

# Internet in the second second

#### TRIDIUM



#### **Defending the Cyber Castle**

*James Johnson Tridium – Sales Engineer* 





## Agenda

- PKI certificates
- HTML certificate management views

TRIDIUM

- Provisioning
- Certificate signing service
- Code signing and HSM
- Multi-tiered security dashboard
- Syslog service



## Public Key Infrastructure (PKI)

- An infrastructure that supports the distribution of certificates containing public identification keys that are used to both securely identify entities and provide confidentiality of transmissions.
- A Certificate Authority (CA) is an organization which issues and signs digital certificates.
- A digital certificate is an electronic document used to identify an entity, digitally signed by a trusted CA.





# Public Key Cryptography

• Uses a **private** and **public key pair**, used together for encrypting and signing.



- Keys are asymmetric, meaning each key is unique but only two specific keys work together.
- The public key is not a secret and is available to everyone.
- Each participant keeps its private key a secret.
- A sender encrypts data with a recipient's public key and only the recipient with the private key can decrypt the data.
- A sender can sign data with their private key, and everyone can validate the sender signed the data using the sender's public key.



## **PKI Certificates 4.13+**

- The global certificate password is unique to each Niagara instance and is used to encrypt a certificate's private key in the key store
- Previously only CA and code signing certificates required a unique password to encrypt the private key and to prevent unauthorized usage
- All new certificates may use a unique password to encrypt the private key
- Certificates with the private key encrypted using the global certificate password can be used or exported by any user with sufficient permissions.



# Rivest – Shamir – Adleman (RSA)

- Invented in 1977
- Uses prime factorization method to achieve the one-way encryption of data
- Widely used today with most web sites and other applications
- Typical key sizes are 2048, 3072 or 4096 bits
- Relies on the fact that factorization of large prime numbers requires significant computing power to crack
- Compute power has increased significantly over time which requires larger keys to be used to provide sufficient encryption strength



# Elliptic Curve Cryptography (ECC)

- Invented in 1985 and mainstream usage began ~2005
- An approach to public key cryptography based on elliptic curves over finite fields.
- Provides the same cryptographic strength as an RSA-based key with much smaller key sizes.
- Smaller key sizes require fewer processing resources.

Security (bits)	RSA and DH Key Size (bits)	ECC Key Size (bits)
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	521+



## **HTML Certificate Management 4.13+**

- Enables managing certificates without Workbench client
- More flexible interface for configuring certificate extensions
- Supports RSA and EC key algorithms

**NS**2024

<b>Generate Certificat</b>
Generates a signed or se

Generates a signed or self-signed certificate and inserts it into the key store

#### **General Attributes**

Alias 🕐	bldg1f1 (required)
Distinguished Name (DN) 🕐	CN=bldg1f1.ns24.lan, OU=Tridium Support, O=Tridium, L=R 💉 🎱 (required)
Not Before ⑦	05-Feb-2024 12:00 AM ~ EST
Not After ⑦	05-Feb-2025 12:00 AM ~ EST
Key Algorithm	○ RSA
Key Spec 🕐	P-256 V
Certificate Usage	$\odot$ Server $\bigcirc$ Client $\bigcirc$ CA $\bigcirc$ Code Signing $\bigcirc$ Custom
	Type     Critical     Value       Key Usage ⑦     true ●     Digital signature, Key encipherment
Extensions	Subject Alternative Name ⑦ • false DNS Name:bldg1f1.ns24.lan, IP Address:192.168.5.79 × (required)
	Extended Key Usage ⑦ • false TLS Web Server Authentication, TLS Web

OK

Cancel



# **Provisioning 4.6+**

- Jobs can automate generating server certificates and signing using a CA certificate in the provisioning station's key store
- Bootstrap mode allows connectivity for initial certificate generation
- Unique Common Name (CN) and Subject Alternative Name (SAN) derived from <hostname> variable using the session connection details.

-	P	rov	isi	ion	ing	ste	ps	to	run	-
---	---	-----	-----	-----	-----	-----	----	----	-----	---

**NS**202

X Enable Bootstrap Mode

Generate Certificate Job Step (Alias="server", Common Name="<hostname>", Organization="Tridium")

Sign Certificate Job Step (Server Certificate Alias="server")

Set Certificate Alias Job Step (Certificate Alias="server")

Generate Self Signed Certificate

#### **Generate Certificate**

Generate a self signed certificate in the user keystore

Alias	server	(required)
Common Name (CN)	<hostname></hostname>	(required)
	* this may contain the host name or address of the server	r
Organizational Unit (OU)	Tridium	
Organization (O)	Tridium	(required)
Locality (L)	Richmond	
State/Province (ST)	Virginia	
Country Code (C)	US (required)	
Not Before	06-Feb-2024 12:00 AM EST	
NotAfter	05-Feb-2025 12:00 AM EST	
Key Size	♦ 1024 bits ♦ 2048 bits ♦ 3072 bits ♦ 4096 bits	
Certificate Usage	♦ Server ♦ Client ♦ CA ♦ Code Signing	
Alternate Server Name		
Alternate Server URI		1
Email Address		1
	Digital signature Non-repudiation Key encipt	herment
Key Usage	🗌 Data encipherment 🔲 Key agreement 📃 Certificate	e signing
	CRL signing CRL signing CRL signing	
	OK Cancel	



# **Certificate Signing Service 4.13+**

- Intended primarily for use on a Niagara supervisor
- Allows components within the local or remote stations to make secure Certificate Signing Requests (CSR) to obtain signed X509 certificates
- Requires a CA certificate in the user key store of the station with the certificate signing service





# **Simple Signing Profile**

- Defines the desired CA certificate to utilize for signing requests
- Configures certificate properties
  - Expiration Period 365 days default
  - Key Purpose Client, Server, Cert Authority, Code Signing
  - Key Type RSA, EC
  - Common Name

**NS**20

Subject Alternative Name



TDID

# **Fox Signing Requester**

- Configures which Niagara network station is used to fulfill CSRs
- Initially invoke action to Onboard which must be approved in the signing service station
- Once onboarded, certificate is automatically renewed before expiry
- Action to manually renew if needed
- Configures certificate properties
  - Common Name
  - Subject Alternative Name

Signing Requester	Fox Signing Requester				
Status	{ok}				
Fault Cause					
Enabled	🔵 true 🔽				
Requester State	Onboarded				
Advance Renewal Percent	8 96 [1 - 100]				
Advance Renewal	+029d 04h 48m				
Retry Period	001d 00h 00m 🛒 [5 minutes-5 days]				
Last Attempt	09-Feb-2024 09:35 AM EST				
Last Success	09-Feb-2024 09:36 AM EST				
Last Failure	07-Feb-2024 03:27 PM EST				
📔 Next Renewal Attempt	10-Jan-2025 04:48 AM EST				
📔 Alarm On Failure To Onboa	rd 🔵 false 🤝				
Alarm On Failure To Renew	true 🔽				
Alarm Source Info	Alarm Source Info				
Requester Id	70b216aa-bfa0-45d0-8df7-f05065209703				
Signing Service Station	AcmeAnvil 🗸				



# **Local Signing Requester**

- Used to fulfill CSRs for services in the local station with the Signing Service
- Initially invoke action to Onboard which is automatically approved
- Once onboarded, certificate is automatically renewed before expiry
- Action to manually renew if needed
- Configures certificate properties
  - Common Name
  - Subject Alternative Name

📔 Signing Requester	Local Signing Requester
Carl Status	{ok}
Fault Cause	
📔 Enabled	🔵 true 🧹
📔 Requester State	Not Onboarded
隌 Advance Renewal Percent	8 % [1 - 100]
📄 Advance Renewal	+000d 00h 00m
📔 Retry Period	001d 00h 00m 🛓 [5 minutes - 5 days]
📄 Last Attempt	null
📄 Last Success	null
📄 Last Failure	null
📔 Next Renewal Attempt	null
📔 Alarm On Failure To Onboard	false 🗸
📔 Alarm On Failure To Renew	🔵 true 🔽
Alarm Source Info	Alarm Source Info
Requester Id	73da877b-d619-4a2d-9582-1a90c5909f5d



#### **Common Name Template**

- Configures the resolution of the certificat subject Common Name (CN)
- Options include station name, device nan network name, host name, UID, IP addres Niagara station address, or custom
- Custom property resolves BFormat agains the component in the station
- Prefix, middle and suffix properties allow flexible resolution
- Added to either the signing profile or the requester
- Signing profile configuration overrides the requester configuration

8	serverProfile Simple Sig	ning Profile		
Þ	Ca Alias And Password	ns2024 int		
	Expiration Period	365d 00h 00m = [1hour-+inf]		
	Key Purpose	Server		
Þ	💦 Certificate Store	Signing Record Store		
Þ	keyType	Certificate Parameter		
₽	minKeySize	Certificate Parameter		
Ŧ	CommonNameTemplat	e Common Name Template		
	Parameter Type	DN_FIELD		
	Dn Field	CN		
	Prefix	None		
	Middle	Host Name Or Ip Address 🚽		
	Suffix	None		
	Custom	<pre>%parent.name%</pre>	0	
	Separator	-	]	
Þ	subjectAlternateName	Certificate Extension Parameter		



# **Subject Alternative Name**

- Configures the certificate's Subject Alternative Name extension
- Options include DNS name, directory name, email, IP address, registered ID and URI
- BFormat options for values include
  - %commonName%
  - %hostnameOrIpAddress%
  - %ipAddress%
  - 0.0.0.0
  - %niagaraStationAddressOrHostname%
  - %niagaraStationAddressOrIpAddress%
  - %niagaraStationAddressOrHostnameOrIpAddress%
- Added to either the signing profile or the requester
- Signing profile and requester configurations are merged

EXTENSION				
2.5.29.17				
xtension Type:	Subject Alternati	ve Name	· (?)	
ritical:	🛑 false 🚽 🗸			
	IP Address:	0.0.0.0,	DNS Name	e: %commonName%
/alue:				
2	xtension Type: ritical: alue:	xtension Type: Subject Alternati ritical: false IP Address: alue:	xtension Type: Subject Alternative Name ritical: IP Address: 0.0.0.0, alue:	xtension Type: Subject Alternative Name           ritical: <ul> <li>false</li> <li>IP Address: 0.0.0.0, DNS Name</li> <li>alue:</li> </ul>

# Signing Requester 4.13+

- Used to connect a Niagara station using MQTT to Amazon Web Services (AWS) utilizing Just In Time Provisioning (JITP)
- JITP allows a fleet of devices to automatically connect to AWS with auto-generated certificates as a means of authentication
- Signing requester handles requesting and renewing the certificate automatically

A	vsJitpMqttAuthenticator (Aws Jit	p Mqtt Authenticator)
Q	Broker Endpoint	
Q	Client I D	
Q	Broker Port	1883 [0-100000]
Ψ.	Callback Router	Mqtt Callback Router
Q	Certificate Alias	
9	Certificate Alias And Password	default
9	Cert Requester	Fox Signing Requester
	Status	{ok}
	Fault Cause	
	Enabled	🔵 true 🕞
	📔 Requester State	NotOnboarded
	Advance Renewal Percent	8 % [1 - 100]
	Advance Renewal	+000d 00h 00m
	Retry Period	001d 00h 00m 🚆 [5 minutes-5 days]
	Last Attempt	null
	Last Success	null
	🗎 Last Failure	null
	📔 Next Renewal Attempt	null
	📔 Alarm On Failure To Onboard	d 🛑 false 🧹
	Alarm On Failure To Renew	🔵 true 🔍
Þ	Alarm Source Info	Alarm Source Info
	Requester Id	
	Signing Service Station	Select Station



# **Combined Signed Cert Config 4.14+**

- Added to the Security Service in the station
- Signing requester initiates CSR and updates the configured Fox, Web and Platform services
- Fox Signing Requester used in remote stations



тріліі



# Individual Signed Cert Config 4.14+

- Added to specific components such as the Fox Service, Web Service or Additional HTTPS Cert
- Enables the parent component to initiate and individual CSR



TDID



## **Session Token Manager**

- Admin user approves or rejects session tokens for stations which have sent an onboarding request
- Once approved the remote station can send CSR to the signing service station

Name	State	Profile	requestingStation	username	comment	*
70b216aa-bfa0-45d0-8df7-f05065209703	Expired	serverProfile	Bldg1F1	SystemAdmin		
c2aad426-9bc1-4596-8ebf-2bf067658e37	Unapproved		Bldg1F2	SystemAdmin	new station Bldg1F1	

✓ Approve

A Reject





# **Provisioning Jobs**

**NS**2024

Notall Combined Signed Cert Config for Fox/Web/Platform to share

#### Install Combined Signed Cert Config for Fox/Web/Platform to share

Configure how the Combined Signed Cert Config will be processed on each target station's SecurityService for signed certificate onboarding/renewal from the given Signing Service Station. NOTES:

This step requires that a reciprocal NiagaraNetwork connection already exists between this station and the remote station (and the remote to Signing Service station if different).
 If needed, use the 'Set Station Connection Credentials', 'Enable Bootstrap Mode', and 'Setup Reciprocal Connection' provisioning steps to make the reciprocal connections first.
 It is expected that the Signing Service with Signing Profile(s) is already configured properly on only the Signing Service Station (usually the local station).

3. After executing, check the Signing Service's Fox Signing Transport store for Session Tokens to approve for any new signed cert onboarding requests from remote stations. 4. The Fox Signing Requester will be specified for any new or overridden Combined Signed Cert Config processed.

Install Combined Signed Cert Job Step					
🃔 Apply To Fox Service	🔵 true 🔍				
) Apply To Web Service	🔵 true 🔍				
隌 Apply To Platform	🔵 true 🔍				
Signing Service Station	AcmeAnvil				
Auto Cert Alias Format	<pre>%sys().station.stat</pre>	ionName%-combined	0		
隌 Auto Cert Password	•••••	Use global certificate	password		
📔 Signing Service Onboarding Comment	initial provisioning	g on boarding			
Behavior If Config Already Exists	Skip (only add and process if non-existent)				
	Skip (only add and process if non-existent)				
	Override (apply new settin	gs to existing and re-proce	ss, or add if non-existent)		
	Onboard existing only (usi	ng existing, non-overridde	en settings)		
	Renew existing only (using	existing, non-overridden	settings)		



×

# **Provisioning Jobs**

A Install Individual Signed Cert Config on a Service

#### Install Individual Signed Cert Config on a Service

Configure how each Individual Signed Cert Config will be processed on each target station for signed certificate onboarding/renewal from the given Signing Service Station. NOTES:

 This step requires that a reciprocal NiagaraNetwork connection already exists between this station and the remote station (and the remote to Signing Service station if different). If needed, use the 'Set Station Connection Credentials', 'Enable Bootstrap Mode', and 'Setup Reciprocal Connection' provisioning steps to make the reciprocal connections first.
 It is expected that the Signing Service with Signing Profile(s) is already configured properly on only the Signing Service Station (usually the local station).
 After executing, check the Signing Service's Fox Signing Transport store for Session Tokens to approve for any new signed cert onboarding requests from remote stations.
 The Fox Signing Requester will be specified for any new or overridden Individual Signed Cert Configs processed.

👼 Install Individual Signed Cert Job Step				
Target Type To Perform Install	web	- WebService	· © ·	
Dptional Remote Target Base Ord	null			- ·
Signing Service Station	AcmeAnvil			
🃔 Auto Cert Alias Format	webserver (2)		0	
📔 Auto Cert Password	•••••	Use global certif	icate password	
🃔 Signing Service Onboarding Comment	initial onboarding for web service			
Behavior If Config Already Exists	Skip (only add and pro	ocess if non-existent)		
		OK Cancel		





×

# **Code Signing**

- The process of digitally signing executables and scripts to confirm the software author and guarantee the code has not been altered or corrupted since it was signed
- Trusted timestamping is the process of securely keeping track of the creation and modification times of a document
- Timestamping Authority (TSA) URL online server which timestamps the code signature so a client can verify when the code was digitally signed





## Hardware Security Module (HSM)

- HSMs are hardened, tamper-resistant hardware devices used to secure cryptographic processes
- HSMs could be a card inside of a PC, a rack mounted device, a USB device or part of a cloud service
- As of mid-2023, all commercial certificate authorities require the use of a HSM to protect the private key of any code signing certificate
- Starting in 4.14 version, the gradle scripts are updated to support code signing compiled modules using a physical HSM device that supports the PKCS11 standard

https://www.niagara-community.com/s/article/Code-Signing-using-Hardware-Security-Modules





# **Reachable Stations 4.13+**

- Defines all downstream Niagara stations that are routable through this Niagara station
- System Database indexing (4.13+) and Security Dashboard (4.14+) may leverage reachable stations



**NS**2024

	Config . Drive	rs . N	lagaraNetwori	к . А	cmeAnvil	. Sys Der	
P	roperty Sheet						
0	Sys Def (Niagara Sys Def Device Ext)						
	Enabled		🔵 true	-			
	Status		{ok}				
	Fault Cause	1					
Þ	Role Manag	er	Peer				
Þ	Sync Task		Sync Task				
Ŧ	Reachable S	Stations	Reachable	Stations			
	Status	1	ok}				
	State	1	dle				
	Enabled		true 🗸				
	Last Atte	empt 0	5-Feb-2024	02:26 1	PM EST		
	Last Suc	cess 0	5-Feb-2024	02:26 1	PM EST		
	📔 Last Fail	ure r	null				
	Fault Ca	use					
	Bldg1F2	R	eachable Sta	ation Info			
	Bldg1F1	R	eachable Sta	ation Info	0		
	Bldg2	R	eachable Sta	ation Info			
	🕨 🚡 Bldg3	R	eachable Sta	ation Info	0		



# **Reachable Stations Info 4.13+**

- Info and Spy view shows routes to each station
- If multiple routes exist to same station, route preference is utilized



**NS**2024

Remote Station | niagaraNetwork | reachableStations | noForceUpdateIncludeUnoperationalIncludeUnoperationa

<b>Reachable Station</b>	Version	Route To Reachable Station	Route Preference (0 = primary)
Bldg1F1	4.14.0.118	AcmeAnvil -> Bldg1F1	0
Bldg1F1	4.14.0.118	AcmeAdhesives -> Bldg1F1	1
Bldg1F2	4.14.0.118	AcmeAnvil -> Bldg1F2	0
Bldg2	4.14.0.118	AcmeAnvil -> Bldg2	0
Bldg3	4.14.0.118	AcmeAnvil -> Bldg3	0

Bldg1F1 Reachable St	ation Info		
Station Name	BldglFl		
Station Version	4.14.0.118		
Route To Station	AcmeAnvil;BldglFl		
Min Version Along Route	4.14.0.118		
Virtual Space Ord	<pre>station: slot:/Drivers/NiagaraNetwork/AcmeAnvil/virtual vi</pre>		
Time To Reach	+00000h 00m 00.000s		
Route Enabled	🔵 true 🤝		

TRIDIUM

# **Multi-tiered Security Dashboard 4.14+**

NS2(

- When enabled the Security Dashboard in the top-level station can display information from all reachable stations
- Security Dashboard Extension displays information from that specific reachable station

		Property Sheet	
		Security Dashboard Ext (Reachable Station Security)	y Dashboard Device Ext)
		Retry Trigger	15 minutes {Sun Mon Tue Wed Thu Fri Sa
		🕨 🦏 Data Importer	Security Dashboard Data Import
SecurityService (Security Service)		Used By Last Security Dashboard Data Import	🔵 true
Status	{ok}		
Fault Cause			
Enabled	🔵 true 🔽		
O Certificates	Certificate Folder		
Save Dashboard Data To Bog	🛑 false 🔍		
Carlon Link Config	Display remote station das	shboard (directly connected and reachable stations)	
) Include Reachable Stations On System Dashboard	🔵 true 🔽		

Bldg1F1

# **Syslog Service**

NS2

- Syslog is a standard protocol for message logging
- Using the BSD message format, messages can be sent over UDP, TCP or TLS
- Allows sending desired event messages from the Niagara station to an external application for analysis

Config : Services : PlatformServices : SyslogPlatformService

#### Syslog Platform Service Plugin

Enabled	Enabled	
Server Host	192.168.5.217	
Server Port	6514 [1	- 65535]
Message Type	BSD	
Transport Protocol	TLS -	
Client Alias And Password	Ŧ	
Platform Log Enabled	Enabled	
Station Log Enabled	Enabled	
Log Level Filter	INFO 👻	
Station Audit Enabled	Enabled	
Security Audit Enabled	Enabled	
Facility	local0 🗸	
Queue Size	1000 [1	- 10000]
Station Server Status	Ok	
Queue Full Percent Station	0%	
Platform Server Status	Ok	
Queue Full Percent Platform	096	
Syslog Server Connection Alarm Enabled	Enabled	
Syslog Server Connection Alarm Support	Ŧ	
Syslog Message Queue Alarm Enabled	Enabled	
Syslog Message Queue Alarm Support	Ŧ	



#### Questions





