not set to `true`, regular fox connections (not over TLS) are permitted. Unless absolutely required, this configuration should not be allowed, because it places the burden of remembering to use TLS on the user initiating the connection. This can easily be forgotten, compromising security. Leaving **Foxx Enabled** property set to `true` with **Foxx only** also set to `true` provides a redirect to the Foxx port if a client attempts to make an Foxx connection that is not secure.

Step 5 Open a connection to the **Station**.

Step 6 Under the **Session** section, change **Type** to `Station TLS Connection`.

The system updates the window.



Step 7 Enter the **IP**, **Port** and credentials for the station and click **OK**.

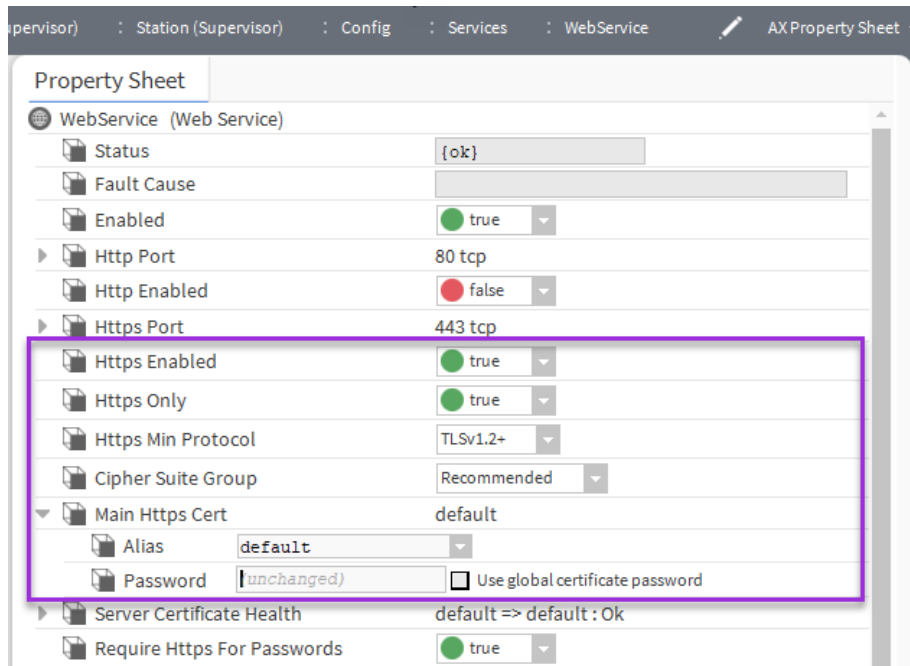
NOTE: A Foxx connection over TLS shows a tiny lock on the fox icon (🦊🔒)

Enable Web TLS only

These steps enable TLS over HTTP.

Step 1 Open a station connection, expand **Config**→**Services** and double-click **WebService**.

The **WebService** Property Sheet opens.



Step 2 Confirm that **Https Enabled** and **Https Only** are set to `true`.

Step 3 Adjust the other Https settings as necessary:

- **HTTPS Port**: defaults to 443. This generally acceptable but may need to be changed due to IT constraints.
- **TLS Min Protocol**: determines what minimum acceptable TLS version to use.

NOTE:

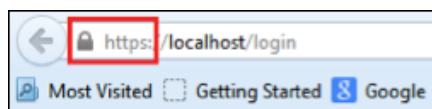
As of Niagara 4.13, TLSv1.0 and TLSv1.1 are still supported for backwards compatibility, but it is recommended to use TLSv1.2 and higher.

- **Https Cert**: selects the certificate to use for Web TLS. The default self-signed “default” certificate only provides encryption and does not verify the server. As of Niagara 4.13, for certificate protection you can set a unique private key password or select the global certificate password option. Refer to the *Niagara Station Security Guide* for more information.

Step 4 To save your settings, click **Save** and close the station connection.

NOTE: If **Https only** is not set to `true`, regular http connections (not over TLS) will still be permitted. Unless absolutely required, this should not be allowed, because it places the burden of remembering to use TLS on the user initiating the connection. This can easily be forgotten, compromising security.

Step 5 To open an HTTPS connection, open a browser and navigate to the station’s login page.



If the server’s certificate was signed by a valid CA (Certificate Authority), you probably will not see a prompt.

Step 6 If prompted, make your decision on whether to accept the certificate based on an understanding of the circumstances. Refer to the *Niagara Station Security Guide* for more information.

You now have an https connection.

Enable TLS on other services

A number Niagara 4 services communicate with an outside server. For example, the `EmailService's OutgoingAccount` and `IncomingAccounts` both contact an email server. This connection is not the same as the Fox and HTTP connections used by the client to talk to the station. TLS is handled separately for these connections. When setting up a new service on a station, check to see if it includes a TLS option. If TLS is an option, make sure that it is enabled. If needed, contact the IT department and make sure that the server the station needs to connect to supports TLS. Refer to the *Niagara Station Security Guide* and *Getting Started with Niagara* for details about setting up e-mail with TLS features.

Set up certificates

Niagara 4 includes tools to help with certificate management. Certificates are required for TLS and should be set up properly.

NOTE: Default certificates are self-signed and can only be used for encryption, not for server identity verification. As of Niagara 4.13, for certificate protection you can set a unique private key password or select the global certificate password option.

There are many things to consider when setting up certificates, and a full discussion is beyond the scope of this document. Refer to the *Niagara Station Security Guide* for more information. about correctly setting up certificates for a Niagara 4 system.

Module installation

When installing modules in Niagara 4, there are extra steps that you can take to make sure that the modules you are installing will not negatively impact the security of your Niagara 4 system.

- Verify module permissions

Verify module permissions

Niagara 4 introduced the Java Security Manager, which places restrictions on who can run which code. Many modules do not have the permissions to run code that handles sensitive data or accesses files. This helps protect Niagara 4 systems from inadvertent or malicious tampering. Starting in Niagara 4 version 4.2, modules can request additional permissions to the baseline granted to all modules. These permissions allow modules to perform certain specific tasks such as authenticating users via an authentication scheme, opening sockets, or reading system properties. When installing new modules, care should be taken to inspect what permissions these modules are requesting and make sure that they match up with the functionality the module claims. For example, a module claiming to add a new UI scheme should probably not be opening a socket to `www.super-suspicious-URL.com`.

Step 1 Go to the station or Workbench spy page.

NOTE: Modules request permissions for Workbench and stations separately, so both should be verified.

Step 2 Go to **Securityinfo**→**Policy information**.

The example below shows how the "sso-rt" module might request Authentication and network_ communication permissions to perform its Single Sign On functionality.

Module Name	Permissions Granted
sso-rt	Type AUTHENTICATION Purpose This module uses Single Sign On to authenticate users. Parameters None Risk Level ● MILD (More Info)
	Type NETWORK_COMMUNICATION Purpose This modules needs to contact the Foo Identity Provider to authenticate users. Parameters [Host: idp.foo.com Ports: 80 Type: client] Risk Level ● MODERATE (More Info)

- Step 3 Verify that the permissions granted to the module match up with its intended functionality. In particular, validate that the **Purpose** property indicates a legitimate need for the permissions.

Chapter 6 Additional recommendations

Topics covered in this chapter

- ◆ Require signed program objects and robots
- ◆ Disable SSH and SFTP
- ◆ Disable unnecessary services
- ◆ Configure necessary services securely
- ◆ Update Niagara 4 to latest release
- ◆ Address needs for dual approval
- ◆ Provide proper management of audit logs
- ◆ Provide mechanism for generating alarm for audit processing failure
- ◆ Allow only authorised management of Niagara installation
- ◆ External factors
- ◆ Install devices in secure location
- ◆ Make sure that stations are behind a VPN
- ◆ Cipher suite group settings

In addition to the settings discussed in previous sections, there are a few general recommendations and settings to configure and in order to secure a Niagara 4 system. These do not fall under a specific category like TLS or passwords but are important to security.

- Require signed program objects/robots.
- Disable SSH and SFTP.
- Disable unnecessary services.
- Configure necessary services securely.
- Update Niagara 4 to the latest release.
- Address needs for dual approval.
- Provide proper management of audit logs.
- Provide mechanism for generating an alarm for audit processing failure.
- Allow only authorized management of Niagara installation.

Require signed program objects and robots

Starting in Niagara 4.2, various components such as program objects and robots can be signed by a code signing certificate. A signed program object or robot will run only if the certificate that it was signed with is present in the Certificate Management's trust stores. Unsigned objects can always run. By default, signing is not required, but you can require program objects and robots to be signed by adding the `program.requireSigning` is set to `true` system property to your system.properties file

NOTE: In future Niagara 4 releases, program object and robot signing may be required.

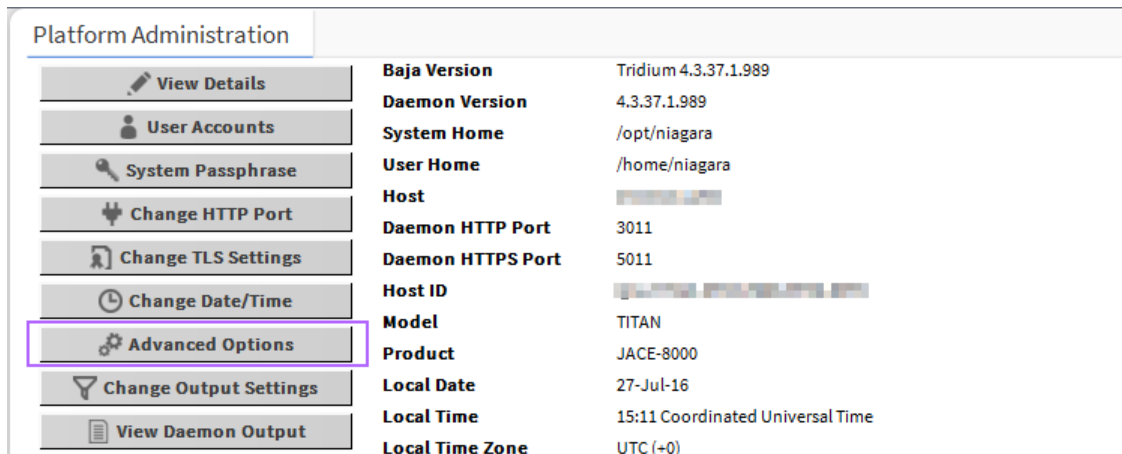
Requiring signed program objects ensures that only program objects and robots from trusted sources are allowed to run and reduces the risk of malicious code being run on your Niagara 4 system.

Disable SSH and SFTP

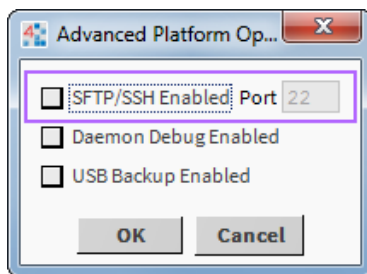
SFTP (Secure File Transfer Protocol) and SSH (Secure Shell) access to a JACE are disabled by default and should remain disabled unless necessary for troubleshooting or as directed by Tridium technical support. This helps prevent unauthorized access to the JACE. Enabling SFTP or SSH on a JACE poses a very significant security risk.

To ensure that SFTP and SSH are disabled on a JACE, follow these steps:

- Step 1 Open a platform connection to the JACE controller.
- Step 2 In the **Platform Administration** view, click on **Advanced Options**.



- Step 3 When the **Advanced Platform Option** window opens, make sure that the **SFTP/SSH Enabled** box is not selected.

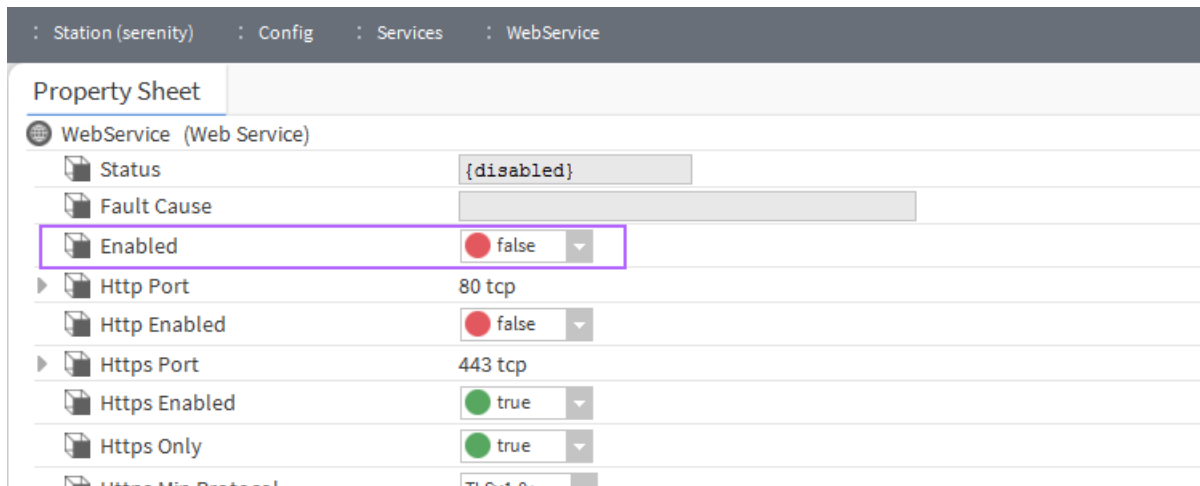


Disable unnecessary services

When setting up a Niagara 4 station, either after creating a new station or copying an existing one, many services may already be installed and enabled in the **Services** folder. However, not every station has the same requirements. Services that are not required for what the station needs to do should be removed or disabled. This helps improve security by providing fewer openings for a potential attacker to exploit.

For example, if the station is not intended to be accessed via the web, then you should disable the **WebService**. This will prevent potential attackers from using the web to attempt to penetrate the station. The same consideration should be given to the other services.

To disable a service, either remove it from the station by deleting it, or go to the service's property sheet and look for an **Enabled** property. If one exists, set it to `false`, as shown below for **WebService**.



Figuring out what services are required means planning ahead of time how the station is intended to be used. Remember, a service can always be added or enabled, so it is best to start with only the services known to be required, and add services later as necessary.

Configure necessary services securely

If a service is required, care should be taken to configure that service securely. Different services have different settings affecting security, and the relevant documentation should be consulted when configuring a new service.

For example, when configuring the `WebService`, in addition to configuring TLS, the following settings should be on considered:

- The **X Frame Options** property, set to `SameOrigin` by default for backwards compatibility, should be set to `Deny` when possible to protect against Cross Frame Scripting (XFS) attacks.
- **Show Stack Traces** should be set to `false` unless specifically debugging an issue, as a stack trace could reveal information that an attacker could use.
- The **Require Https for Passwords** property should be set to `true`. This enforces a TLS connection to perform operations such as updating a password.

Update Niagara 4 to latest release

Niagara 4 updates often include a number of important fixes, including security fixes. Niagara 4 systems should always be updated as soon as possible to ensure the best available protection. This is very important. Older releases may have known vulnerabilities – these are fixed as soon as possible, but if a system is not updated, it does not get the fixes.

Address needs for dual approval

The Niagara Framework does not currently provide a mechanism for dual approval. However, the Niagara Framework API can be employed to implement this type of functionality.

Provide proper management of audit logs

The Niagara Framework does not currently provide a mechanism for generating alarms when an audit log reaches capacity. However, the Niagara Framework API can be employed to generate an alarm or other type of warning. Best practices recommend proper maintenance of audit logs (backups and logs pushed to supervisors, and so on.)

Provide mechanism for generating alarm for audit processing failure

The Niagara Framework does not currently provide a mechanism for generating alarms when an audit processing failure occurs. However, the Niagara Framework API can be employed to generate an alarm or other type of warning, if desired.

Allow only authorised management of Niagara installation

The installation of the Niagara Framework should be done in a controlled and managed environment. Unauthorized modification could result in unexpected behavior of Niagara. Best practices recommend only authorized users be given permission to manage or modify a Niagara installation.

External factors

In addition to station and platform settings, there are some external factors to consider when securing a Niagara 4 system.

- Install JACE in a secure location.
- Make sure that stations are behind a VPN.

Install devices in secure location

Restricting physical access to JACE and Edge controllers, as well as any fieldbus device, is essential to security. If an attacker can physically connect to the device, they can gain complete control of the system. This could potentially be disastrous. Keep controllers and fieldbus devices secure in a locked room with restricted access.

CAUTION: Protect against unauthorized access by restricting physical access to the computers and devices that manage your building model. Additionally, set up user authentication with strong passwords, and secure components by controlling permissions. Failure to observe these recommended precautions could expose your network systems to unauthorized access and tampering.

Make sure that stations are behind a VPN

A station exposed to the Internet is a station at risk. Anyone who discovers the station's IP address can attempt an attack, either to gain access to the system or to bring the system down. Even stations that have been configured to use TLS only are at risk for a denial-of-service attack. Keeping stations behind a properly configured VPN ensures that they are not exposed, reducing the system's attack surface. For more information, see "Using a VPN with Niagara Systems" available from the Niagara Framework Software Security Resource Center on Niagara Community.

Do not assume that because you have not shared the station's IP address with anyone that it cannot be discovered that is not the case. There are tools that already exist to discover exposed Niagara 4 systems without knowing the IP addresses beforehand.

Cipher suite group settings

Cipher suites are sets of instructions that enable secure network connections through Transport Layer Security (TLS). These cipher suites provide a set of algorithms and protocols required to secure communications between clients and servers.

For the **Cipher Suite Group** property in the Web Service, Fox Service or Platform TLS settings, you have two options to choose from:

- Recommended: Selects the cipher suite settings that are recommended and are derived from the OWASP recommendations at the time of the release.

- **Supported:** Selects the longer list of ciphers suites that are provided for support for legacy systems, which may not have been updated.

Figure 1 TLS Cipher Suite Group property in Platform TLS Settings

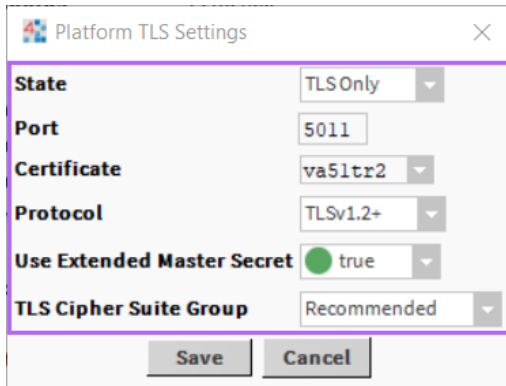


Figure 2 Cipher Suite Group property in Fox Service

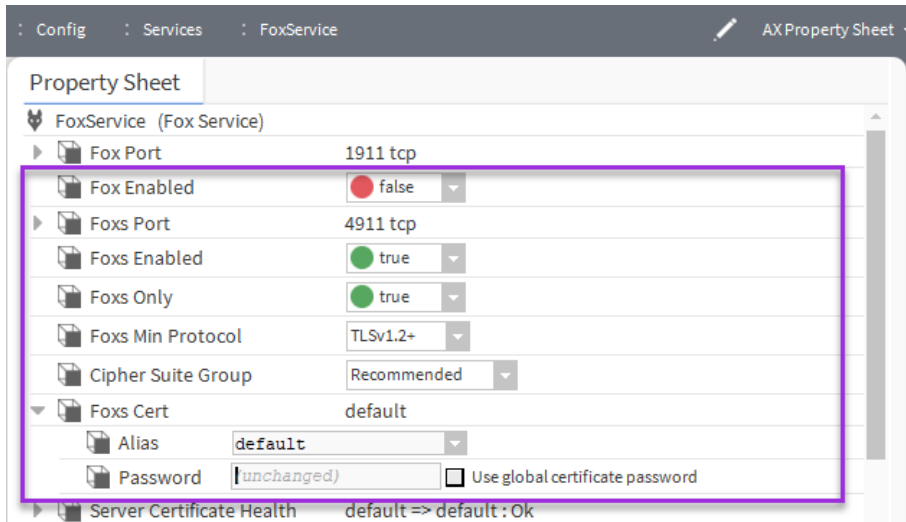
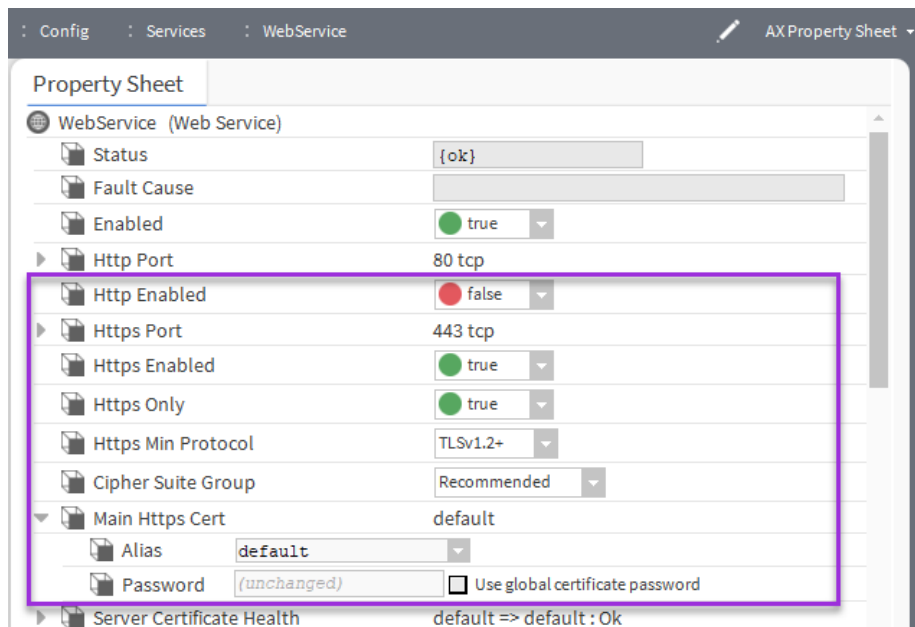


Figure 3 Cipher Suite Group property in Web Service



Excluding additional cipher suites

In some scenarios, you are confronted with regulations and policies that require additional restrictions on certain cipher suites. In those situations, administrators can set the `cipherSuite.exclude.patterns` system property to exclude additional cipher suites. This property is a comma-separated list of patterns to exclude. The patterns can be short to exclude whole types of cipher suites. For example, "ECDSA,CCM" would exclude any cipher suite containing either "ECDSA" or "CCM". Individual cipher suites can also be excluded with longer patterns. For example, "TLS_CHACHA20_POLY1305_SHA256,CCM" specifically excludes the "TLS_CHACHA20_POLY1305_SHA256" cipher suite as well as any cipher suite with "CCM" in it.

NOTE: Cipher suites can only be removed, not added. Wildcards are not supported.

A Additional information

Topics covered in this appendix

- ◆ Creating strong passwords that are actually strong
- ◆ Hardening checklist
- ◆ Create a blacklist for sensitive files and folders

The following topics include additional information for general security best practices. This includes a description of how to create strong passwords, how to use a “blacklist” for sensitive files and folders, and finally a *Hardening Checklist*.

Refer to the following sections for additional information.

Creating strong passwords that are actually strong

Most Niagara 4 systems which use passwords enforce some password strength, which may or may not be customizable. However, password strength requirements alone are not enough to ensure that a password is truly strong. A good example is “Password10”: it satisfies all the password strength requirements but is a weak password that is easy to crack. Dictionary words followed by a few numbers are an extremely common password pattern and will be quickly guessed by an attacker.

When creating passwords, the following guidelines can help generate stronger passwords:

- A random string of characters, including digits and uppercase, lowercase and special characters, (e.g. s13pj96tlcD) is typically a strong password. However, these can be hard to remember.
- A long, nonsensical sentence (e.g. “I happily tarnished under 21 waterlogged potatoes, which meet up on Sundays”) can be used as is. For systems that restrict password length, it can be contracted to include only the first character of each word (e.g. “lhtu21wp,wmuoS”). These are difficult for attackers to guess but are typically easy (albeit silly) for users to remember.

NOTE: when picking a sentence as a passphrase, it is best to avoid well-known phrases and sentences, as these may be included in dictionary attacks (e.g. “Luke, I am your father”).

- A string of random words (e.g. “coffee Strange@ Halberd 11 tortoise!”) provides a much longer password than a single word or a random string of characters. However, password crackers are becoming more aware of this technique, and inserting few random numbers and symbols in there can help. Remember, a good password is easy for a user to remember, but difficult for an attacker to guess.

Hardening checklist

This section presents the information in the Niagara 4 Hardening Guide in a convenient checklist. The list can be used to verify that all the described steps to secure your Niagara 4 system have been followed.

The checklist is included for convenience. However, it is important to remember that the goal is not to check boxes on a list. You need to have a good understanding of the security reasoning behind each of the boxes. Moreover, security is an ongoing process. You should always be on the lookout for areas in which you can improve security, whether they are on the list or not.

1. Passwords

- Use the Password Strength Feature
- Enable the Account Lockout Feature
- Expire Passwords
- Use the Password History

- Use the Password Reset Feature
 - Leave the “Remember These Credentials” Box Unchecked
2. System Passphrase
 - Change the Default System Passphrase
 - Use TLS To Set the System Passphrase
 - Choose a Strong System Passphrase
 - Protect the System Passphrase
 - Ensure Platform Owner Knows the System Passphrase
 3. Platform Account Management
 - Use a Different Account for Each Platform User
 - Use Unique Account Names for Each Project
 - Ensure Platform Owner Knows the Platform Credentials
 4. Station Account Management
 - Use a Different Account for Each Station User
 - Use Unique Service Type Accounts for Each Project
 - Disable Known Accounts When Possible
 - Set Up Temporary Accounts to Expire Automatically
 - Change System Type Account Credentials
 - Disallow Concurrent Sessions When Appropriate
 5. Role & Permission Management
 - Configure Roles with Minimum Required Permissions
 - Assign Minimum Required Roles to Users
 - Use the Minimum Possible Number of Super Users
 - Require Super User Permissions for Program Objects
 - Use the Minimum Required Permissions for External Accounts
 6. Authentication
 - Use an Authentication Scheme Appropriate for the Account Type
 - Remove Unnecessary Authentication Schemes
 7. TLS & Certificate Management
 - Enable Platform TLS Only
 - Enable Fox TLS Only
 - Enable Web TLS Only
 - Enable TLS on Other Services
 - Set Up Certificates
 8. Module Installation
 - Verify Module Permissions
 9. Additional Settings
 - Require Signed Program Objects and Robots

- Disable SSH and SFTP
- Disable Unnecessary Services
- Configure Necessary Services Securely
- Update Niagara 4 to the Latest Release

10. External Factors

- Install Devices in a Secure Location
- Make Sure that Stations Are Behind a VPN

Create a blacklist for sensitive files and folders

In Niagara, a blacklist feature is available. Many of the files listed in the blacklist are blocked by the Security Manager in Niagara 4. However, there may be cases where it is useful to blacklist additional files and/or folders.

When you implement this feature, files and folders on the blacklist are not accessible remotely through the station. This helps to protect sensitive files from being tampered with. For example, if an attacker is able to get into the station using a web connection, access to any file in the blacklist is still denied. Some folders are always blacklisted, such as the following: /backups, /bin, /daemon, /files, /jre, /modules, /registry, /security, /users and /workbench. Refer to the “system.properties notes” section in the *Platform Guide* for more details about the location of the system.properties file, blacklisting and more notes and cautions about editing the file. To edit the system.properties blacklist:

1. Open the system.properties file.
2. Uncomment the “niagara.remoteBlacklist.fileNamePatterns” line and add any file patterns that should be blacklisted (for example, *.bog).

```
# The following property allows for specification of additional
# file name patterns to blacklist from remote station access.
# File name patterns are delimited by a semicolon, and follow the format
# defined in javax.baja.util.PatternFilter. For example, a value of
# *.txt;*.xml would restrict any text or xml file from being accessed
# remotely through the station (i.e. from the web or through a fox
# connection in Workbench).
#niagara.remoteBlacklist.fileNamePatterns=*.bog
```

3. Uncomment the “niagara.remoteBlacklist.filePaths” line and add any folders that should be blacklisted (for example, /lib).

```
# The following property allows for specification of additional
# file name patterns to blacklist from remote station access.
# File name patterns are delimited by a semicolon, and follow the format
# defined in javax.baja.util.PatternFilter. For example, a value of
# *.txt;*.xml would restrict any text or xml file from being accessed
# remotely through the station (i.e. from the web or through a fox
# connection in Workbench).
#niagara.remoteBlacklist.fileNamePatterns=*.bog
```

4. The station must be restarted before changes to system.properties become effective.

The added file patterns or folders depend on the Niagara installation. Consider what needs to be protected and does not absolutely need to be accessed remotely.

Index

A

Account credentials	25
Account management	19
Additional Recommendations	39
Assign minimum required roles to users.....	27
Audit processing failure	42
Authentication.....	28
Authentication schemes.....	30
Authorised management of Niagara installation	
Allow only	42

B

Blacklist sensitive files and folders	47
---	----

C

Checklist	45
Choose strong system passphrase.....	16
cipher suite group.....	42
Configuration of password expiration.....	9
Configure necessary services securely	41
Create new categories	27
Credentials.....	13

D

Disable known accounts when possible.....	23
Disable unnecessary services.....	40
Disallow Concurrent Sessions.....	25
document change log	5
Dual Approval	41

E

Enable TLS on other services.....	36
Expire Passwords	9
External Accounts.....	28
External Factors	42

F

Fox TLS.....	33
--------------	----

I

Install Devices	42
-----------------------	----

L

LockOut Feature.....	8
----------------------	---

M

module installation	36
module permissions.....	36

N

Niagara 4	5
-----------------	---

P

password	45
Password	7
Password expiration: edit user window	10
Password history	11
Password reset	12
Password Strength.....	7
Permission.....	27
Platform account	19
Platform credentials.....	22
platform TLS.....	31
Provide proper management of audit logs	41

R

Roles and permission management	27
---------------------------------------	----

S

Set up certificates	36
SSH and SFTP.....	39
Station User	22
Stronger Password	8
Superuser Permissions	28
Superusers	28
System passphrase	16–17
System Passphrase	15, 17

T

temporary accounts.....	24
TLS & certificate management.....	31

U

Unique service type accounts.....	23
Update Niagara 4 to latest release	41
Use authentication scheme appropriate for	
account type.....	29
Use unique account names for each project	21

V

VPN42

W

Web TLS.....34