Configuring pxGrid in an ISE Distributed Environment
# Table of Contents

**About this Document** ........................................................................................................... 4

**Introduction** .......................................................................................................................... 5

**ISE Distributed Deployment with pxGrid persona Introduction** ............................................. 6

**pxGrid persona Configuration** ............................................................................................. 8

  - Configuring Microsoft CA 2008 R2 Enterprise pxGrid Template ........................................... 8
  - pxGrid Node Configuration without pxGrid Active-Standby ..................................................... 10
  - CA-Signed Node Certificate Generation .................................................................................. 10
  - Exporting pxGrid node public/private key in Primary PAN & MnT node ................................. 14
  - Bulk Session Downloads ........................................................................................................ 17

**Registering ISE Nodes for Distributed Environment** .............................................................. 18

**pxGrid Client Management** .................................................................................................. 20

**pxGrid Client Configuration** ................................................................................................ 22

  - pxGrid Java sdk Installation .................................................................................................. 22
  - Introduction to pxGrid client SDK Java Keystores .................................................................. 23
  - pxGrid client certificate configuration .................................................................................... 24
  - pxGrid client Active-Standby Examples .................................................................................. 30

**Testing pxGrid client in ISE Distributed Environment** .......................................................... 36

  - Viewing Keystore Entries ...................................................................................................... 37

**ISE Distributed Deployment with pxGrid Active-Standby Introduction** ............................... 45

**Registering ISE nodes for Distributed Environment pxGrid Active-Standby** ............................ 46

**Testing pxGrid client in ISE Distributed Environment pxGrid Active-Standby** ....................... 50

  - Testing pxGrid Active-Standby ............................................................................................... 51
    - Basic Operation ..................................................................................................................... 51
  - Testing FailOver ..................................................................................................................... 54
  - Returning Back to Primary ...................................................................................................... 56

**ISE Self-Signed Identity Certificates** .................................................................................... 60

**Sample Certificates from SDK** ............................................................................................. 64

  - Testing pxGrid Client ............................................................................................................ 65

**References** ............................................................................................................................. 67

**Appendices** ............................................................................................................................ 68

  - Alternative way of generating pxGrid node Certificate Signing Request (CSR) ..................... 68
  - Troubleshooting ...................................................................................................................... 70
  - Removing Java and Installing JDK 8.0 on Centos 6.5 ............................................................. 70
Delete Old version of Java..................................................................................................................... 70
Install JDK 8.0........................................................................................................................................... 71
About this Document

This document is intended for Cisco engineers, partners, and customers deploying pxGrid in a production Cisco Identity Services (ISE) 1.3 environment. The reader should be familiar with ISE and pxGrid.

This document focuses on deploying external CA Signed Certificates for the ISE pxGrid node and pxGrid clients.

Other certificate deployment considerations such as: self-signed ISE Identity certificates and pxGrid sample certificates are discussed in detail in the series of following documents: Deploying Certificates with Cisco pxGrid

- Certificate Authority (CA)-signed pxGrid ISE node and CA-signed pxGrid client
- Certificate Authority (CA)-signed pxGrid client and self-signed ISE pxGrid node certificate
- Self-signed certificates with ISE pxGrid node and pxGrid client

For Configuring pxGrid in a test environment, please see reference:


This document will cover an external ISE pxGrid node configuration and pxGrid Active-Standby configuration in a distributed ISE environment. The pxGrid client for testing these configurations is a MacBook Pro running OSX 10.8.5 and Oracle Java Development Kit(jdk-8u-20-macos-x64.dmg) for the pxGrid java SDK. If you are running other versions of Linux, please see http://www.cisco.com/c/dam/en/us/td/docs/security/ise/how_to/HowTo-84-Configure_and_Test_Integration_with_Cisco_pxGrid.pdf

This document also touches on configuring the pxGrid ISE node with self-signed certificates and sample certificates used in POC deployments. However for detailed information, please refer to the associated documents.

A Microsoft Enterprise 2008 CA R2 Enterprise server was used for the Certificate Authority (CA) and signed both the pxGrid client certificate, pxGrid node certificates and ISE node certificates.
Introduction

Cisco Platform Exchange Grid (pxGrid) enables multivendor, cross platform network system collaboration among parts of the IT infrastructure such as security monitoring and network detection systems, network platforms, asset and configuration management, identity and access management platforms and virtually any other IT operations platform. When business or operations needs arise, ecosystem partners can use pxGrid to exchange contextual information via a publish/subscribe method with Cisco platforms that use pxGrid as well as any other ecosystem that uses pxGrid.

There are essentially three components to pxGrid: the publisher, pxGrid client, and the pxGrid controller, Cisco Identity Service Engine (ISE) pxGrid node.

- pxGrid publisher of topics of information, the pxGrid client will subscribe to- In Cisco Identity Services Engine (ISE) version 1.3, ISE is the sole publisher of this information or capabilities as they called as well.
- pxGrid client- can be a supported Cisco Security platform, pxGrid ecosystem partner, or a Linux or MAC host running the pxGrid SDK, that subscribes to the published information.
- pxGrid controller – Cisco Identity Services Engine (ISE) pxGrid node, that controls the client registration/management and topic/subscription processes.

ISE will publish these information topics:

- SessionDirectory- session attributes from authenticated 802.1X sessions
- EndpointProtectionService- Adaptive Network Control (ANC) quarantine/unquarantine mitigation actions
- TrustsecMetadataCapability- Security Group Tag (SGT) Information
- EndpointProfileMetadata- ISE policy information
- IdentityGroup- group and profiling information

The pxGrid client will subscribe to these topics and obtain the ISE contextual information.

ISE is deployed in a distributed environment, where all the nodes have separate personas: Primary PAN (Policy Admin node), Primary MnT (Monitoring) node, PSN (Policy Service Node). The pxGrid node will also be deployed as a separate persona and will require a customized pxGrid template in a CA (Certificate Authority) signed environment. This document covers the procedural steps in configuring pxGrid in this ISE distributed environment using CA signed certificates for both the ISE pxGrid node and the ISE pxGrid client.

This document also covers a pxGrid Active-Standby configuration.

A MAC running OSX 10.8.5 will serve as the pxGrid client in this document.

A Microsoft Enterprise CA (Certificate Authority) 2008 R2 Server, will be the designated CA server. Please note that a customized template for pxGrid will have an Enhanced KeyUsage (EKU) of both client and server authentication. The EKU defines the purpose of the certificate, and is defined by the ISO-defined object identifiers (OIDs), in this use case one for client authentication (1.3.6.1.5.5.7.3.2) and another one for server authentication (1.3.6.1.5.5.7.3.1).
ISE Distributed Deployment with pxGrid persona

Introduction

The Windows 2008 R2 Enterprise CA Server was used as the CA Authority. The CA root certificate was imported into the trusted system certificates store of each of the ISE nodes. The CSR node requests were serviced by the CA using the web server template and admin “usage” certs defined in the ISE nodes, except for the pxGrid nodes.

The pxGrid nodes use a custom template, containing EKU’s for both client and server authentication.

**Note:** The pxGrid template can be a duplicate of the user template using Windows 2003 format, and a duplicate of the user template with EKUs for both client authentication and server authentication added.

The public/private key pair from the pxGrid node must be copied to the system certificate store for each of the Primary PAN (admin) and Primary MnT (monitoring) modes for successful pxGrid operation.

**Note:** In the case of an Active-Standby pxGrid configuration, the public/private key pair of the first pxGrid node (primary pxGrid node) will be exported into the Primary PAN and Primary MnT nodes. The public/private key pair of the second pxGrid node (secondary pxGrid node) will be exported into the secondary PAN or secondary MnT nodes.

The below diagram represents a typical ISE distributed environment with regards to certificate generation for the various ISE nodes. Note the admin “usage” certificate for CSR request generation for all ISE nodes except for the pxGrid node. The CA server will service these requests using the “web server” template. The pxGrid “usage” certificate for the pxGrid node(s) CSR request is serviced by the custom pxGrid template.
The below diagram represents the pxGrid node configuration in a Distributed ISE environment. The pxGrid node is external in all productional environments.

The public/private key from the pxGrid node is copied into the system certificate stores of both the Primary PAN and Primary MnT before enabling the pxGrid controller.
pxGrid persona Configuration

Configuring Microsoft CA 2008 R2 Enterprise pxGrid Template

This section covers the pxGrid certificate template configuration. The pxGrid template must contain both EKU’s for client authentication and server authentication.

The pxGrid template is created in the following steps:

**Step 1** Select->Administrative Tools->Certificate Authority->“+” dropdown next to CA server->Right-Click on Certificate Templates->Manage

**Step 2** Right-Click and Duplicate User template->Select->Windows 2003 Enterprise->OK

*Note:* Select Windows Server 2003 Enterprise so it will appear in the template CA window drop-down
Step 3 Enter name of certificate template, uncheck “Publish certificate in Active Directory”, provide validity period and renewal period.

Step 4 Click on Extensions->Add->Server Authentication->OK->Apply
pxGrid Node Configuration without pxGrid Active-Standby

This section illustrates the steps of defining the ISE nodes, generating CSR requests and obtaining certificates from the CA authority. The process is typical of any ISE distributed deployment. This occurs in stand-alone before joining the nodes from the Primary Admin node.

The pxGrid node will use the ISE pxGrid usage certificates for the initial CSR requests and serviced by the MS CA “pxGrid” template as defined earlier. The returned certificate will be bound to the initial pxGrid CSR request.

The public/private key pair will be exported from the pxGrid node and imported into the Primary PAN and Primary MnT nodes.

Note: In a pxGrid Active-Standby configuration, the public/private key pair from the second or secondary pxGrid node will be imported into the Secondary PAN and Secondary MnT nodes.

The Microsoft CA root certificate will be downloaded and imported into the trusted system certificates store in each ISE node and enabled for “Trust for Authentication within ISE”.

CA-Signed Node Certificate Generation

The following steps outline the procedure for downloading the CA root certificate, generating ISE node CSR requests, and binding certificates to CSR requests.

Note: The CA Root certificate and other serviced certificate requests should be download in base 64 format.

Step 1  Download CA root in base 64 format.

Step 2  Import into the Trusted Certificates Store
Step 3  Generate CSR for the desired Admin, MnT, nodes, in stand-alone environment. Administration-System->Certificates->Certificate Signing Requests-“admin” certificate usage

Step 4  Use the MS CA “Web Server” template to service the certificate requests for the Admin, MnT, PSN nodes.
Step 5  Download in base 64 encoded format

Step 6  Bind the certificate to the CSR requests for each ISE node, individually. (i.e. admin, MnT, PSN)
        Administration-System->Certificates->Certificate Signing Requests->Select certificate and Bind

Step 7  Import the node certificate for each ISE node, individually, then submit
Step 8  Generate CSR for the pxGrid node.
Administration->System->Certificates->Certificate Signing Requests—“pxGrid” certificate usage

Step 9  Submit request MS CA “pxGrid” template to service certificate request for the pxGrid node.
SECURE ACCESS HOW-TO GUIDES

Step 10  Bind the pxGrid certificate to the pxGrid node CSR request.
Administration-System->Certificates->Certificate Signing Requests->Select pxGrid node & bind certificate

Exporting pxGrid node public/private key in Primary PAN & MnT node

There is an alternative way for exporting the primary pxGrid node public/private key pair into the Primary PAN and Primary MnT node for pxGrid operation, please see: Alternative way of generating pxGrid node CSR (Certificate Signing Request) in the Appendices.
The public/private key pair of the pxGrid client node must be copied into the Primary PAN and MnT node. The steps are described below:

**Step 1**  Export the public and private keys from the pxGrid node’s system certificate store and import to the system store for the desired Primary Admin and MnT nodes in Stand-alone mode new installation.

*Note:* If you have an existing ISE 1.3 deployment and adding external pxGrid persona, you can export the public/private key pairs for the pxGrid node from the Primary PAN’s system certificates store and import into the Primary PAN and Primary MnT Administration->System->Certificates->System Certificates, select the certificate and export the certificate and private key. You will have to provide a name for the private key (i.e. cisco123). This will be saved as a zipped file containing both the PEM and PVK (public/private key pair).

**Step 2**  For the desired Primary Admin node, import both public and private key into the System Certificate store, then submit. Administration->System->Certificates->System Certificates and import both the pxGrid PEM and PVK certificates.
Step 3  For the desired Primary MnT node, import both public and private key into the System Certificate store, then submit.
Administration->System->Certificates->System Certificates and import both the pxGrid PEM and PVK certificates.

Step 4  You will see the pxGrid public/private key in the system certificates store of the Primary PAN and Primary MnT nodes.
Secure Access How-To Guides

Bulk Session Downloads

Bulk Session downloads provide active session download queries from the ISE MnT node, using the pxGrid session_download script. This provides available session attributes from authenticated 802.1X authenticated sessions with regards to available ISE contextual information. The public key (PEM) from the MnT node is copied to the pxGrid client, converted to DER, and imported into the truststoreFilename keystore. This will be covered later, but for now export the MnT node certificate as indicated below.

**Note:** In a pxGrid Active Standby Configuration, both the Primary MnT node and the Secondary MnT node certificates need to be imported into the pxGrid client. If either of these certificates does not exist, there will be problem when registering clients, and non-connectivity to pxGrid nodes.

**Step 1**  
Export the public certificate key only from the desired MnT node. This will be used by the pxGrid client for bulk session downloads  
Administration->Certificates->Certificate Management->System Certificates and select the MnT identity cert and export the public certificate
Registering ISE Nodes for Distributed Environment

The desired stand-alone ISE nodes for Primary PAN, Primary MnT, PSN, and pxGrid are registered through the Primary Admin (PAN) node.

These steps are defined below:

**Step 1**  Set the desired Admin node to initially include Primary Admin and Primary MnT personas.

**Step 2**  Register the desired MnT node which will become the primary MnT.

*Note:* The Primary PAN will automatically become the secondary MnT persona. Disable the secondary MnT persona.

**Step 3**  Register the PSN node
**Step 4**
Register the pxGrid node

**Step 5**
Ensure that the pxGrid services started and you have the ISE published capabilities
Administration->pxGrid Services and also enable Auto Registration
pxGrid Client Management

The pxGrid Service menu provides: client management: client registration/deletion, authorization of client “pending” requests when Auto-Registration is disabled. This menu also provides a log history view of the client registered capabilities or information topics.

Enable Auto-Registration – enables auto registration, pxGrid clients will automatically register after the initial pxGrid client authentication has completed.

Disable Auto-Registration – disables auto registration, pxGrid clients will remain in a “pending” state until the administrator moves them into the appropriate “session” or “EPS” group.

Client Groups- client groups will register primarily to the “session” group for pxGrid operation.

Administrator- reserved for ISE

Session- access to session attribute information

EPS- superset of “session” group, used for ANC “Adaptive Network Control” mitigations

Live Log- displays history of client registration and topic subscriptions
<table>
<thead>
<tr>
<th>Client Name</th>
<th>Capability Name</th>
<th>Event Type</th>
<th>Timestamp</th>
<th>Other Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:ise-admin-pnn1@xgrid.cisco.com">ise-admin-pnn1@xgrid.cisco.com</a></td>
<td>GridControllerAdminServiceCapability-1.0</td>
<td>Client subscribed</td>
<td>7:49:49 PM EST, Apr 17 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-admin-mnt1@xgrid.cisco.com">ise-admin-mnt1@xgrid.cisco.com</a></td>
<td>Resync database</td>
<td></td>
<td>7:49:49 PM EST, Apr 17 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-admin-mnt1@xgrid.cisco.com">ise-admin-mnt1@xgrid.cisco.com</a></td>
<td>Client online</td>
<td>3:13:34 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-admin-mnt1@xgrid.cisco.com">ise-admin-mnt1@xgrid.cisco.com</a></td>
<td>Client deleted</td>
<td>3:13:33 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-admin-mnt1@xgrid.cisco.com">ise-admin-mnt1@xgrid.cisco.com</a></td>
<td>Resync database</td>
<td>3:13:30 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>SessionDirectoryCapability-1.0</td>
<td>Publisher added</td>
<td>3:07:34 PM EST, Apr 16 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>IdentityGroupCapability-1.0</td>
<td>Publisher added</td>
<td>3:07:33 PM EST, Apr 16 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>Client online</td>
<td>3:07:33 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>Client deleted</td>
<td>3:07:31 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>SessionDirectoryCapability-1.0</td>
<td>Publisher deleted</td>
<td>3:07:31 PM EST, Apr 16 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-mnt-mnt1@xgrid.cisco.com">ise-mnt-mnt1@xgrid.cisco.com</a></td>
<td>IdentityGroupCapability-1.0</td>
<td>Publisher deleted</td>
<td>3:07:19 PM EST, Apr 16 2015</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:ise-admin-pnn1@xgrid.cisco.com">ise-admin-pnn1@xgrid.cisco.com</a></td>
<td>GridControllerAdminServiceCapability-1.0</td>
<td>Client subscribed</td>
<td>2:44:47 PM EST, Apr 16 2015</td>
<td></td>
</tr>
<tr>
<td>wsa2lab6.com-pgrid_client@xg...</td>
<td>Resync database</td>
<td>2:44:47 PM EST, Apr 16 2015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wsa2lab6.com-pgrid_client@xg...</td>
<td>SessionDirectoryCapability-1.0</td>
<td>Client subscribed</td>
<td>8:17:37 PM EST, Apr 15 2015</td>
<td></td>
</tr>
<tr>
<td>wsa2lab6.com-pgrid_client@xg...</td>
<td>TrustSecMetaDataCapability-1.0</td>
<td>Client subscribed</td>
<td>8:17:37 PM EST, Apr 15 2015</td>
<td></td>
</tr>
</tbody>
</table>
pxGrid Client Configuration

In this section, we cover the pxGrid java SDK installation for pxGrid sample script testing. Register.sh will be run to connect and establish connection with the pxGrid controller. Session_download.sh will be run to download active session records from ISE. These scripts are used for basic testing to ensure that the connection and communication between the pxGrid client and ISE are working. If you have a desire to test all the shell scripts including Adaptive Network Control (ANC) mitigation actions, formerly known as Endpoint Protection Service (EPS). Please see: (http://www.cisco.com/c/dam/en/us/td/docs/security/ise/how_to/HowTo-84-Configure_and_Test_Integration_with_Cisco_pxGrid.pdf)

pxGrid Java sdk Installation

Please see your Cisco Account team for obtaining the pxGrid java SDK libraries

Download the Oracle Java Development Kit for your Linux operating system: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

In order to install the Oracle Java Development Kit, you must uninstall the older version of Java that exists on your system.

**Note:** If you are using a MAC for testing, please see: https://www.java.com/en/download/help/mac_uninstall_java.xml for uninstalling Java

If you are using Centos 6.5, please refer to the Appendices Removing Java and Installing JDK 8.0 on Centos 6.5

Untar the folder. tar -zxf pxgrid-sdk-x.x.x-dist.tar.gz

You will see the following:

- Lib- contains all the GCL Libraries
- Samples- contains bin, certs, conf, lib and src directories
- Bin- contains all the sample scripts
- Certs- contains all the sample pxGrid identity and rootSample certificates
- Src- contains all the java source files

In order to run the pxGrid sample scripts, include the path of jre in the “JAVA_HOME=” environment variable.

An example for the MAC is provided below.

To view the location of the jre path, run the following:

**Note:** you will need root privileges when running sudo

```
sudo find / -name java
Password:
/Applications/pxGridSDK/pxgrid-sdk-1.0.0/samples/src/java
find: /dev/fd/3: Not a directory
find: /dev/fd/4: Not a directory
/Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java
/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java
/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/jre/bin/java
```
Add the path “/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/jre” to JAVA_HOME

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/jre
```

With different versions of Linux, such as Centos 64, make sure “keytool” is included in the path

```
Append the “../jdk1.7._51/bin” to PATH
export PATH=/usr/lib64/qt3.3/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/jeppich/bin:/usr/java/jdk1.7.0_51/bin
```

### Introduction to pxGrid client SDK Java Keystores

Java keystores contain the public/private key pairs of certificates, such as the CA root certificates, host identity or pxGrid client certificate, self-signed certificate. The java keystore itself is a PKCS #12 format (.JKS).

The certificates themselves are either in a PEM or CER format, and converted over to DER and imported into the java keystore.

In this document, we will use CA signed pxGrid client certificates and CA signed ISE certificates.

For pxGrid, there is the keystoreFilename that contains the pxGrid client identity certificate and truststoreFilename keystore that represents the CA root certificates, and the Mnt node certificates.

In addition these keystore values contains associated passwords, keystorePassword and truststorePassword when importing the certificates into the keystore.

The keystoreFilename, keystorePassword, truststoreFilename, truststorePassword, are using in the pxGrid SDK scripts for SASL authentication and connection to the pxGrid persona.

In the illustrated example below, the pxGrid client registers and connects to the pxGrid controller.

```
./register.sh -keystoreFilename pxGridClient.jks -keystoreFilename cisco123 -truststoreFilename root3.jks -truststorePassword cisco123 -group Session -description test -username macbook-pro -hostname 10.0.0.48

-------- properties --------
version=1.0.0
hostnames=10.0.0.48
username=macbook-pro
descriptipon=test
keystoreFilename=pxGridClient.jks
keystorePassword=cisco123
truststoreFilename=root3.jks
truststorePassword=cisco123

---------------------------
registering...
connecting...
account enabled
connected.
done registering.
connection closed
```
In the illustrated example below, the pxGrid client downloads active session records from the MnT node

```
./session_download.sh -keystoreFilename pxGridClient.jks -keystoreFilename cisco123 -truststoreFilename root3.jks -truststorePassword cisco123 -username macbook-pro -hostname 10.0.0.48

-------- properties --------
version=1.0.0
hostnames=10.0.0.48
username=MacBook-Pro
keystoreFilename=pxGridClient.jks
keystorePassword=cisco123
truststoreFilename=root3.jks
truststorePassword=cisco123
filter=null
start=null
end=null

-------------
connecting...
connected.
starting at Wed Dec 10 18:44:49 EST 2014...

session (ip=10.0.0.18, Audit Session Id=0A0000020000000B006E1086, User Name=jeppich, AD User DNS Domain=lab6.com, AD Host DNS Domain=null, AD User NetBIOS Name=LAB6, AD Host NETBIOS Name=null, Calling station id=00:0C:29:D1:8D:90, Session state= STARTED, Epsstatus=null, Security Group=null, Endpoint Profile=VMWare-Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/15, RADIUSAVPairs=[ Acct-Session-Id=00000002], Posture Status=null, Posture Timestamp=, Session Last Update Time=Wed Dec 10 16:41:48 EST 2014 )... ending at: Wed Dec 10 18:44:49 EST 2014

downloaded 1 sessions in 26 milliseconds
-------------

connection closed
```

**pxGrid client certificate configuration**

The following procedure represents the steps for generating keys for the pxGrid client, creating the CSR request, importing certificates converting them to DER adding to the keystores.

**Note:** The pxGrid client configuration is for having a CA-signed pxGrid client and CA-signed pxGrid node certificate. Please see references for other certificate deployment considerations.

The process is described below:

- A private key is generated for the pxGrid client
- A CSR (Certificate Signing Request) is generated from the private key. A challenge key is required which will be used later on for keystore management
  
- The CA Authority signs the CSR request with a valid pxGrid template as defined earlier
- A PKCS#12 file will be created from the public/private key pair and root certificate. This will be used for keystore creation of the keystoreFilename (JKS) and truststoreFilename (JKS)
- The keystoreFilename (JKS) will be created
- The truststoreFilename (JKS) will be created
- Import the ISE identity certificate from the ISE MnT primary and ISE MnT secondary nodes used for active session record or bulk download sessions.
- Convert the ISE identity certificate PEM file to a DER format and add to the truststore filename keystore along with the CA root certificate
- Import the pxGrid client certificate into the keystore filename (JKS)
- Import the CA Root certificate into the truststore filename (JKS)
- Copy both files into the pxGrid “../samples/bin/..” folder and run the scripts

**Step 1**  **Generate a private key**
Generate a private key (i.e. mac.key) for the pxGrid client.

*Note:* this can be .key name can be any name, here I called it mac.key

```
openssl genrsa -out mac.key 4096
Generating RSA private key, 4096 bit long modulus
..............................................................................
..............................................................................
ed is 65537 (0x10001)
```

**Step 1**  **Generate the CSR request**
Generate a CSR request (i.e. mac.csr) to the CA Authority. Provide a challenge password (i.e. cisco123)

*Note:* the .csr can be any name, here I called it mac.csr for uniformity, also the challenge password can be any name

```
openssl req -new -key mac.key -out mac.csr

You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []: cisco123
An optional company name []: Eppich, Inc
the same password throughout this document, easier to maintain, and cut down on errors
```

**Step 2**  **CA authority signs the pxGrid CSR request.**
The CA authority must service the user certificate using a pxGrid template with both EKUs for client authentication and server authentication.

*Note:* A CA template of Windows 2003 was selected, so it would appear in the Drop-down. A user template was duplicated with both EKUs for client and server authentication.
**Step 3  Create the PKCS12 file**

Create a pxGrid client pkcs12 file (mac.p12) from the private key in the pxGrid client certificate (i.e. mac.cer). This will be used for keystore management and can be a random filename with a .p12 extension. Include the CA root file (i.e. root2a).

```bash
openssl pkcs12 -export -out mac.p12 -inkey mac.key -in mac.cer -chain -CAfile root2a.cer
```

Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123

**Step 4  Create the keystoreFilename for the pxGrid client**

Create the pxGrid client identity keystore (i.e. mac.jks). This will be the pxGrid client identity keystore. This can be a random filename with a .jks extension. This will serve as the keystoreFilename and associated keystorePassword in the pxGrid script examples.

```bash
keytool -importkeystore -srckeystore mac.p12 -destkeystore mac.jks -srcstoretype PKCS12
```

Enter destination keystore password: cisco123
Re-enter new password: cisco123
Enter source keystore password:
Enter for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled

**Step 5  Export the public ISE Identity certificate from the ISE MnT Primary and ISE MnT Secondary nodes**

Export only the public ISE Identity certificate into the pxGrid client, note that this will be in .pem format. You can rename the file with .pem extension to make it easier to read. In this example, the file was renamed to mnt1.pem.

**Note:** Both ISE MnT Primary and ISE MnT Secondary nodes are required by the pxGrid client if pxGrid Active-Standby is configured.
Step 6  Convert the ISE Identity MnT node PEM format to DER format

```
openssl x509 -outform der -in mnt1.pem -out mnt1.der
```

Step 7  Add the ISE MnT DER file to truststoreFilename

Add the ISE identity cert to the trust keystore (i.e. caroot1.jks). this will be the trusted keystore. This can be a random filename with a .jks extension. This will become the truststoreFilename and truststorePassword used in the pxGrid scripts.

```
keytool -import -alias isemnt -keystore caroot1.jks -file mnt1.der
```

Enter keystore password:  cisco123
Re-enter new password: cisco123

Owner: CN=ise.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ0531D-CA, DC=lab6, DC=com
Serial number: 61262d760000000000000d
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.2.840.113549.1.9.15 Criticality=false
  0000: 30 35 30 0E 06 08 2A 86 48 86 F7 0D 03 02 02 02 050...H........
  0010: 00 80 30 0E 06 08 2A 86 48 86 F7 0D 03 04 02 02 ..0...H........
  0020: 00 80 30 07 06 05 2B 0E 03 02 07 30 0A 06 08 2A ..0...+.0....
  0030: 36 84 86 F7 0D 03 07 .H.....

#2: ObjectId: 1.3.6.1.4.1.311.21.10 Criticality=false
  0000: 30 32 30 0A 06 08 2B 06 01 05 05 07 03 01 30 0A 020...+....0.
  0010: 06 08 2B 06 01 05 05 07 03 02 30 0A 06 08 2B 06 ..+......0...+
  0020: 02 01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 ....0...+
  0030: 37 0A 03 04 7...

#3: ObjectId: 1.3.6.1.4.1.311.21.7 Criticality=false
  0000: 30 2D 06 25 2B 06 01 04 01 82 37 15 08 DC FD 1A 0....%.....7....
  0010: 87 CB EB 79 81 89 9D 2D 86 E6 FC 53 86 82 A1 38 ...y....S...8
  0020: 5E 86 D1 B8 23 85 FC EF 40 02 01 64 02 01 03 ^...@...d...

#4: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
Step 8: Import the pxGrid client into keystoreFilename

Import the pxGrid client certificate into the identity keystore.

```bash
Johns-MacBook-Pro:pxGridsdk jeppich$ keytool -import -alias pxGridMAC -keystore mac.jks -file mac.cer

Enter keystore password: cisco123
Certificate already exists in keystore under alias <1>
Do you still want to add it? [no]: yes
Certificate was added to keystore
```
Note: If you receive the following message the certificate was already added to a pre-existing keystore, you can say “no” and still be okay. I selected “yes” so we can verify that the certificate was added later on.

Step 9  Add the CA root certificate to the truststoreFilename

Add the CA root certificate to trusted keystore. The CA root certificate needs to be trusted as well.

```
keytool -import -alias ca_root1 -keystore caroot1.jks -file root2a.cer
```

```
Enter keystore password: cisco123
Owner: CN=lab6-WIN-BG7GP0Q053ID-CA, DC=lab6, DC=com
Issuer: CN=lab6-WIN-BG7GP0Q053ID-CA, DC=lab6, DC=com
Serial number: 448a6d6486c91cb14c6888c127d16c4e
Certificate fingerprints:
  SHA256:
Signature algorithm name: SHA256withRSA
Version: 3
Extensions:
  #1: ObjectId: 1.3.6.1.4.1.311.21.1 Criticality=false
      0000: 02 01 00

  #2: ObjectId: 2.5.29.19 Criticality=true
      BasicConstraints: [CA: true
                         PathLen:2147483647]

  #3: ObjectId: 2.5.29.15 Criticality=false
      KeyUsage:
        DigitalSignature
        KeyCertSign
        Crl_Sign

  #4: ObjectId: 2.5.29.14 Criticality=false
      SubjectKeyIdentifier [
        KeyIdentifier [
          0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ...&...Z.6&...
          0010: 6A C8 79 2C
        ]
      ]

Trust this certificate? [no]: yes
Certificate was added to keystore
```

Step 10  Copy the identity keystore (mac.jks) and trust keystore (caroot1.jks) into the pxGrid “../samples/bin/..” folder.
pxGrid client Active-Standby Examples

For pxGrid Active-Standby, you need to export both primary MnT and secondary MnT public certificates (PEM) to pxGrid client and convert them both to DER. Both certificates need to be added to the truststoreFilename keystore along with the CA Root certificate (root2a.cer)

Johns-Macbook-Pro:mntnodes jeppich$ openssl x509 -outform der -in mnt1.pem -out mnt1.der
Enter keystore password:
Re-enter new password:

Owner: CN=mn1.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Valid from: Tue Jan 20 20:08:40 EST 2015 until: Fri Jan 20 20:18:40 EST 2017
Certificate fingerprints:
SHA256:
Signature algorithm name: SHA256withRSA
Version: 3
Extensions:
#1: ObjectID: 1.2.840.113549.1.9.15 Criticality=false
0000: 30 35 30 0E 06 08 2A 86  48 86 F7 0D 03 02 02 02 050...*.H.......
0010: 80 30 0E 06 08 2A 86  48 86 F7 0D 03 04 02 02 ...0...*.H....
0020: 80 30 07 06 05 02 B0 0E  03 02 07 30 0A 06 08 2A ...0+...0...
0030: 80 48 86 F7 0D 03 07 ..H....

#2: ObjectID: 1.3.6.1.4.1.311.21.10 Criticality=false
0000: 30 32 30 0A 06 08 2B 06  01 05 05 07 03 01 30 0A 020+.......
0010: 06 08 2B 06 01 05 05 07  03 02 30 0A 06 08 2B 06 ...+.....0...+
0020: 01 05 05 07 03 04 30 0C  07 06 0A 2B 06 01 04 01 82 ......0+....
0030: 37 0A 03 04 7...

#3: ObjectID: 1.3.6.1.4.1.311.21.7 Criticality=false
0000: 30 2D 06 25 2B 06 01 04  01 82 37 15 08 0C FD 1A 0%-+.7.....
0010: 87 01 79 81 89 9D 2D  86 E6 FC 53 86 82 A1 38 ...y......8
0020: 05 8E 86 D1 B8 83 85 FC EF  40 02 01 64 02 01 03 ^...#...d...

#4: ObjectID: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [ ]
  [  
  accessMethod: caIssuers
  accessLocation: URIName: ldap://CN=lab6-WIN-BG7GPQ053ID-CA,CN=AIACN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectCla ss=certificationAuthority
  ]
]

#5: ObjectID: 2.5.29.35 Criticality=false
AuthorityKeyIdIdentifier [ ]
  KeyIdentifier [ ]
    0000: A9 C7 8E 26 9C F5 37 0A  E6 5A 15 36 26 D4 A2 06 ...6..7.Z.6&...
    0010: 6A C8 79 2C j.y,

#6: ObjectID: 2.5.29.31 Criticality=false
CRLDistributionPoints [ ]
  [DistributionPoint:
[URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-BG7GPQ053ID,DC=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]
]

#7: ObjectId: 2.5.29.32 Criticality=false
CertificatePolicies [
  [CertificatePolicyId: [2.5.29.32.0]]
]

#8: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  serverAuth
  clientAuth
  emailProtection
  1.3.6.1.4.1.311.10.3.4
]

#9: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  DigitalSignature
  Key_Encipherment
]

#10: ObjectId: 2.5.29.17 Criticality=false
SubjectAlternativeName [
  DNSName: mnt1.lab6.com
]

#11: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
    0000: DA 39 A3 EE 5E 6B 0D 32 55 BF EF 95 60 18 90 ...kK.2U...`
    0010: AF D8 07 09 .
  ]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
Johns-Macbook-Pro:mntnodes jeppich$ openssl x509 -outform der -in mnt2.pem -out mnt2.der
Johns-Macbook-Pro:mntnodes jeppich$ keytool -import -alias lab1 -keystore caroot1.jks -file mnt2.der
Enter keystore password: keytool error: java.lang.Exception: Certificate not imported, alias <lab1> already exists
Johns-Macbook-Pro:mntnodes jeppich$ keytool -import -alias lab2 -keystore caroot1.jks -file mnt2.der
Enter keystore password: Owner: CN=mnt2.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 613244ec000000000044
Valid from: Wed Mar 04 18:11:54 EST 2015 until: Fri Mar 03 18:11:54 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.3.6.1.4.1.311.20.2 Criticality=false
0000: 1E 12 00 57 00 65 00 62 00 53 00 72 00 76 ...W.e.b.S.e.r.v
0010: 00 65 00 72

#2: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
  ]

accessLocation: URIName: ldap://CN=lab6-WIN-BG7GPQ053ID-CA,CN=IAA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com
#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [KeyIdentifier [0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ...&..7..Z.6&...0010: 6A C8 79 2C j.y,
]
]

#4: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [{DistributionPoint: [URIName: ldap://CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-BG7GPQ053ID,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]}]

#5: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [serverAuth]

#6: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [DigitalSignature Key_Encipherment]

#7: ObjectId: 2.5.29.17 Criticality=false
SubjectAlternativeName [DNSName: mnt2.lab6.com]

#8: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [KeyIdentifier [0000: DA 39 A3 EE 5E 6B 4B 0D 32 55 BF EF 95 60 18 90 .9..^k.2U... 
0010: AF D8 07 09 ....
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
Johns-Macbook-Pro:mntnodes jeppich$ keytool -list -v -keystore caroot1.jks
Enter keystore password: 

Keystore type: JKS
Keystore provider: SUN
Your keystore contains 2 entries
Alias name: lab2
Creation date: Mar 4, 2015
Entry type: trustedCertEntry
Owner: CN=mnt2.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 613244ec0000000000044
Valid from: Wed Mar 04 18:11:54 EST 2015 until: Fri Mar 03 18:11:54 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3
Extensions:

#1: ObjectId: 1.3.6.1.4.1.311.20.2 Criticality=false
0000: 1E 12 00 57 00 65 00 62 00 53 00 65 00 72 00 76 ...W.e.b.S.e.r.v
0010: 00 65 00 72

#2: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess {
  accessMethod: caIssuers
  accessLocation: URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN,BG7GPQ053ID,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectClass=certificationAuthority
}

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier {
  KeyIdentifier {
  0000: A9 C7 8E 26 9C F5 0A   E6 5A 15 36 26 D4 A2 06 ...&..7..Z.6&...
  0010: 6A C8 79 2C                                        }y,
  }
}

#4: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints {
  [DistributionPoint:
    [URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN,BG7GPQ053ID,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]
  ]
}

#5: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages {
  serverAuth
}

#6: ObjectId: 2.5.29.15 Criticality=true
KeyUsage {
  DigitalSignature
  Key_Encipherment
}

#7: ObjectId: 2.5.29.17 Criticality=false
SubjectAlternativeName {
  DNSName: mnt2.lab6.com
}

#8: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier {
  KeyIdentifier {
  0000: DA 39 A3 EE 5E 6B 4B 0D 32 55 BF EF 95 60 18 90 ...^K.2U...`...
  0010: AF D8 07 09                                        ....
  }
}

*******************************************
*******************************************
Alias name: lab1
Creation date: Mar 4, 2015
Entry type: trustedCertEntry
Owner: CN=mnt1.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 61326a180000000000031
Valid from: Tue Jan 20 20:08:40 EST 2015 until: Fri Jan 20 20:18:40 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 1.2.840.113549.1.9.15 Criticality=false
0000:  30 35 30 0E 06 08 02 0A 86 48 86 F7 0D 03 02 02 02 0500...*H.....
0010:  00 01 05 05 07 03 01 30 0A 06 08 02 0B 06 03 02 02 02 0500...*H.....
0020:  00 01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 0000...0......
0030:  37 0A 03 04 7..0...

#2: ObjectId: 1.3.6.1.4.1.311.21.10 Criticality=false
0000:  30 32 30 0A 06 08 02 0B 06 01 05 05 07 03 02 30 0A 06 08 02 0B 06 0300...0......
0010:  01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 0000...+......0......

#3: ObjectId: 1.3.6.1.4.1.311.21.7 Criticality=false
0000:  30 32 0D 06 25 2B 06 01 04 01 82 37 15 08 DC FD 1A 0...+%......7......
0010:  87 CB EB 79 81 89 9D 2D 86 E6 FC 53 86 82 A1 38 ...y--------.8...
0020:  02 06 86 D1 B8 23 85 FC EF 40 02 01 64 02 01 03 ^...#...d...

#4: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [ ]
    accessMethod: caIssuers
    accessLocation: URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectClass=certificationAuthority

#5: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [ ]
    KeyIdentifier [ ]

#6: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [ ]
    DistributionPoint: [URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-BG7GPQ053ID.CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]

#7: ObjectId: 2.5.29.32 Criticality=false
CertificatePolicies [ ]
    [CertificatePolicyId: [2.5.29.32.0] []]

#8: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [ ]
    serverAuth
clientAuth
    emailProtection
    1.3.6.1.4.1.311.10.3.4]
#9: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [  
  DigitalSignature  
  Key_Encipherment  
]

#10: ObjectId: 2.5.29.17 Criticality=false
SubjectAlternativeName [  
  DNSName: mnt1.lab6.com  
]

#11: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [  
  KeyIdentifier [  
  0000: DA 39 A3 EE 5E 6B 4B 0D 32 55 BF EF 95 60 18 90 .9..^k2U...'..  
  0010: AF D8 07 09  
  ]  
]

*****************************************************************************  
*****************************************************************************  

Johns-Macbook-Pro:mntnodes jeppich$ openssl x509 -outform der -in root2a.cer -out root2a.der  
Johns-Macbook-Pro:mntnodes jeppich$ keytool -import -alias lab3 -keystore caroot1.jks -file root2a.der  
Enter keystore password:  
Owner: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com  
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com  
Serial number: 448a6d6486c91cb14c6888c127d16c4e  
Certificate fingerprints:  
Signature algorithm name: SHA256withRSA  
Version: 3  

Extensions:  
#1: ObjectId: 1.3.6.1.4.1.311.21.1 Criticality=false  
  0000: 02 01 00  ...  

#2: ObjectId: 2.5.29.19 Criticality=true  
BasicConstraints: [  
  CA:true  
  PathLen:2147483647  
]  

#3: ObjectId: 2.5.29.15 Criticality=false  
KeyUsage [  
  DigitalSignature  
  Key_CertSign  
  Crl_Sign  
]  

#4: ObjectId: 2.5.29.14 Criticality=false  
SubjectKeyIdentifier [  
  KeyIdentifier [  
  0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ...&..Z.6&...  
  0010: 6A CB 79 2C  
  ]  
]

Trust this certificate? [no]: yes  
Certificate was added to keystore
Testing pxGrid client in ISE Distributed Environment

The pxGrid scripts: register.sh and session download.sh will be run to ensure pxGrid client connection and pxGrid registration. Session downloads will ensure that there are no issues with the ISE MNT certificate and the pxGrid client.

**Step 1** Register the pxGrid client

```bash
Johns-Macbook-Pro:bin jeppich$ ./register.sh -keystoreFilename mac.jks -keystorePassword cisco123 -truststoreFilename caroot1.jks -truststorePassword cisco123 -hostname 10.0.0.48 -username mac1 -group Session
------- properties -------
version=1.0.0
hostnames=10.0.0.48
username=mac1
description=null
keystoreFilename=mac.jks
keystorePassword=cisco123
truststoreFilename=caroot1.jks
truststorePassword=cisco123
--------------------------
registering...
connecting...
connected.
done registering.
connection closed
Johns-Macbook-Pro:bin jeppich$
```

Verify the pxGrid client has registered to the pxGrid controller

Administration->pxGrid Services

**Step 2** Run the Session download

```bash
Johns-Macbook-Pro:bin jeppich$ ./session_download.sh -keystoreFilename mac.jks -keystorePassword cisco123 -truststoreFilename caroot1.jks -truststorePassword cisco123 -hostname 10.0.0.48 -username mac1
------- properties -------
version=1.0.0
hostnames=10.0.0.48
username=mac1
keystoreFilename=mac.jks
keystorePassword=cisco123
truststoreFilename=caroot1.jks
truststorePassword=cisco123
filter=null
start=null
```
Viewing Keystore Entries

By viewing the keystore entries you can view the trusted certificate entries for the keystoreFilename and truststoreFilename keystores.

Step 1  Verify caroot1.jks, the truststoreFilename keystore

Johns-Macbook-Pro:bin jeppich$ keytool -list -v -keystore caroot1.jks
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 3 entries

Alias name: lab3
Creation date: Mar 4, 2015
Entry type: trustedCertEntry

Owner: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 448a6d6486c91cb14c6888c127d16c4e
Certificate fingerprints:

Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectID: 1.3.6.1.4.1.311.21.1 Criticality=false
0000: 02 01 00 ... #2: ObjectID: 2.5.29.19 Criticality=true
BasicConstraints:
CA: true
PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [ DigitalSignature
   Key_CertSign
   Crl_Sign
]

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [ KeyIdentifier [ 0000: A9 C7 8E 26 9C F5 37 0A   E6 5A 15 36 26 D4 A2 06 ...&..7..Z.6&...
0010: 6A C8 79 2C ][ ]
]

===============================================
===============================================

Alias name: lab2
Creation date: Mar 4, 2015
Entry type: trustedCertEntry

Owner: CN=mnt2.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 613244ec000000000044
Valid from: Wed Mar 04 18:11:54 EST 2015 until: Fri Mar 03 18:11:54 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 1.3.6.1.4.1.311.20.2 Criticality=false
0000: 1E 12 00 57 00 65 00 62 00 53 00 65 00 72 00 76 ...W.e.b.S.e.r.v
0010: 00 65 00 72 .e.r

#2: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [ ]
   accessMethod: caIssuers
   accessLocation: URName: ldap:///CN=lab6-WIN-BG7GPQ053ID-
   CA,CN=AIACN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectCla
   ss=certificationAuthority
]
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [ KeyIdentifier [ 0000: A9 C7 8E 26 9C F5 37 0A   E6 5A 15 36 26 D4 A2 06 ...&..7..Z.6&...
0010: 6A C8 79 2C ][ ]
]

#4: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [ ]


SECURE ACCESS HOW-TO GUIDES

[URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-
BG7GPQ053ID,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]

#5: ObjectId: 2.5.29.37 Criticality=false
  ExtendedKeyUsages [serverAuth]

#6: ObjectId: 2.5.29.15 Criticality=true
  KeyUsage [DigitalSignature Key_Encipherment]

#7: ObjectId: 2.5.29.17 Criticality=false
  SubjectAlternativeName [DNSName: mnt2.lab6.com]

#8: ObjectId: 2.5.29.14 Criticality=false
  SubjectKeyIdentifier [KeyIdentifier [0000: DA 39 A3 EE 5E 6B 0D 32 55 BF EF 95 60 18 90 ..^kK.2U....]
  0010: AF D8 07 09 ....]

***************************************************
************
******************************
Alias name: lab1
Creation date: Mar 4, 2015
Entry type: trustedCertEntry
Owner: CN=mnt1.lab6.com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 61326a1800000000000031
Valid from: Tue Jan 20 20:08:40 EST 2015 until: Fri Jan 20 20:18:40 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.2.840.113549.1.9.15 Criticality=false
  0000: 30 35 30 0E 06 08 2A 86 48 86 F7 0D 03 02 02 02 050...*.H........
  0010: 00 80 30 0E 06 08 2A 86 48 86 F7 0D 03 04 02 02 ..0...*.H........
  0020: 00 80 30 0E 06 08 02 8A 86 48 86 F7 0D 03 04 02 02 ..0...*.H........
  0030: 86 48 86 F7 0D 03 07 ....

#2: ObjectId: 1.3.6.1.4.1.311.21.10 Criticality=false
  0000: 30 32 30 0A 06 08 2B 06 01 05 05 07 03 01 30 0A 020...+....0.
  0010: 00 68 08 2B 06 01 05 05 07 03 02 30 0A 06 08 2B 06 ..+....+...+
  0020: 00 01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 ......0...+
  0030: 37 0A 03 04 ....7

#3: ObjectId: 1.3.6.1.4.1.311.21.7 Criticality=false
  0000: 30 2D 06 25 2B 06 01 04 01 82 37 15 08 DC FD 1A 0-..+....7....
  0010: 87 CB EB 79 81 89 9D 2D 86 E6 FC 53 86 82 A1 38 ...y....8 S...8
  0020: 5E 86 D1 B8 23 85 FC EF 40 02 01 64 02 01 03 ^...#..@d...
## SECURE ACCESS HOW-TO GUIDES

### #4: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false

**AuthorityInfoAccess**

- **accessMethod**: caIssuers
- **accessLocation**: URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectClass=certificationAuthority

### #5: ObjectId: 2.5.29.35 Criticality=false

**AuthorityKeyIdentifier**

**KeyIdentifier**

- 0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ...7Z6...
- 0010: 6A C8 79 2C

### #6: ObjectId: 2.5.29.31 Criticality=false

**CRLDistributionPoints**

- DistributionPoint:
  - URIName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-BG7GPQ053ID,CN=CFP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint

### #7: ObjectId: 2.5.29.32 Criticality=false

**CertificatePolicies**

- CertificatePolicyId: [2.5.29.32.0]

### #8: ObjectId: 2.5.29.37 Criticality=false

**ExtendedKeyUsages**

- serverAuth
- clientAuth
- emailProtection
- 1.3.6.1.4.1.311.10.3.4

### #9: ObjectId: 2.5.29.15 Criticality=true

**KeyUsage**

- DigitalSignature
- Key_Encipherment

### #10: ObjectId: 2.5.29.17 Criticality=false

**SubjectAlternativeName**

- **DNSName**: mnt1.lab6.com

### #11: ObjectId: 2.5.29.14 Criticality=false

**SubjectKeyIdentifier**

**KeyIdentifier**

- 0000: DA 39 A3 EE 5E 6B 4B 0D 32 55 BF EF 95 60 18 90 ...^K.2U...
- 0010: AF D8 07 09

---

**Johns-Macbook-Pro:bin jeppich$**
Step 2 Verify mac.jks, the keystoreFilename keystore

Johns-Macbook-Pro:bin jeppich$ keytool -list -v -keystore mac.jks
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

Alias name: 1
Creation date: Jan 28, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: O=Internet Widgits Pty Ltd, ST=Some-State, C=AU
Issuer: CN=lab6-WIN-BG7GPQ531D-CA, DC=lab6, DC=com
Serial number: 61186413000000000034
Valid from: Wed Jan 28 14:35:54 EST 2015 until: Sat Jan 28 14:45:54 EST 2017
Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.2.840.113549.1.9.15 Criticality=false
0000: 30 35 30 0E 06 08 02 A8 86 48 86 F7 0D 03 02 02 02 050...*H......
0010: 00 30 30 0E 06 08 02 A8 86 48 86 F7 0D 03 04 02 02 ...0...*H........
0020: 00 80 30 07 06 05 02 A8 0E 06 08 02 A8 86 48 86 F7 0D 03 04 02 02 ..0...*H........
0030: 38 48 86 F7 0D 03 07 .H..

#2: ObjectId: 1.3.6.1.4.1.311.21.10 Criticality=false
0000: 30 32 30 0A 06 08 02 B0 06 01 05 05 07 03 01 30 0A 020...*....0.
0010: 00 30 30 0E 06 08 02 A8 86 48 86 F7 0D 03 04 02 02 ...0...*H......
0020: 01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 ......0...*
0030: 37 0A 03 04 7...

#3: ObjectId: 1.3.6.1.4.1.311.21.7 Criticality=false
0000: 30 2D 06 25 2B 06 01 04 01 82 37 15 08 DC FD 1A 0...4%....7......
0010: 87 CB EB 79 81 89 9D 2D 86 E6 FC 53 86 82 A1 38 ...y...$...8
0020: 5E 86 D1 B8 23 85 FC EF 40 02 01 64 01 02 01 03 ^...#...d...

#4: ObjectId: 1.3.6.1.5.7.1.1 Criticality=false
AuthorityInfoAccess [ ]

AuthorityKeyIdentifier [ ]

#5: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [ ]

KeyIdentifier [ ]

#6: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [ ]
#7: ObjectId: 2.5.29.32 Criticality=false

#8: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [ serverAuth clientAuth emailProtection 1.3.6.1.4.1.311.10.3.4 ]

#9: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [ DigitalSignature Key_Encipherment ]

#10: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [ KeyIdentifier [ 0000: 81 52 81 84 98 22 43 85 5E 95 06 14 D2 5A A8 70 .R...C.^....Z.p 0010: 15 06 CF DB .... ] ]

Certificate[2]:
Owner: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 448a6d6486c91cb4c8888c127d16c4e
Certificate fingerprints:
   SHA256:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.3.6.1.4.1.311.21.1 Criticality=false
0000: 02 01 00 ...

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
   CA:true
   PathLen:2147483647 ]

#3: ObjectId: 2.5.29.15 Criticality=false
KeyUsage [ DigitalSignature Key_CertSign Crl_Sign ]

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [ KeyIdentifier [ 0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ....&...Z.6... 0010: 6A C8 79 2C ] ]
SECURE ACCESS HOW-TO GUIDES

*******************************************

Alias name: macstore
Creation date: Jan 28, 2015
Entry type: trustedCertEntry

Owner: O=Internet Widgits Pty Ltd, ST=Some-State, C=AU
Issuer: CN=lab6-WIN-BG7GPQ053ID-CA, DC=lab6, DC=com
Serial number: 6118d613000000000034
Valid from: Wed Jan 28 14:35:54 EST 2015 until: Sat Jan 28 14:45:54 EST 2017

Certificate fingerprints:
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 1.2.840.113549.1.9.15 Criticality=false
0000: 30 35 30 0E 06 08 2A 86 48 86 F7 0D 03 02 02 02 050...*.H......
0010: 00 80 30 0E 06 08 2A 86 48 86 F7 0D 03 04 02 02 ..O...*.H......
0020: 00 80 30 07 06 05 02 0B 0E 03 02 07 30 0A 06 08 2A .O...+.0..*
0030: 86 48 86 F7 0D 03 07 ..H.....

#2: ObjectId: 1.3.6.1.4.1.311.21.10 Criticality=false
0000: 30 32 30 0A 06 08 2B 06 01 05 05 07 03 01 30 0A 020.........0.
0010: 06 08 2B 06 01 05 05 07 03 02 30 0A 06 08 2B 06 ..+........+
0020: 01 05 05 07 03 04 30 0C 06 0A 2B 06 01 04 01 82 .......0.....
0030: 37 0A 03 04 7....

#3: ObjectId: 1.3.6.1.4.1.311.21.7 Criticality=false
0000: 30 2D 06 02 2B 06 01 04 01 82 37 15 08 DC FD 1A 0-.+%......7.....
0010: 87 CB EB 79 81 89 9D 2D 86 E6 FC 53 86 82 A1 38 ...y........S....
0020: 5E 86 D1 B8 23 85 FC EF 40 02 01 64 02 01 03 ^..#@...d...

#4: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess {
  [accessMethod: caIssuers
      accessLocation: URLName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?cACertificate?base?objectClass=certificationAuthority]
}

#5: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier {
  KeyIdentifier [ 
    0000: A9 C7 8E 26 9C F5 37 0A E6 5A 15 36 26 D4 A2 06 ........7.Z.66...
    0010: 6A C8 79 2C j.y,
  ]
}

#6: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints {
  [DistributionPoint: 
    [URLName: ldap:///CN=lab6-WIN-BG7GPQ053ID-CA,CN=WIN-BG7GPQ053ID,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=lab6,DC=com?certificateRevocationList?base?objectClass=cRLDistributionPoint]]}
#7: ObjectId: 2.5.29.32 Criticality=false
CertificatePolicies [  
  [CertificatePolicyId: [2.5.29.32.0]  
  [] ] ]

#8: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [  
  serverAuth  
  clientAuth  
  emailProtection  
  1.3.6.1.4.1.311.10.3.4 ]

#9: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [  
  DigitalSignature  
  Key_Encipherment ]

#10: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [  
  KeyIdentifier [  
    0000: 81 52 81 84 98 22 43 85 5E 95 06 14 D2 5A A8 70 .R...^....Z.p  
    0010: 15 06 CF DB .... ]  
  ] ]

*****************************************************************************
*****************************************************************************

Johns-Macbook-Pro:bin jeppich$
ISE Distributed Deployment with pxGrid Active-Standby

Introduction

In this section, we cover pxGrid Active-Standby. In an ISE distributed deployment, there can be only (2) pxGrid nodes. One handling the pxGrid client connections controlling the pxGrid services and the other one, for fail-over. One pxGrid node can be active at a time.

The ISE Distributed Deployment with pxGrid Active-Standby consists of the Primary Admin node, Secondary Admin node, Primary MnT node, Secondary MnT node, two PSNs, and two separate pxGrid personas.

We will add the Secondary Admin node, Secondary MnT node, and secondary pxGrid node to form the pxGrid Active-Standby Configuration.

The public/private keys are exported from the first or primary pxGrid persona into the Primary Admin and Primary MnT nodes System Certificates stores.

Note: This has already been configured as part of the initial ISE distributed deployment.

The public/private keys are exported from the second or secondary pxGrid persona into the Secondary Admin and Secondary MnT nodes System Certificate stores.

The Primary and Secondary MnT Identity certificates are exported into the pxGrid client for bulk active session downloads. If either of these certificates are missing, you may not see the pxGrid client register.

Registered client accounts, subscriptions, topics, etc. are Active-Active synchronized between pxGrid servers through PANs. The primary and secondary pxGrid nodes are Active-Standby.

The pxGrid clients connect to the primary PxGrid node. If the primary pxGrid node goes down, the client connects to the secondary pxGrid node and all registered clients and transactions will be there. This will be illustrated in this document.

pxGrid Active-Standby Configuration
Registering ISE nodes for Distributed Environment pxGrid Active-Standby

Here we register the secondary nodes.

**Step 1** Import public/private key pair from the secondary pxGrid node into the secondary PAN.
Administration->System->Certificate->Certificate Management->System Certificates and import the public/private key of the secondary pxGrid node

**Note:** This can be done when all nodes are in Stand-alone. This can also be done directly from the Primary PAN. It is assumed that the public/private key pair from the secondary pxGrid node have been exported.

There is an alternative way for exporting the secondary pxGrid node public/private key pair into the Secondary PAN and Secondary MnT node for pxGrid operation, please see: Alternative way of generating pxGrid node Certificate Signing Request (CSR) in the Appendices.

**Step 2** Import public/private key pair from the secondary pxGrid node into the secondary PAN.
Administration->System->Certificate->Certificate Management->System Certificates and import the public/private key of the secondary pxGrid node
Step 3  You should verify that the public/private key pairs have been successfully imported into the ISE Secondary PAN, and ISE Secondary MnT nodes. Administration->System->Certificates->Certificate Management->System Certificates

Step 4  Register the Secondary Primary admin node through Primary Admin Node Administration->System->Deployment, register ISE node as the Secondary Admin node
Step 5  Register the Secondary Monitoring Node through the Primary Admin node 
Administration->System->Deployment, register ISE node as the Secondary Monitoring node

Note: Ensure that the Secondary MnT has joined the domain, check External Identity Services, if the secondary MnT node has not joined the domain, there will be no connectivity to the pxGrid node.
Step 6  Add the Secondary pxGrid node
Administration->System->Deployment, register ISE node as the Secondary pxGrid node

Step 7  Ensure that the pxGrid services have started, and you see the ISE published nodes:
Administration->pxGrid Services
Testing pxGrid client in ISE Distributed Environment pxGrid Active-Standby

This section illustrates the pxGrid-Standby configuration by adding the Secondary PAN, Secondary MnT and secondary pxGrid node. In addition we will test the configuration by:

Basic Operation:

- Registering pxGrid client to primary pxGrid node

**Note:** In a pxGrid Active-Standby Configuration, only the primary pxGrid node can be active, the secondary pxGrid node is “not running” as displayed by “sh application status ise” on the pxGrid secondary node

- Download active session records from the MnT Primary node
- View Registered pxGrid client status in ISE
- View Deployment node status to indicate pxGrid node status

Testing pxGrid Node Failover to secondary pxGrid node

- “application stop ise” on primary pxGrid node to simulated down pxGrid node
- “application start ise” on secondary pxGrid node to start secondary pxGrid node
- Download active sessions from MnT Primary node to compare sessions, they should be the same
- Register pxGrid client to secondary pxGrid node
- View Registered pxGrid client in ISE
- View Deployment node status to indicate pxGrid node status

Returning back pxGrid primary node

- application stop ise” on secondary pxGrid node
- “application start ise” on primary pxGrid node
- Download active sessions from MnT Primary node to compare sessions, they should be the same
- Register pxGrid client to primary pxGrid node
- View Registered pxGrid client in ISE
- View Deployment node status to indicate pxGrid node status
Testing pxGrid Active-Standby

Basic Operation

Here we register the pxGrid client to the first pxGrid node or the primary pxGrid node while in the pxGrid Active Standby configuration.

Basic Operation:

- Registering pxGrid client to primary pxGrid node

**Note:** In a pxGrid Active-Standby Configuration, only the primary pxGrid node can be active, the secondary pxGrid node is “not running” as displayed by “sh application status ise” on the pxGrid secondary node

- Download active session records from the MnT Primary node
- View Registered pxGrid client status in ISE
- View Deployment node status to indicate pxGrid node status

The diagram below illustrates that all nodes are active

---

**Step 1**  Verify that all nodes are active
Administration->System->Deployment, you should see all the nodes
Step 2 Verify that pxGrid services up and the ISE Primary PAN, ISE Secondary PAN, ISE Primary MnT, and ISE Secondary MnT node are registered clients.

Administration->pxGrid Services

![pxGrid Services](image)

Step 3 Register a pxGrid client and download the active session records using the pxGrid register and session_download shell scripts. Note the IP address of the primary and secondary IP address of the pxGrid nodes for –hostname.

Note: In a productional environment you may have a GUI that designates secondary pxGrid node.

```
Johns-Macbook-Pro:bin jeppich$ ./register.sh -keystoreFilename mac.jks -keystorePassword cisco123 -truststoreFilename caroot1.jks -truststorePassword cisco123 -hostname 10.0.0.48 10.0.0.49 -username mac_engineering15
------- properties -------
version=1.0.0
hostnames=10.0.0.48, 10.0.0.49
username=mac_engineering15
description=null
keystoreFilename=mac.jks
keystorePassword=cisco123
truststoreFilename=caroot1.jks
truststorePassword=cisco123
--------------------------
registering...
connecting...
done registering.
connection closed
Johns-Macbook-Pro:bin jeppich$ ./session_download.sh -keystoreFilename mac.jks -keystorePassword cisco123 -truststoreFilename caroot1.jks -truststorePassword cisco123 -hostname 10.0.0.48 10.0.0.49 -username mac_engineering15
------- properties -------
version=1.0.0
hostnames=10.0.0.48, 10.0.0.49
username=mac_engineering15
keystoreFilename=mac.jks
keystorePassword=cisco123
truststoreFilename=caroot1.jks
truststorePassword=cisco123
filter=null
start=null
end=null
---------------
-----------------
connecting...
connected.
starting at Thu Mar 05 00:54:43 EST 2015...
```
Step 4  You should see the registered client, mac_engineering15  
Administration->pxGrid Services
Testing FailOver

Testing pxGrid Node Failover to secondary pxGrid node

- “application stop ise” on primary pxGrid node to simulated down pxGrid node
- “application start ise” on secondary pxGrid node to start secondary pxGrid node
- Download active sessions from MnT Primary node to compare sessions, they should be the same
- Register pxGrid client to secondary pxGrid node
- View Registered pxGrid client in ISE
- View Deployment node status to indicate pxGrid node status
- “application stop ise” on primary pxGrid node to simulated down pxGrid node
- “application start ise” on secondary pxGrid node to start secondary pxGrid node
- Download active sessions from MnT Primary node to compare sessions, they should be the same

**Step 1** Verify that the primary pxGrid node or pxGrid 1 is down.
Administration->System->Deployment

![Deployment Nodes](image)

**Step 2** Run the register and session download commands to verify that you have connection the secondary pxGrid node.
Step 3 Verify that pxGrid services up and you see the ISE published nodes.
Administration->pxGrid Services

Step 4 Register a pxGrid client and download the active session records using the pxGrid register and session_download shell scripts while the primary pxGrid node is down.
Step 5 Verify that you can see registered pxGrid client, mac_engineering20
Administration->pxGrid Services

Returning Back to Primary

Returning back pxGrid primary node

- application stop ise” on secondary pxGrid node
- “application start ise” on primary pxGrid node
- Download active sessions from MnT Primary node to compare sessions, they should be the same
- Register pxGrid client to primary pxGrid node
- View Registered pxGrid client in ISE
- View Deployment node status to indicate pxGrid node status

Step 1  Verify that the primary pxGrid node is back up
   Administration->System->Deployment, you should see all the nodes

Step 2  Verify the pxGrid services are running and you see the ISE published nodes
   Administration->pxGrid Services

Step 3  Verify that you still have connection be downloading the active sessions by running session_download
Step 4  
Register a pxGrid client to verify that all is working.

Step 5  
View pxGrid client, mac_engineering50, on the ISE pxGrid controller
Administration->pxGrid Services
## Secure Access How-To Guides

### Cisco Identity Services Engine Live Log

<table>
<thead>
<tr>
<th>Client Name</th>
<th>Client Description</th>
<th>Capabilities</th>
<th>Status</th>
<th>Client Group</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>iee-mnt-mnt2</td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
<td>View</td>
</tr>
<tr>
<td>iee-mnt-mnt1</td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
<td>View</td>
</tr>
<tr>
<td>iee-admin-mnt1</td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
<td>View</td>
</tr>
<tr>
<td>iee-admin-admin1</td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
<td>View</td>
</tr>
<tr>
<td>iee-admin-mnt2</td>
<td></td>
<td></td>
<td></td>
<td>Online</td>
<td>View</td>
</tr>
<tr>
<td>mac</td>
<td></td>
<td></td>
<td></td>
<td>Offline</td>
<td>View</td>
</tr>
<tr>
<td>mac_engineering17</td>
<td></td>
<td></td>
<td></td>
<td>Offline</td>
<td>View</td>
</tr>
<tr>
<td>mac_engineering15</td>
<td></td>
<td></td>
<td></td>
<td>Offline</td>
<td>View</td>
</tr>
<tr>
<td>mac_engineering20</td>
<td></td>
<td></td>
<td></td>
<td>Offline</td>
<td>View</td>
</tr>
<tr>
<td>mac_engineering58</td>
<td></td>
<td></td>
<td></td>
<td>Offline</td>
<td>View</td>
</tr>
</tbody>
</table>

*Note: This table displays the status and capabilities of various client names.*
ISE Self-Signed Identity Certificates

The ISE Self-Signed Identity certificate for pxGrid may be used when no external CA Authority is available and for testing pxGrid and ISE implementations without using the sample certificate from the pxGrid SDK. By default the self-signed Identity Certificate contains both an Enhanced Key Usage (EKU) of Server Authentication (1.3.6.1.5.5.7.3.1) and Client Authentication (1.3.6.1.5.5.7.3.2) and is located in the ISE System Certificates store.

pxGrid clients may use self-signed certs, please see: Using Self-Signed Certificates with pxGrid ISE node and pxGrid clients

pxGrid clients may also use CA-signed certs, please see: Using Self-Signed Certificates with pxGrid ISE node and CA-signed pxGrid clients

**Step 1** Enable pxGrid usage in the ISE Self-Signed Identity Certificate
Administration->System->Certificates->System Certificates->Edit the ISE self-signed certificate and select pxGrid and then Save
Step 2  
Export the public ISE Identity Self-Signed certificate under Trusted Certificates. Administration->System->Certificates->Edit the self-signed certificate and “Export Certificate Only” 

Note: This will be saved as a PEM file

Step 3  
Import the PEM file into the Trusted Certificate Store
SECURE ACCESS HOW-TO GUIDES

Administration->System->Certificates->Trusted Certificates->select PEM certificate and enable “Trust for authentication within ISE”, then submit

Step 4 Enable pxGrid persona
Administration->System->Deployment->Edit deployment node and enable pxGrid, then Save

Step 5 Start the pxGrid Services
Administration->pxGrid Services
Note: If you see no connectivity to pxGrid node, this may take some time to come up.
Sample Certificates from SDK

In this example, we use sample certificates from the pxGrid SDK, that will be used for POC only, and not in productional environments. Here we import rootSample.crt for the trusted CA certificate and import iseSample1.crt and iseSample1.key, which will server as the public/private pair for the pxGrid client for client registration. For more detailed information on a POC deployment, sample Certificates and pxGrid sample shell scripts, please see: http://www.cisco.com/c/dam/en/us/td/docs/security/ise/how_to/HowTo-84-Configure_and_Test_Integration_with_Cisco_pxGrid.pdf

Step 1  Import rootSample.crt into the Trusted System Certificates of ISE
Administration->System->Certificates->Trusted Certificate->import the rootSample.crt, then Submit

Step 2  Import iseSample1.crt and iseSample1.key into the System Certificate store of ISE
Administration->System->Certificates->System Certificates->Import and use cisco123 for the password, then submit.

Step 3  Enable pxGrid persona
Step 4  Start the pxGrid Services
Administration->pxGrid Services

Note: If you see no connectivity to pxGrid node, this may take some time to come up.

Testing pxGrid Client

Make sure you export the ISE Self-Signed Identity Certificate from the MnT node, or in the case of a Stand-Alone deployment to the pxGrid client for Bulk Session downloads (Please see Bulk Session Downloads).
Here we check for client registration and session downloads

**Step 1**  Use the register.sh script and run the following:

```
./register.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -truststoreFilename rootSample.jks -truststorePassword cisco123 -group Session -username iseSample -hostname 10.0.0.39 -group Session
-------- properties -------
version=1.0.0
hostnames=10.0.0.39
username=iseSample
descriptipon=null
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
------------------------
registering...
connecting...
connected.
done registering.
connection closed
```

Verify that pxGrid client iseSample shows up as a registered client under pxGrid Services
References

How-to-Configure and Test pxGrid-


Using Self-Signed Certificates with pxGrid ISE node and pxGrid clients

Using Self-Signed Certificates with pxGrid ISE node and CA-signed pxGrid clients

Using CA-Signed Certificates with pxGrid ISE node and pxGrid clients
Appendices

Alternative way of generating pxGrid node Certificate Signing Request (CSR)

Here is an alternative of way of generating pxGrid node CSR requests. The primary PAN will generate the primary pxGrid CSR request for the primary PAN and primary MNT nodes for pxGrid operation. The primary PAN will also generate the secondary pxGrid CSR request for the secondary PAN and secondary MNT nodes for pxGrid active-standby configuration.

This section describes an alternative for exporting the public/private key pair from the primary pxGrid node and importing into the Primary PAN and the Primary Mnt node for pxGrid operation.

Step 1 Administration->System->Certificates->Generate Certificate Signing Request (CSR) for “pxGrid” usage for both the Primary PAN and the Primary Mnt nodes. Provide the Common Name (CN) and (DNS) values, then->Generate

Step 2 You will be prompted to download a csrs.zip file containing the CSR requests for the Primary PAN and Primary Mnt nodes

Step 3 Copy/Paste the Primary PAN request into the Advanced Certificate Request pxGrid template and download the certificate

Step 4 Bind the certificate to the pan#1pxGrid node
Step 5  Copy/Paste the Primary MnT request into the Advanced Certificate Request pxGrid template and download the certificate

Step 6  Bind the certificate to the MnT1pxGrid node

Step 7  Administration->System->Certificates->Generate Certificate Signing Request (CSR) for “pxGrid” usage for both the Secondary PAN and the Secondary MnT nodes. Provide the Common Name (CN) and (DNS) values, then->Generate

Step 8  You will be prompted to download a csrs.zip file containing the CSR requests for the Secondary PAN and Secondary MnT nodes

Step 9  Copy/Paste the Primary PAN request into the Advanced Certificate Request pxGrid template and download the certificate

Step 10 Bind the certificate to the pan2pxGrid node

Step 11 Copy/Paste the Secondary MnT request into the Advanced Certificate Request pxGrid template and download the certificate

Step 12 Bind the certificate to the MnT2pxGrid node
Troubleshooting

This section provides information on troubleshooting.

Make sure the pxGrid client, pxGrid nodes and ISE are DNS resolvable if you see the following error message:

```
jeppich$ ./session_download.sh -keyStoreFilename mac.jks -keyStorePassword cisco123 -trustStoreFilename caroot1.jks -trustStorePassword cisco123 -hostname 10.0.0.48 10.0.0.49 -username mac
------- properties -------
version=1.0.0
hostnames=10.0.0.48, 10.0.0.49
username=mac
keyStoreFilename=mac.jks
keyStorePassword=cisco123
trustStoreFilename=caroot1.jks
trustStorePassword=cisco123
filter=null
start=null
end=null
------------------------
connecting...
connected.
at org.apache.cxf.interceptor.MessageSenderInterceptor$MessageSenderEndingInterceptor.handleMessage(MessageSenderInterceptor.java:64) ~[cxf-api-2.7.3.jar:2.7.3]
at org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:271) ~[cxf-api-2.7.3.jar:2.7.3]
at org.apache.cxf.jaxrs.client.AbstractClient.doRunInterceptorChain(AbstractClient.java:581) [cxf-rt-frontend-jaxrs-2.7.3.jar:2.7.3]
at org.apache.cxf.jaxrs.client.WebClient.doChainedInvocation(WebClient.java:904) [cxf-rt-frontend-jaxrs-2.7.3.jar:2.7.3]
at org.apache.cxf.jaxrs.client.WebClient.doInvoke(WebClient.java:772) [cxf-rt-frontend-jaxrs-2.7.3.jar:2.7.3]
at org.apache.cxf.jaxrs.client.WebClient.invoke(WebClient.java:355) [cxf-rt-frontend-jaxrs-2.7.3.jar:2.7.3]
at com.cisco.pxgrid.stub.identity.impl.SessionIteratorImpl.open(SessionIteratorImpl.java:128) [pxgrid-identity-client-stub-1.0.0.jar:1.0.0]
at com.cisco.pxgrid.samples.ise.SessionDownload.main(SessionDownload.java:132) [pxgrid-sdk-1.0.0.jar:1.0.0]
```

Removing Java and Installing JDK 8.0 on Centos 6.5

**Delete Old version of Java**

Step 1 Ensure Centos 6.5 is up-to-date, type: `yum update`
Note: you may need root privileges, type: `su root yum update`

Step 2 Once updates are completes, remove any other installed JAVA packages by typing:

```
rpm -qa | grep -E '^open[jre|jdk]|j[jre|dk]'  
```

Note: there was the java-1.6.0-openjdk-1.6.0.0-1.56.1.11.8.el6_3.i686 package already installed so I removed it by running)

Step 3 Type: `yum remove java-1.6.0-openjdk`

### Install JDK 8.0

Step 1 Change to root user, type: `su`, you will be prompted for your password.

Step 2 Install JDK 8, type: `rpm -Uvh jdk-8u20-linux-x64.rpm`

Step 3 You will also need to run the alternatives commands:

```
alternatives --install /usr/bin/java java /usr/java/latest/jre/bin/java 200000
alternatives --install /usr/bin/javaws javaws /usr/java/latest/jre/bin/javaws 200000
alternatives --install /usr/lib64/mozilla/plugins/libjavaplugin.so libjavaplugin.so.x86_64 /usr/java/latest/jre/lib/amd64/libnpjp2.so 200000
alternatives --install /usr/bin/javac javac /usr/java/latest/bin/javac 200000
alternatives --install /usr/bin/jar jar /usr/java/latest/bin/jar 200000
```

Check java version, type: `java --version`, you should see: java version “1.8.0_20”