Niagara^{AX} Hardening Guide

Tips to Secure a Niagara^{AX} System

July 2013

Table of Contents

Introduction

This document describes how to implement security best practices in a Niagara^{AX} system. While it is impossible to make any system completely impenetrable, there are many ways to build up a system that is more resistant to attacks. In particular, this document describes how you can help make a Niagara^{AX} system more secure by carefully configuring and using:

- Passwords
- Accounts and Permissions
- Authentication
- TLS/SSL and Certificate Management
- Other Settings and External Factors

Please note that while all of these steps should be taken to protect your Niagara^{AX} system, they do not constitute a magic formula. Many factors affect security – and vulnerabilities in one area can affect security in another; it doesn't mean much to configure a system expertly if your JACE is left physically unsecured where anyone can access it.

Passwords

The Niagara^{AX} system 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 Niagara^{AX} system and will 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 can be taken to help secure the passwords in a Niagara^{AX} system:

- Change the Default Platform Credentials
- Use Strong Passwords
- Enable the Account Lockout Feature
- Expire Passwords
- Use the Password History
- Use the Password Reset Feature
- Leave the "Remember These Credentials" Box Unchecked

Change the Default Platform Credentials

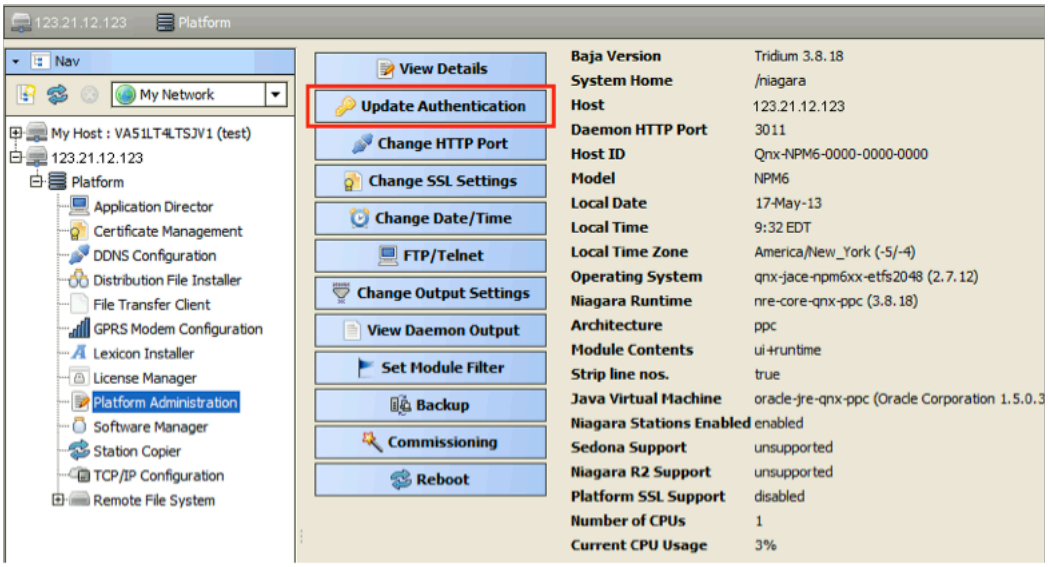
Many controllers are shipped with the following default platform daemon credentials:

Username: tridium **Password: niagara**

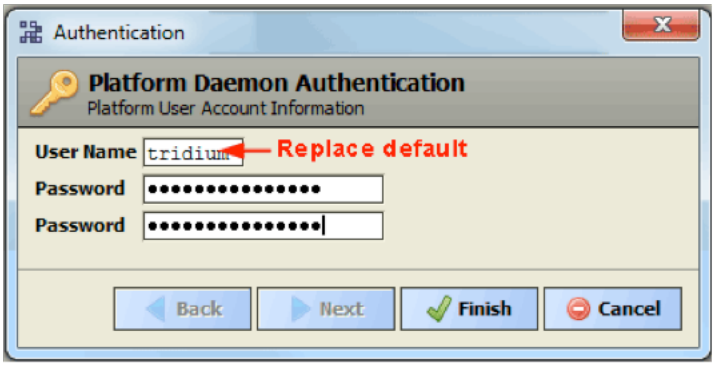
These default credentials are the same for all JACEs, and make it easy to set up a new JACE and quickly connect to it. However, these credentials provide the highest level of access. If left unchanged, anyone who knows the default credentials could potentially make a platform connection and gain complete control of the controller. It is essential to change the default credentials as soon as possible, before exposing the JACE to an open network.

Use the following steps to change the platform credentials:

1. *Open a platform connection to the JACE and go to the Platform Administration view.*



2. *Click the "Update Authentication" button. An authentication dialog box will pop up.*



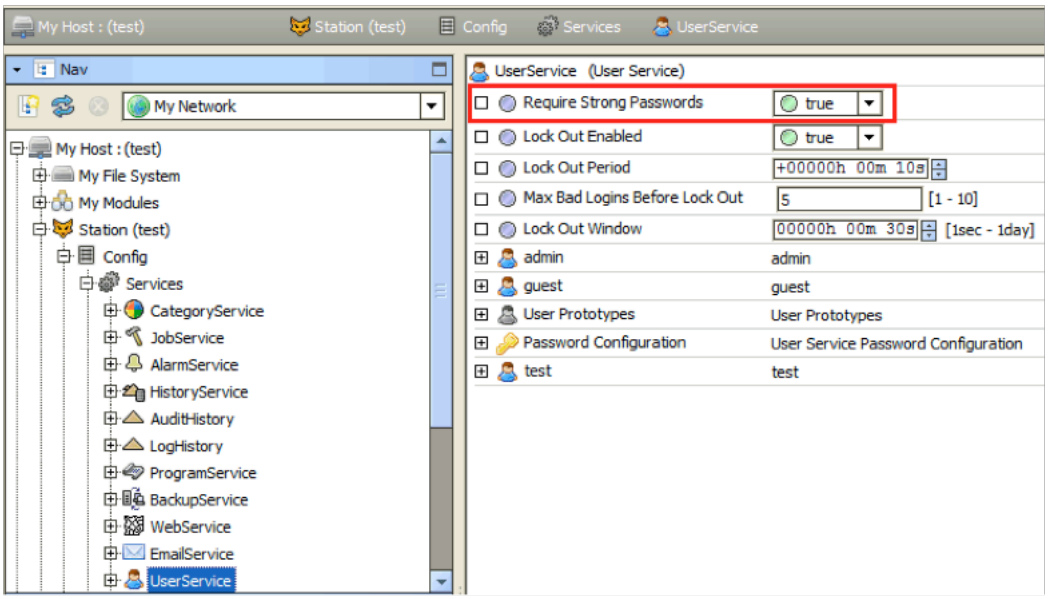
3. *Enter the new credentials, and click **"Finish."** The platform daemon credentials are updated. Note that the **"Finish"** button is available only if the two Password fields match.*

Use Strong Passwords

Not all passwords are equally effective. Ensuring that users are choosing good, strong passwords is essential to securing a Niagara^{AX} system. Niagara^{AX}-3.5u4, 3.6u4 and 3.7u1 *require* strong passwords, by default. However, if you are using a Niagara^{AX} version that does not require strong passwords by default, you should change the default setting, as described below.

Configure the Niagara^{AX} UserService to require strong passwords as follows:

1. *Go to the station's UserService's property sheet.*
2. *Set the "Require Strong Passwords" property to "true."*



3. *Save the changes.*

***NOTE:** This does not force a user to change weak passwords to strong passwords. If a user **changes** their password after this property is set to true, they will be required to use a strong password. Also, in older systems it may be necessary to task users with changing their own passwords or have a system administrator change all user passwords.*

Strong Passwords

In Niagara^{AX}-3.7 or newer systems, user accounts can be flagged to require the user to change their password on their next login. When "Require Strong Passwords" is set to true, users are required to choose passwords that are **at least eight characters that are not ALL characters or ALL digits.**

Stronger Passwords

Even with the "Required Strong Passwords" option enabled, there are some passwords that are stronger than others. It is important to educate users on password strength. A single word, followed by a number (for example, "password1), or a birthday (for example, "May151970) may be

(continued on page 6)

Use Strong Passwords (continued)

easy to remember, but it is also easy for an attacker to guess. A random string of characters (for example, “s13pj96t!cd”), or a few words strung together in a nonsensical sentence (such as, “coffee Strange Halberd 11 tortoise”) are much stronger and more difficult to guess.

Examples:

- !2345678 passes as a strong password, but doesn’t contain any alpha characters.
- abcdefg! passes as a strong password, but it doesn’t contain any numeric characters.
- abcd1234 passes as a strong password, but it doesn’t contain an special characters.

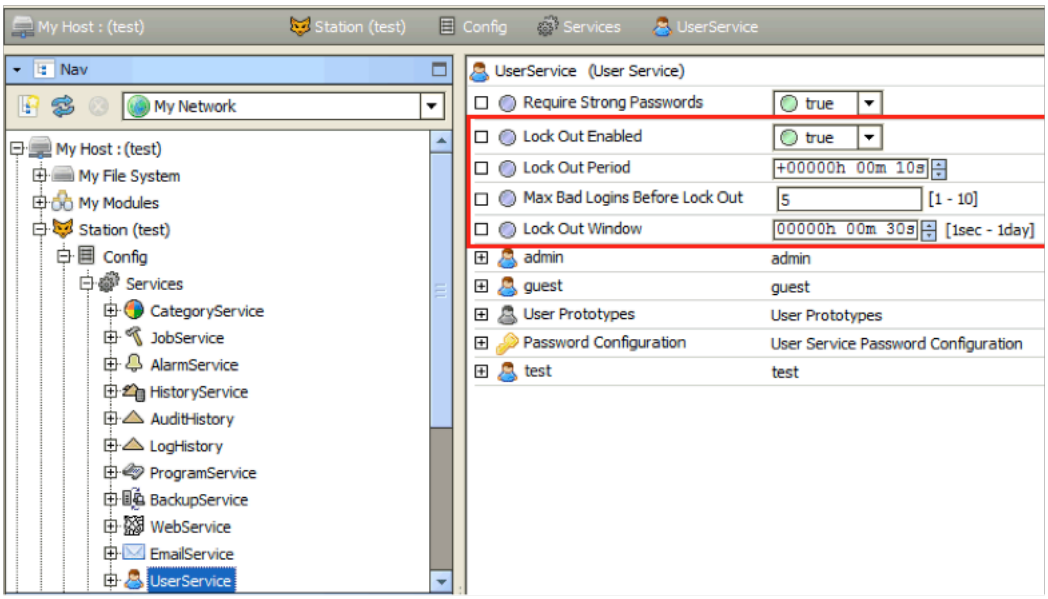
Remember, a good password is easy for a user to remember, but difficult for an attacker to guess.

Enable the Account Lockout Feature

The user lockout feature allows the UserService to lock out a user after a specified number of failed login attempts. That user is not able to log back in to the station until the lockout is removed. This helps protect the Niagara^{AX} system against attackers trying to guess or “brute force” users’ passwords.

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

1. *Go to the station’s UserService’s property sheet.*
2. *Set the “Lock Out Enabled” property to true.*



(continued on page 7)

Enable the Account Lockout Feature (continued)

3 *Adjust the other lockout properties as necessary.*

- ***Lock Out Period.*** *This determines how long the user is locked 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).*

4 *Save the changes.*

Expire Passwords

Starting in Niagara^{AX}-3.7, 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. You need to make sure that expiration settings are configured on the Password Configuration property sheet Password Configuration as well as on individual user properties.

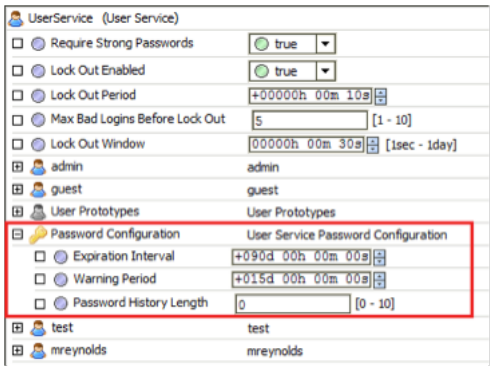
Password Expiration: Password Configuration Property Sheet

The general password expiration settings are configured in the UserService property sheet, as described below:

1 *Go to the UserService's property sheet and expand the "Password Configuration" property.*

2 *Configure the expiration settings as necessary.*

- ***Expiration Interval.*** *This 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 Users dialog box).*
- ***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.*



3 *Save the changes.*

(continued on page 8)

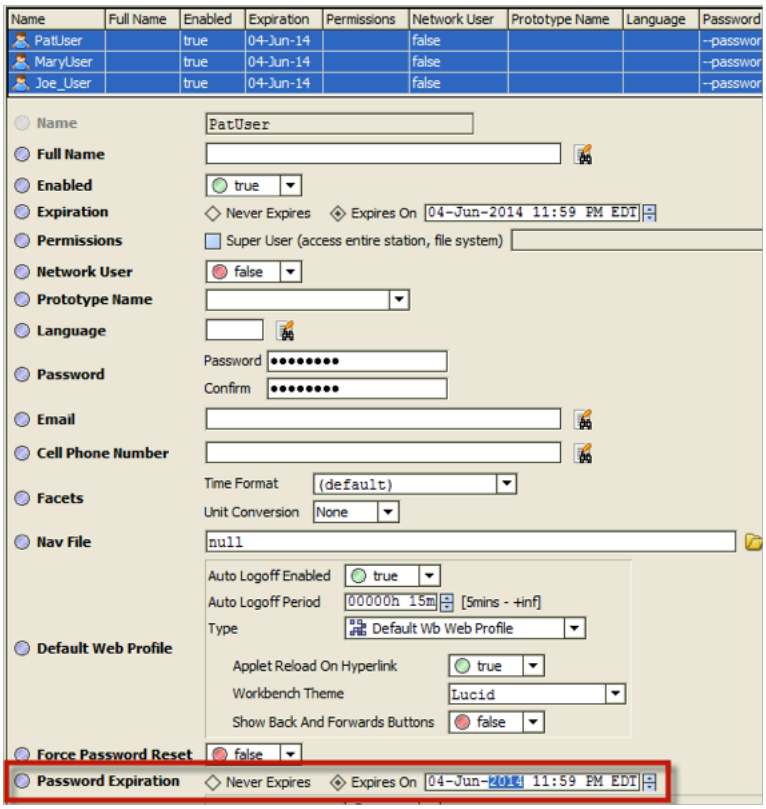
Expire Passwords (continued)

Password Expiration: Edit Users Dialog Box

Password expiration **must also be enabled on each user.** This property is also available, by user, from the UserService property sheet but it may be more conveniently configured from the User Manager view, as described below:

To enable user password expiration, do the following:

1. *Select the User Manager view on the UserService (Station > Config > Services > UserService).*
2. *In the User Manager view, select one or more users and click the **Edit** button to open the Edit dialog box.*



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

(continued on page 9)

Expire Passwords (continued)

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 (UserService > User Prototypes > Default Prototype > Password Configuration) to "Expires On" under the "Default Prototype" but be sure to actually set the "Expires On" date for each user.*
- *You could set the "Expires On" date to an arbitrary date far enough into the future that the user will likely have logged into the system before expiring and also set the "Force Reset At Next Login" to true so the user is forced to change their password on first login. This would then get their expiration in sync.*

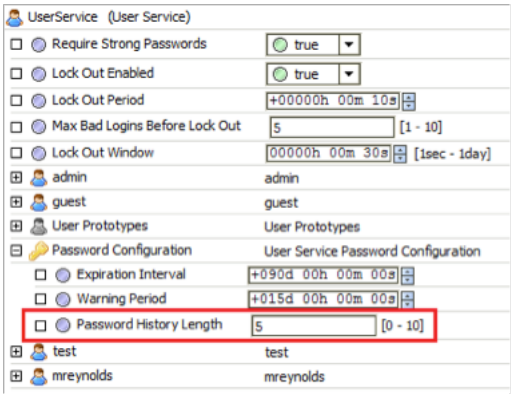
4. *Save the changes. The next time the user changes their password, the expiration date is automatically updated to the UserService's "Expiration Interval" added to current date and time.*

Use the Password History

Starting in Niagara^{AX}-3.7, the UserService can be configured to remember users' previously used passwords. This password history is used to ensure than 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 "0" should always be changed to a reasonable number for your system.

To enable password history, do the following:

1. *Go to the UserService's property sheet.*
2. *Expand the "Password Configuration" property.*
3. *Set the "Password History Length" property to a non-zero value. This determines how many passwords are remembered. The maximum password history length is 10.*

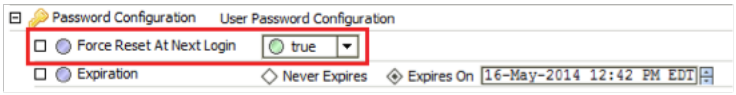


Use the Password Reset Feature

Starting in Niagara^{AX}-3.7, 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:

1. *Go to the user's property sheet view.*
2. *Expand the "Password Configuration" property.*
3. *Set the "Force Reset At Next Login" property to "True."*



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."

Leave the "Remember These Credentials" Box Unchecked

When logging in to a Niagara^{AX} system via workbench, the login dialog includes a checkbox 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. However, it is important to be aware that, if the box is checked, anyone with access to that workbench is able to log in using those credentials. For highly sensitive systems, privileged accounts, or unsecure computers, you should always leave the box unchecked.

NOTE: *Starting in Niagara^{AX}-3.7, there is a "Allow User Credential Caching" property on the "General" tab in the Workbench Options dialog box (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.*

Account Management and Permissions

A Niagara^{AX} 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 User
- Use Unique Service Type Accounts for Each Project
- Disable Known Accounts When Possible
- Change System Type Account Credentials
- Assign the Minimum Required Permissions
- Use Minimum Possible Number of Super Users
- Require Super User Permissions for Program Objects
- Use the Minimum Required Permissions for External Accounts

Use a Different Account for Each 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.”*

test/AuditHistory 33r

Timestamp	Operation	Target	Slot Name	Old Value	Value	User Name
10-May-13	Invoked	history:/test/AuditHistory	Clear	12 records	0 records	admin
10-May-13	Login Failure	/Services/WebService	127.0.0.1		1	admin
10-May-13	Logout	/Drivers/NiagaraNetwork/foxService/serverConnections/Session4	127.0.0.1			admin
16-May-13	Login Failure	/Drivers/NiagaraNetwork/foxService/serverConnections/Session1	127.0.0.1		1	admin
16-May-13	Login Failure	/Drivers/NiagaraNetwork/foxService/serverConnections/Session2	127.0.0.1		2	admin
16-May-13	Login	/Drivers/NiagaraNetwork/foxService/serverConnections/Session3	127.0.0.1			admin
16-May-13	Changed	/Services/UserService	requireStrongPasswords	false	true	admin
16-May-13	Logout	/Drivers/NiagaraNetwork/foxService/serverConnections/Session3	127.0.0.1			admin
16-May-13	Login	/Drivers/NiagaraNetwork/foxService/serverConnections/Session1	127.0.0.1			admin
16-May-13	Added	/Services/UserService	mreynolds		mreynolds	admin
16-May-13	Changed	/Services/UserService/mreynolds/baja_UserPasswordConfiguration	forceResetAtNextLogin	false	true	admin
16-May-13	Changed	/Services/UserService/mreynolds	password	--password--	--password--	admin
16-May-13	Logout	/Drivers/NiagaraNetwork/foxService/serverConnections/Session1	127.0.0.1			admin
16-May-13	Login	/Drivers/NiagaraNetwork/foxService/serverConnections/Session9	127.0.0.1			mreynolds
16-May-13	Changed	/Services/WebService	httpsEnabled	false	true	mreynolds
16-May-13	Changed	/Services/WebService	httpsOnly	false	true	mreynolds
16-May-13	Logout	/Drivers/NiagaraNetwork/foxService/serverConnections/Session9	127.0.0.1			mreynolds
16-May-13	Login Failure	/Drivers/NiagaraNetwork/foxService/serverConnections/Session11	127.0.0.1		1	admin
16-May-13	Login	/Drivers/NiagaraNetwork/foxService/serverConnections/Session13	127.0.0.1			admin
16-May-13	Changed	/Services/UserService/mreynolds/baja_UserPasswordConfiguration	forceResetAtNextLogin	false	true	admin

- *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 platform credentials and 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

Starting in 3.7 it is possible to disable the admin account. The admin account is a known account name in a Niagara^{AX} system. If the admin, or any other known account name for that matter, is enabled a potential hacker need only guess the user's password. However, do not disable the admin user account before you make sure that there is another super user account available. Also, you should keep the "Guest" account disabled.

Change System Type Account Credentials

It may be necessary to periodically change the system type account credentials (station to station, station to rdbms, etc). 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.

Assign the Minimum Required Permissions

When creating a new user, think about what the user needs to do in the station, and then assign the minimum permissions required to do that job. For example, 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 Category Service, you should create categories as needed to ensure that users have access only to what they absolutely need.

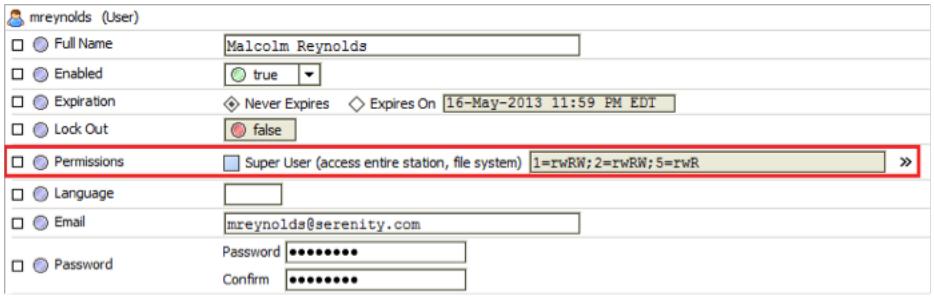
Consider the following example of how you can minimize the risk of using the Niagara^{AX} TunnelingService:

> *If a station is using the TunnelService (if not used, disable it!) this can increase security risk because Station login from the Niagara client-side applications (serial tunnel and Lon tunnel) uses "basic authentication." This means that the password is transmitted in the clear. You can minimize this risk by using a special category to which you assign only the TunnelService component. Then create a new station user, and assign permissions only on that one category (admin write). By using only this (new) user for login from the serial tunnel or Lon tunnel client application, tunneling is allowed, but with minimal impact if credentials are compromised.*

For more information on setting categories and permissions, refer to the "Security model overview" section and various subsections in the *Niagara^{AX} User Guide.*

Use Minimum Possible Number of Super Users

Only assign Super User permissions when absolutely necessary. A Super User is an extremely powerful account – it allows complete access to everything. A compromised Super User account can be disastrous. Only the system administrator should have access to the account.



Although it can be very tempting to take the easy route and check the Super User box for each user, doing so puts your system at risk. Instead, create users with the appropriate permissions.

Require Super User Permissions for Program Objects

Program Objects are extremely powerful components in a Niagara[AX] station. Similar to Super User permissions, they can give a user complete control over a station, and should only be editable by the Super User. Allowing any other user to edit Program Objects defeats the purpose of setting permissions.

> *While Program Objects are restricted to Super Users 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 Super Users should be allowed to edit Program Objects, it can be acceptable for other users to invoke the Program Object's "Execute" action.*

Use the Minimum Required Permissions for External Accounts

Some stations use accounts for external servers – for example, an 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[AX] 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[AX] 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^{AX} stations offer several authentication policy options. These options determine how a client talks to the station and how the user's password is transmitted to the station for proof of identity. Be sure to use the strongest authentication policies to increase protection for user passwords, keeping those accounts safer from attacks.

The following steps ensure the strongest authentication is used.

- Use "Digest" Authentication in the FoxService
- Set the FoxService Legacy Authentication to "Strict"
- Use "cookie-digest" Authentication in the WebService

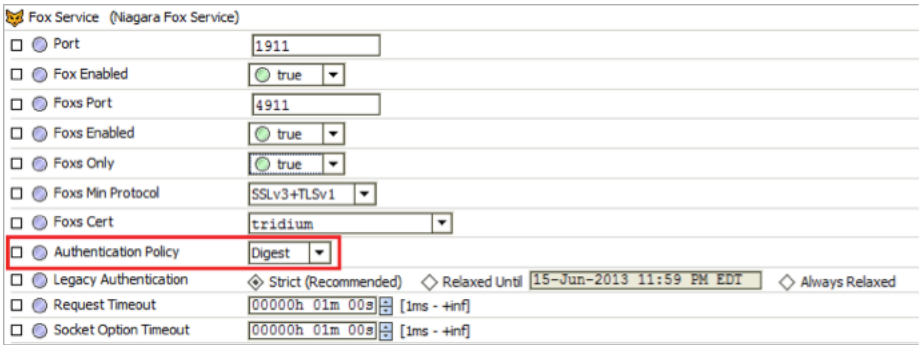
Use "Digest" Authentication in the FoxService

The FoxService allows you to set two different authentication policies. With **basic** authentication, the user's password is transmitted to the station in the clear, without any kind of processing. With **digest** authentication, the user's password is specially processed and is never actually sent to the station. In addition, the station must provide proof of its own identity before authentication is successful.

Unless otherwise required, the authentication policy should *always* be set to Digest. One exception to this is use of the LdapUserService, which cannot do digest authentication for LDAP users.

To set the authentication policy to Digest, do the following:

1. *Open the Drivers > NiagaraNetwork > Fox Service's property sheet.*
2. *Set the "Authentication Policy" property to "Digest."*



3. *Save the changes.*

***NOTE:** In later releases of Niagara^{AX}-3.5, 3.6, and 3.7, Digest is the default setting.*

Set the FoxService Legacy Authentication to "Strict"

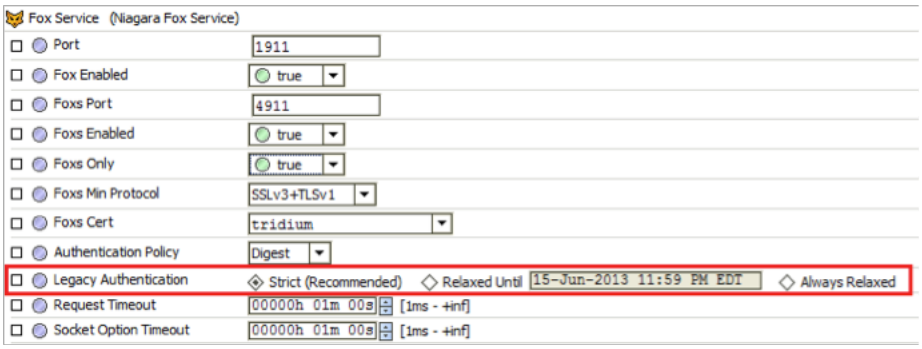
Starting in Niagara[AX]-3.7u1 (Update 1), 3.6u4 (Update 4), and 3.5u4 (Update 4), the digest authentication policy has been improved. However, because of these changes, a newer station cannot authenticate clients attempting to use the older digest MD5 algorithm.

To enable stations with digest authentication to talk to older clients, the "Legacy Authentication" property has been added to the FoxService. If set to "Always Relaxed", or "Relaxed Until", the station will degrade authentication to basic to allow the older client to authenticate. If the client can do the new digest, the new digest is used. If the client cannot do the new digest, basic is used. It only degrades for older clients. In the "Relaxed Until" case, it only allows this to happen until the specified date.

This feature is only intended for temporary use, to allow stations to continue to communicate while they are in the process of being upgraded.

To set the "Legacy Authentication" to "Strict", follow these steps:

1. *Open the Drivers > NiagaraNetwork > FoxService's property sheet.*
2. *Set the "Legacy Authentication" property to "Strict."*



3. *Save the changes.*

Use "cookie-digest" Authentication in the WebService

Similarly to the FoxService, the WebService has three different authentication policies available: basic, cookie, and cookie-digest. With **cookie** authentication, the user's password is transmitted to the station in the clear, without any kind of processing. With **cookie-digest** authentication, the user's password is specially processed and is never actually sent to the station. In addition, the station must provide proof of its own identity before authentication is successful.

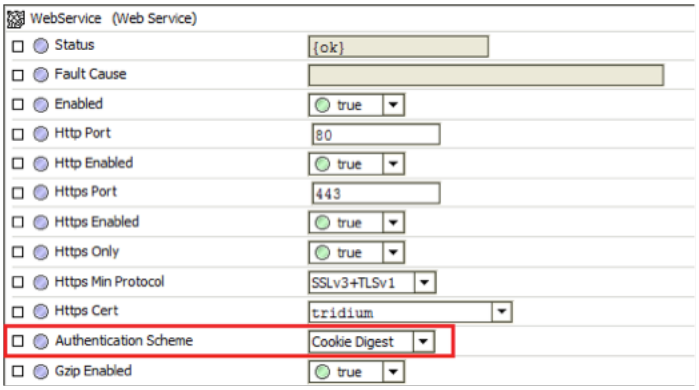
Unless otherwise required, the authentication policy should *always* be set to cookie-digest. One exception to this is use of the LdapUserService, which cannot use cookie-digest authentication for LDAP users.

(continued on page 18)

Use “cookie-digest” Authentication in the WebService (continued)

To set the authentication policy to (the default) cookie-digest option, follow these steps:

1. *Open the Services > WebService’s property sheet.*
2. *Set the “Authentication Scheme” property to “cookie-digest.”*



3. *Save the changes.*

TLS/SSL & Certificate Management

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) provide 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 fox Basic authentication) over an encrypted connection.

Using SSL 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 SSL.

Niagara^{AX} provides several opportunities for using SSL. Starting in Niagara^{AX}-3.7, a number of additional options are available. You should use these options whenever they are feasible. The Niagara^{AX} SSL options are listed below:

- Enable Platform SSL Only (3.7 only)
- Enable Fox SSL Only (3.7 only)
- Enable Web SSL Only
- Enable SSL on Other Services
- Set Up Certificates

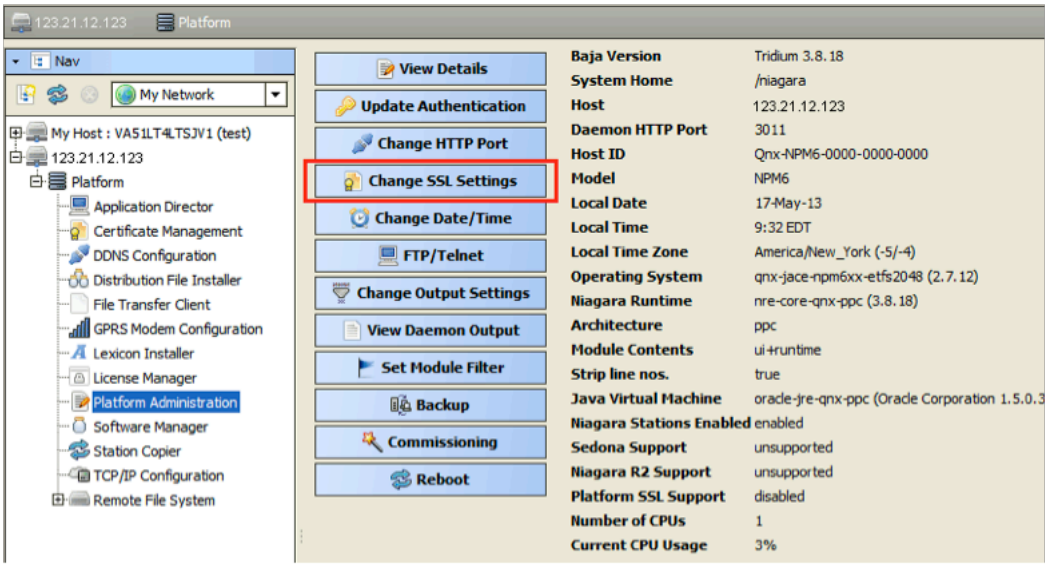
NOTE: *In Niagara^{AX}-3.6, SSL for the Web is a licensed feature and requires the crypto module installed, and the CryptoService added in the station. In Niagara^{AX}-3.7, SSL is built in for non-JACE2/4/5 hosts, and does not require a license. For Nigara^{AX}-3.7 on non-JACE2/4/5 hosts, remove the crypto module and replace it with the cryptoCore, daemonCrypto and platCrypto modules.*

Enable Platform SSL Only (3.7 only)

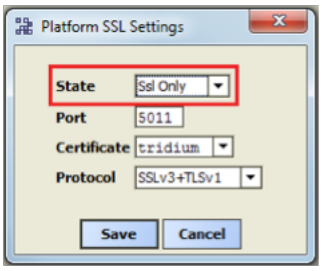
Starting in Niagara[AX]-3.7, SSL can be enabled for platform connections.

To enable platform SSL, do the following:

1. *Open a platform connection.*
2. *Navigate to the "Platform Administration" view, and select "Change SSL Settings."*



3. *A "Platform SSL Settings" dialog opens. Select "SSL Only" from the drop down menu for the "State" field.*



4. *Adjust the other fields as necessary.*
   - ***Port.*** *The default port (5011) is generally acceptable, but may need to be changed due to IT constraints.*
   - ***Certificate.*** *This allows you to select the certificate you want to use for SSL Note: the default self-signed tridium certificate only provides encryption and does not provide for server identity verification. Refer to the Niagara[AX] SSL Connectivity Guide for more details on certificates.*
   - ***Protocol.*** *This specifies whether to use SSLv3 or TLSv1. Both are equally secure and the SSLv3+TLSv1 option, which can use either, is often suitable. However, IT or contractual constraints may require you to pick one or the other.*
5. *Click the "Save" button. Close the platform connection.*

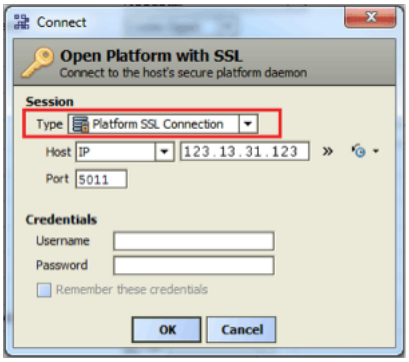
(continued on page 20)

Enable Platform SSL Only (3.7 only) (continued)

NOTE: *If "State" is set to "Enabled" rather than "SSL Only," regular platform connections (not over SSL) are still be permitted. Unless absolutely required, this should not be allowed, because it places the burden of remembering to use SSL on the user initiating the connection – this can easily be forgotten, compromising security.*

Now that SSL has been enabled for platform connections, a platform connection over SSL can be opened, as described below:

1. *Open the "Open Platform" dialog box.*
2. *Under the "Session" section, change the "Type" field to "Platform SSL Connection." Note that the dialog is updated.*



3. *Enter the IP, port and credentials for the platform and click "OK."*

NOTE: *The platform connection over SSL has a tiny lock on the platform icon ( ).*

Enable Fox SSL Only (3.7 only)

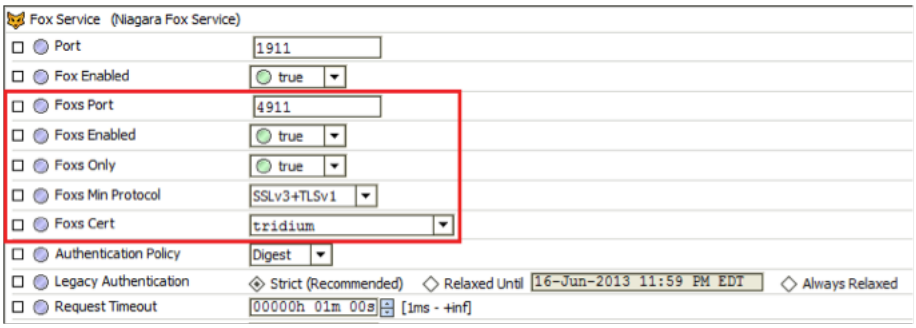
Starting in Niagara^{AX}-3.7, SSL can be enabled for Fox connections. **The steps to follow are outlined below.**

1. *Open a station connection.*
2. *Open Drivers > NiagaraNetwork > FoxService's property sheet.*
3. *Set "Foxs Enabled" to "true."*

(continued on page 21)

Enable Fox SSL Only (3.7 only) (continued)

4 *Set "Foxs only" to "true."*



5 *Adjust the other Foxs settings as necessary.*

- ***Foxs Port.*** *The default port (4911) is generally acceptable, but may need to be changed due to IT constraints.*
- ***Foxs Min Protocol.*** *This determines whether to use SSLv3 or TLSv1. Both are equally secure and the SSLv3+TLSv1 option, which can use either, is often suitable. However, IT or contractual constraints may require you to pick one or the other.*
- ***Foxs Cert.*** *This allows you to select the certificate you want to use for SSL. See the Niagara^{AX} SSL Connectivity Guide for more details on certificates.*

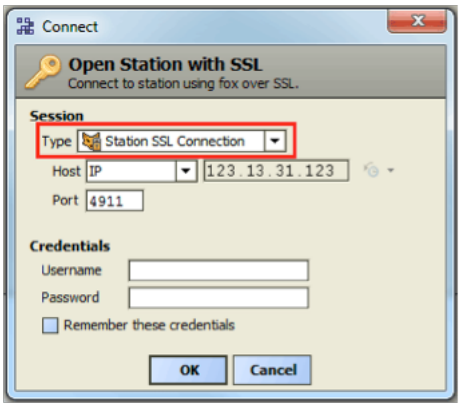
6 *Save the settings and close the station connection.*

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

Now that SSL is enabled, a Foxs (Fox over SSL) connection can be opened, as described below:

1 *Open the "Open Station" dialog box.*

2 *Under the "Session" section, change the "Type" field to "Station SSL Connection." Note that the dialog box is updated.*



3 *Enter the IP, port and credentials for the station and click "OK".*

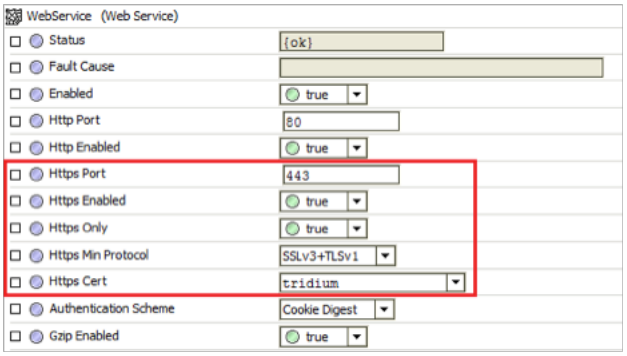
NOTE: *A fox connection over SSL has a tiny lock on the fox icon ( ).*

Enable Web SSL Only

Some of the information in this section is contingent on the type of Java Virtual Machine (IBM J9 or Sun Hotspot) being used by controllers running Niagara^{AX}.

The steps to follow to enable SSL over HTTP are outlined below:

1. *Open a station connection.*
2. *Open Services > WebService's property sheet.*
3. *Set the "Https Enabled" property to "true."*
4. *Set the "Https Only" property to "true."*



5. *Adjust the other Https settings as necessary.*
   - ***Https port.*** *The default port (443) is generally acceptable, but may need to be changed due to IT constraints.*
   - ***Https Min Protocol (Niagara^{AX}-3.7 and the Sun Hotspot JVM only. This does not apply to controllers using IBM J9 JVM).*** *This determines whether to use SSLv3 or TLSv1. Both are equally secure and the SSLv3+TLSv1 option, which can use either, is often suitable. However, IT or contractual constraints may require you to pick one or the other.*
   - ***Https Cert (Niagara^{AX}-3.7 and the Sun Hotspot JVM only. This does not apply to controllers using IBM J9 JVM).*** *This property allows you to select the certificate you want to use for SSL. Note that the default self signed "tridium.certificate" only provides encryption and does not provide server identity verification. See the Niagara^{AX} SSL Connectivity Guide for more details on certificates.*
6. *Save the settings.*

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

Now that SSL is enabled, an HTTPS connection can be opened. Here's how:

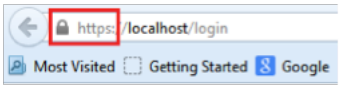
1. *Open a browser.*
2. *Navigate to the station's login page. If the server's certificate was signed by a valid CA then you probably will not see a prompt.*

(continued on page 23)

Enable Web SSL Only (continued)

3 *If prompted, you need to make your decision on whether or not to accept the Certificate based on an understanding of the circumstances. See the Niagara^{AX} SSL Connectivity Guide for more details.*

4 *Note that you now have an https connection.*



SPECIAL NOTE FOR JACE-2/4/5 CONTROLLERS

*The following points concern using **Web SSL only** in Niagara^{AX}-3.6, Niagara^{AX}-3.5, or any J9 JVM host (JACE-2/4/5, regardless even if 3.7).*

- *In these controllers you must use the older crypto module and the station CryptoService.*
- *For security purposes, Hx access to the station is preferred over Wb Web Profile access. Both profiles use HTTPS to pass user credentials, however, with Wb Web Profile, some data (**not user credentials**) is passed over an insecure Fox connection.*

Enable SSL on Other Services

There are a number of services in Niagara^{AX} that communicate with an outside server. For example, the EmailService's OutgoingAccount and IncomingAccount both contact an email server. This connection is not the same as the fox or http connection used by the client to talk to the station, and SSL is handled separately for these types of connections. When setting up a new service on a station, check to see if it includes an SSL option. If SSL 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 talk to supports SSL.

See *Niagara^{AX} SSL Connectivity Guide* and *Niagara^{AX} User Guide* for details about setting up email with SSL features.

Set Up Certificates

Starting in Niagara^{AX}-3.7, new tools are included to help with certificate management. Certificates are required for SSL, and should be set up properly.

NOTE:

Default certificates are self signed and can only be used for encryption, not for server identity verification. This is true for both the legacy crypto features and the new 3.7 and up SSL features.

There are many things to consider when setting up certificates, and a full discussion is beyond the scope of this document. See the *Niagara^{AX} SSL Connectivity Guide* for more information about correctly setting up certificates for a Niagara^{AX} system.

Additional Settings

In addition to the settings discussed in previous sections, there are a few general settings to configure in order to secure a Niagara^{AX} system. These don't fall under a specific category like SSL or passwords, but are nonetheless important to security.

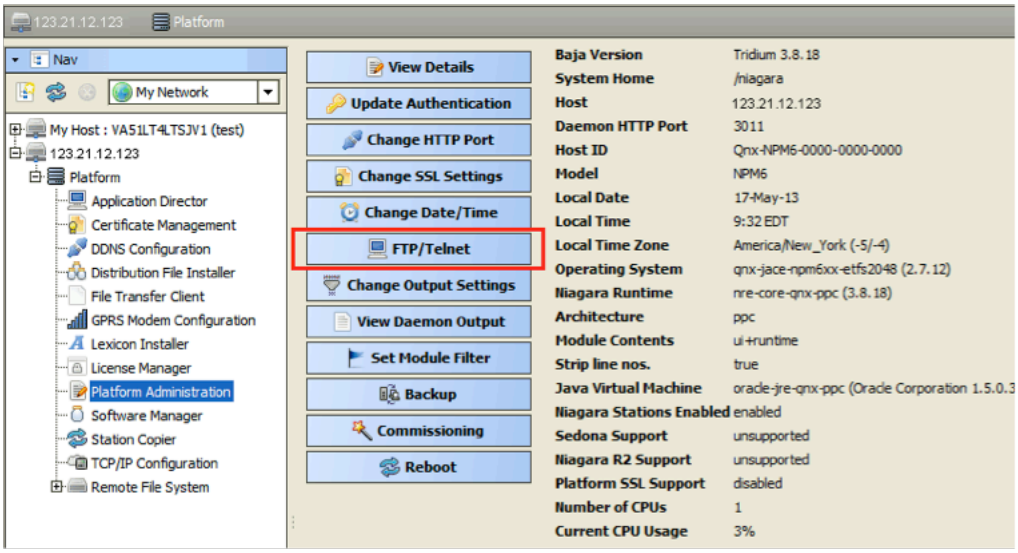
- Disable FTP and Telnet
- Disable Unnecessary Services
- Blacklist Sensitive Files and Folders
- Update Niagara^{AX} to the Latest Release

Disable FTP and Telnet

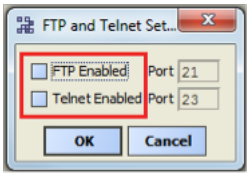
FTP (File Transfer Protocol) and Telnet access to a JACE are disabled by default and should *remain disabled unless necessary for troubleshooting or as directed by Tridium technical support.* This helps prevent unauthorized access to the JACE. Enabling FTP or Telnet on a JACE poses a very significant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

To ensure that FTP and Telnet are disabled on a JACE, follow these steps:

1. *Open a platform connection to the JACE controller.*
2. *In the Platform Administration view, click on "FTP/TELNET."*



3. *When the "FTP and Telnet Settings" dialog box opens, make sure that the "FTP Enabled" and "Telnet Enabled" boxes are* ***not selected.***

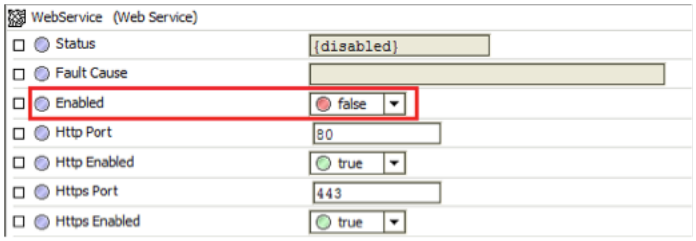


Disable Unnecessary Services

When setting up a Niagara^{AX} 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, the WebService you should disable it. This will prevent potential attackers from using the web to attempt to penetrate the station. The same consideration should be given to the TunnelService and others. See the *Niagara^{AX} Drivers Guide* for information about minimizing possible risks involved with tunneling.

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, or possibly for unused TunnelService.



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.

Blacklist Sensitive Files and Folders

In Niagara^{AX}-3.5u4, 3.6u4 and 3.7u1, a blacklist feature is available. When you implement this feature, files and folders on the blacklist are not accessible remotely through the station. This helps 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 security notes" section in the *Niagara^{AX} Platform Guide* for more details about blacklisting and more notes and cautions about editing the system.properties file.

Additional files may be blacklisted by editing the system.properties files, as described below:

1. *Open the system.properties file.*

(continued on page 26)

Blacklist Sensitive Files and Folders (continued)

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 (ie. 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 paths to blacklist from remote station access (ie. from the
# web or through a fox connection in Workbench).
# File paths are delimited by a semicolon, and follow the body format
# defined in javax.baja.file.FilePath. For example, a value of
# !licenses;!modules would restrict access to the licenses and modules
# directories under the Niagara sys home.
#niagara.remoteBlacklist.filePaths=!lib
```

3 *A station must be restarted before changes to system.properties become effective.*

The additional file patterns or folders to blacklist depend on the particular Niagara^{AX} installation. Think about what needs to be protected and does not absolutely need to be accessed remotely via a fox or web connection.

Update Niagara^{AX} to the Latest Release

Niagara^{AX} updates often include a number of important fixes, including security fixes. Niagara^{AX} 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.

External Factors

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

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

Install JACEs in a Secure Location

Restricting physical access to JACE controllers is essential to security. If an attacker can physically connect to the JACE using a cable, they can gain complete control of the system. This could potentially be disastrous. Keep JACEs secure in a locked room with restricted access.

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 SSL 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.

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. A clever attacker can discover exposed Niagara^{AX} systems without knowing the IP addresses beforehand.