



pxGrid Virtual Hosting Environment

Secure Access How -To Guides Series

Author: John Eppich

Date: November 2014

Table of Contents

Getting Started	4
About this Cisco Secure Access How-To Guides	4
Overview	4
Introduction	5
What is Cisco pxGrid?.....	5
Technical Overview	6
On-Premises Requirements	7
Physical Diagram	8
VMware vSphere Client POD-Access	9
OEAP AP 602 Physical Connection	10
Configuring OEAP AP	11
AP Local Provisioning	11
AP Cisco Provisioned.....	12
Linux Host Configuration	14
Linux Network Configuration	14
MAC Network Configuration.....	15
pxGrid SDK Installation	15
Installing Identity Certificate on Linux Client	16
Testing Authentications using RADIUS Simulator	18
Adding Linux Host as a Network Device	18
Creating an Internal User	18
User Authentication Test.....	19
RADIUS simulator PARAMETERS and Defaults	20
pxGrid Script Testing	21
Register	21
Verification	21
Definition	21
Session_Subscribe	22
Verification	22
Definition	22
Example	22
Session Download	24
Verification	24
Definition	24

Example	24
Identity Group Download	26
Verification	26
Definition	26
Example	26
EndPointProfileQuery	26
Verification	26
Definition	26
Example	26
CapabilityQuery	27
Verification	27
Definition	27
Example	27
SecurityGroupQuery	28
Verification	28
Definition	28
Example	28
Session_Group_Query_BY_IP	28
Verification	28
Definition	28
Example	29
EndpointProfile_Subscribe	29
Verification	29
Definition	29
Example	29
Security_Group_Subscribe.....	31
Verification	31
Definition	31
Example	31
Endpoint Protection Service (EPS).....	32
Verification	32
Definition	32
Example	32
Compliance Checks.....	35
Compliant.....	35

Getting Started

About this Cisco Secure Access How-To Guides

The Cisco Secure Access team is producing this series of How-To documents to describe best practices for Cisco Secure Access deployments. The documents in the series build on one another and guide the reader through a successful implementation of the Cisco Secure Access solution. You can use these documents to follow the prescribed path to deploy the entire system, or simply pick the single use-case that meets your specific need.

Overview

This document provides partners, Cisco field engineers and TME's with a procedural document and test guide for accessing and testing pxGrid sample scripts within the pxGrid virtual environment. A Linux host and Cisco Office Extended Access Point (OEAP) are required on-premises. The Linux host will serve as the pxGrid client and connect the virtual ISE pxGrid server through a remote LAN port on the Cisco OEAP AP. Users can use this document to run the pxGrid sample scripts directly on their Linux host. Users can view the script results locally including user authentications, pxGrid operations, such as client registrations and session subscription to topics of information on ISE.

The pxGrid sample scripts and examples enable partners to become familiar with pxGrid operation. Java scripts expose the pxGrid APIs where developers can build on these scripts and create customized scripts for their environment.

Basic pxGrid operations include:

- pxGrid client registration
- Retrieval of all active sessions based on IP address or all active sessions from ISE published session directory
- Downloadable user and identity groups
- Retrieval of all endpoint profiles
- Retrieval of all Trustsec Security Groups
- Using Endpoint Protection Service (EPS) pxGrid API for quarantining actions for a given IP address
- Using Endpoint Protection Service (EPS) pxGrid API for unquarantining actions for a given MAC address
- Subscription to ISE session directory and changes in Trustsec Security Groups

In addition posture, compliance and endpoint profile session attributes will be obtained by running 802.1X user authentications from a Window 7 VMware POD.

Introduction

What is Cisco pxGrid?

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

Cisco pxGrid provides a unified framework that enables ecosystem partners to integrate to pxGrid once, and then share context uni- or bi-directionally with many platforms without the need to adopt platform APIs. pxGrid is fully secured and customizable, enabling partners share only what they want to share and consume only context relevant to their platform.

Key features of pxGrid include:

- **Control what context is shared and with which platforms** - pxGrid is customizable and partners can “publish” only the specific contextual information they want to share and can control the partner platform where this shared.
- **Bidirectional context sharing** – pxGrid allows platforms to share or publish context as well as consume or “subscribe to” context from specific platforms. These features are orchestrated and secured by the pxGrid controller.
- **Share context data in native formats** - Contextual information shared via pxGrid is done in each platform’s native data format.
- **Connect to multiple platforms simultaneously**- pxGrid enables platforms to publish only the context data to relevant partner platforms. Numerous context “topics” may be customized for a variety of partner platforms, yet always shared via the same reusable pxGrid framework. Furthermore, only sharing relevant data enables both publishing and subscribing platforms to scale their context sharing by eliminating excess, irrelevant data.
- **Integration with Cisco platforms** - pxGrid provides a unified method of publishing or subscribing to relevant context with Cisco platforms that utilize pxGrid for 3rd party integrations.

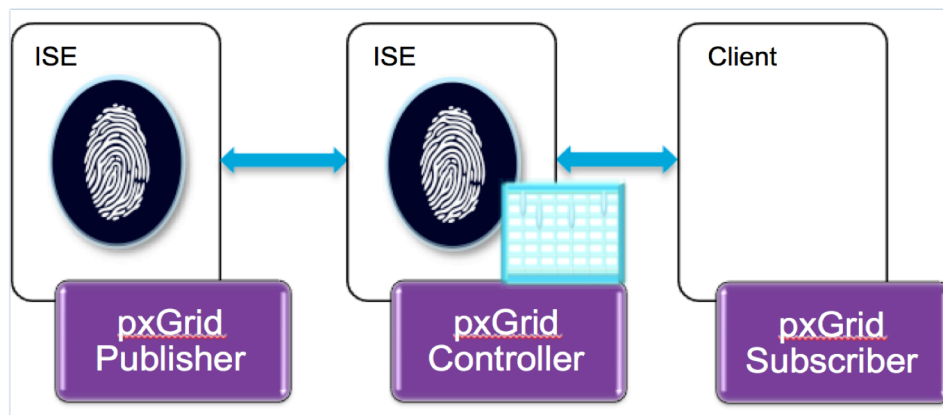


Figure 1. pxGrid Controls

Technical Overview

The pxGrid controller manages all client authentications, authorizations, capabilities/topics and their subscription list. The pxGrid controller controls all control aspects of the client communication (including management) and other participating client with mutual trust and authorization's enforcement.

The pxGrid publisher provides topics of interest or capabilities which the pxGrid subscriber or pxGrid client consumes or "subscribes" to. The client registers as an authorized session, EPS (Endpoint Protection Service), or Basic group to the pxGrid controller for authentication and authorization. This occurs through mutual authentication. Once the client is authorized, it can then subscribe to these capabilities and obtain the contextual information based on the pxGrid API scripts.

In most cases the client will register to either a session group or EPS group. The session group allows the client to subscribe to the different capabilities. The EPS group is a superset group to session and allows the client to subscribe to the EndpointProtectionService Capability taking EPS action. The basic group will not entitle the client to do anything on pxGrid, and needs manual authorization from the pxGrid administrator to be placed in an authorized session. The administrator group cannot be set and is reserved for ISE nodes such as the MnT Publisher and PAP UI. In addition, it can perform the other actions from the other groups.

ISE 1.3 will ship with pre-canned capabilities:

- **SessionDirectory**- exposes the existing attributes in the ISE Session directory for pxGrid session objects:
 - ◆ Session State, IP Address, Username, User AD domain, MAC, NAS IP Address, Trustsec Security Group Name, Endpoint Profile Name, (profiling policy name), Posture Status, Audit Session ID, Acct Session IP (In the RADIUS AV Pair, Last Update Time)
- **EndpointProfileMetadataCapability**- exposes the profiling policies in ISE. Any addition/deletion/update in these policies are notified through this capability:
 - ◆ id/name/fully-qualified name
- **TrustsecMetadataCapability**- exposes the Trustsec Security Groups metadata configured in ISE:
 - ◆ Trustsec tag name, unique identifier, description and value
- **EndpointProtectionService**- exposes the EPS quarantine/unquarantine APIs

The pxGrid client will be the Linux host and connect and register with the ISE pxGrid controller. This is enabled as a pxGrid persona on the ISE primary node. Once the pxGrid client has successfully registered, Java sample scripts will be from the Linux host targeting authenticated user session. These scripts uncover the pxGrid APIs and the results of these scripts are in real-time and can be seen as notifications. This is illustrated throughout this document along with the pxGrid sample scripts.

On-Premises Requirements

This section lists the on-premises requirements.

The partner must provide Cisco with the hostname of the Linux host, so this can be added to the Cisco Virtual Environment's DNS server. The FQDN and Cisco DNS server information will be returned to the partner and added to their Linux host configuration.

The Cisco WLC Controller IP address will also be provided to the partner for their OEAP WLC controller IP address.

On-Premises Requirements

- Linux host for running pxGrid GCL (Grid Control Libraries) libraries
- Firefox Brower for ISE access
- Java Development Kit ([JDK](#)) for Linux Operating System

Note: Centos 64-bit is used in this document. Any Linux operating system can be used providing Java Development Kit 7. MAC using OSX 10 and JDK 8.0 is also supported. Linux on Windows is not supported for running the pxGrid SDK libraries.

- [pxGrid-SDK-1.0.0-144.tar.gz](#) file

Note: This SDK will be use for the VM ISE pxGrid server. Select **Software->ISO&SDK->Archive->1.3.0.722->pxGrid-SDK-1.0.0-144.tar.gz file**. This file will be updated to reflect the ISE 1.3 build in the virtual environment. Partners will be notified of the upgrade

- Cisco Office Extended Access Point (OEAP) Access Point (Model AIR-OEAP602I-A-K9) with 7.6.120 Image
- CAPWAP UDP ports 5246 and 5247 must be opened between the AP and the Cisco WLC 2504 Wireless Controller.
- Ports TCP 443 must be opened for VMware's vSphere client to Cisco ESXi/ESX Host Management Connection
- PC running VMware vSphere client

Note: If the vSphere client is out of date, you can <https://> to one of the POD's in the vSphere Client access section. This will be used for accessing the Windows 7 VMware Pod for user authentication for obtaining posture, compliant and non-compliant session attributes.

Physical Diagram

The partner OEAP configuration and the Cisco DMZ Pod Configuration are shown in Figure 2. There are (5) PODs available with the same networking configuration and same VMware POD IP addresses.

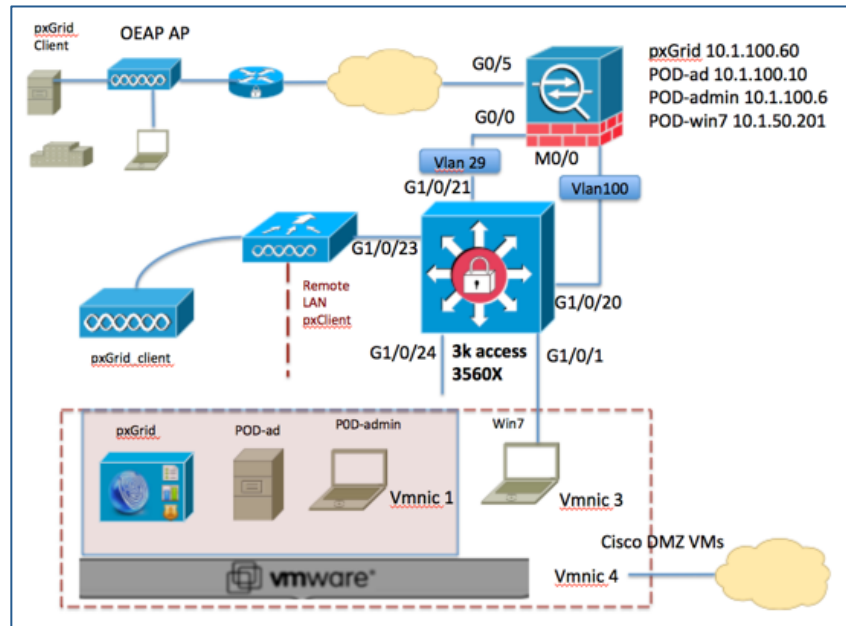


Figure 2. OEAP/CiscoDM2 Pod Configuration

The pxGrid client (Linux host) will connect to the Port 4 of the AP. The traffic is backhauled to the Cisco remote LAN pxClient, configured on the Cisco WLC controller. The pxGrid client have access to the VMware pxGrid ISE server and Windows 7 POD. The Windows 7 POD provides additional session information for posture, compliant and non-compliant attributes. The Linux host will be exclusively used for running the pxGrid sample script or running customized ones. You can access the VMware pxGrid ISE server through a Firefox browser.

A separate PC will connect to local port 2 and will be used for viewing the OEAP AP Event Log to ensure connection, and also for internet access. This PC will also be used to connect to the VM Management server in the associated PODs.

VMware vSphere Client POD-Access

The VMware vSphere client is used to access the Windows 7 workstation pod (Pod-Win7pc-corp). This will be used for obtaining 802.1X user session information for posture, compliance and profiled endpoint information.

The VM Management IP addresses for the POD's are as follows:

- POD 1 128.107.255.21
- POD 2 128.107.255.22
- POD 3 128.107.255.23

Select a VM Management IP address and launch the vSphere client and enter the username/password: root/ISEisC00L.

Note: Take down the POD information

Step 1 Select the POD_w7pc_corp and power-on

Step 2 Login as username/password: employee1/ISEisC00L

Note: User may already be logged into Windows

OEAP AP 602 Physical Connection

This lists the physical port connection of the OEAP 602

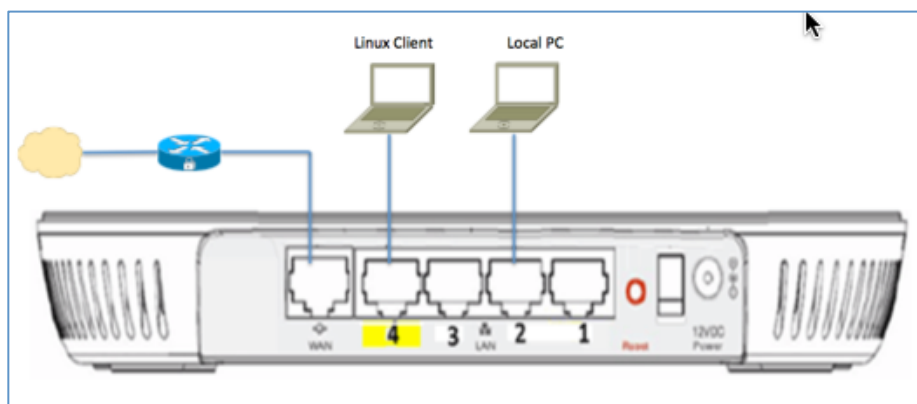


Figure 3. OEAP 602 Ports

WAN - connect to Router

Port 4 – Remote LAN Port – Linux Client – this will connect to **pxclient** remote LAN on the Cisco WLC

Port 2 – Connect to local client via Ethernet – this PC can be used to connect to the Cisco DMZ Win7 pod, and to view the Event_Log of the OEAP AP.

Note: Disable wireless on local client, otherwise you will receive “... overloading Ethernet port “ from the Cisco WLC controller and will not be able to connect

Configuring OEAP AP

The AP can either be provisioned locally or a provisioned AP can be requested from Cisco, pending availability. Local provisioning requires that you have a 2504 WLC or equivalent controller, installed data license and a 7.6.120 image. Once the AP has been provisioned with the proper image, you will need to change the WLC Controller IP address and choose a Cisco WLC Controller IP address from one of the PODs listed below or be assigned one from Cisco.

A Cisco provisioned AP contains the 7.6.120 image and you may need to change the provisioned local IP address to one that you can manage in your network.

Note: Do not use Port 4, as this will be used the remote LAN

Whether you provision your own AP locally or use a Cisco loaner AP, it must contain a 7.6.120 image and include a Cisco WLC Controller IP address.

The Cisco WLC IP address for the PODs are listed below:

- POD 1 128.107.255.101
- POD 2 128.107.255.102
- POD 3 128.107.255.103

AP Local Provisioning

Step 1 Connect a PC to any of the local ports

Note: Make sure the WAN port is connected to the Router, remember not to use Port 4 since this is reserved for the remote LAN port.

Step 2 Open your browser to <http://10.0.0.1>, the default IP address of the OEAP AP

Step 3 Under the **DHCP** Configuration, change the local IP address from 10.0.0.1 to one that you can manage from your network.

Note: You will lose the connection after this change, connect back using the local IP Address

Step 4 Change the default username and password from: admin, admin to ones that are more secure.

Step 5 Under the **WAN** Configuration, change the WLC controller IP address to your local WLC IP address

Note: The uplink port may need to be configured, to initially include the DNS information and actual AP IP address.

Step 6 The AP should successfully join the WLC and start downloading the 7.6.120 image.

Step 7 Once the AP has the 7.6.1.20 image, change the controller IP to a Cisco WLC DMZ controller IP address

Note: The WLC IP Address will be provided by Cisco

Step 8 Connect the Linux host to Port 4.

If everything is configured successfully you should see the following:

```
.
.
*Aug 23 17:45:01.129: Change State Event Response from 128.107.255.103
*Aug 23 17:45:01.215: Change State Event Response from 128.107.255.103
*Aug 23 17:45:01.300: Change State Event Response from 128.107.255.103
```

```
*Aug 23 17:45:01.397: SSID pxtest, WLAN Profile Name: pxtest, added to the slot[0], disabled
*Aug 23 17:45:01.765: SSID pxtest, WLAN Profile Name: pxtest, added to the slot[1], enabled
*Aug 23 17:50:11.914: Re-Tx Count= 0, Max Re-Tx Value=5, NumofPendingMsgs=1

*Aug 23 18:09:46.014: Ethernet Backhaul WLAN ID = 2,qos=3,split-tunnel=0
```

Note: if you receive repeated discovery notifications, it is possible the WLC is down, or your DNS may not be working properly.

AP Cisco Provisioned

If you received a Cisco OEAP AP, you may need to change the local IP address and DHCP information under DHCP Configuration. The Cisco WLC Controller IP address may already have been entered under WAN configuration, or you may choose one from the available PODs.

- Step 1** Connect a PC to any of the local ports.
- Step 2** Open your browser and `http://10.0.0.1`, and type: `admin`, `admin` for username and password, and click **Apply**.
- Step 3** Under the DHCP Configuration, change the local IP address from `10.0.0.1` to one that you can manage from your network.
- Step 4** Enable **DHCP**.
- Step 5** Click **Apply**.

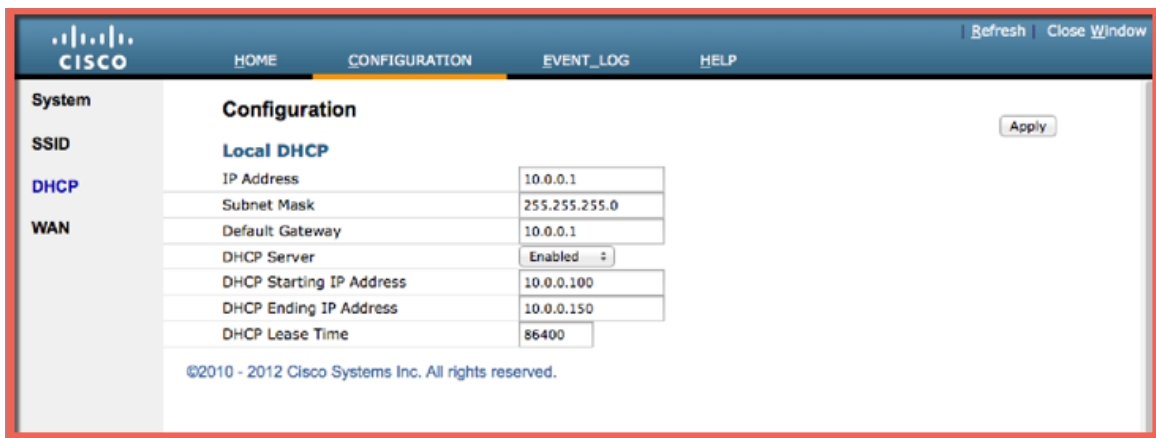


Figure 4. OEAP Configuration

Note: Some routers may use `10.0.0.1`, you may need to change this to reflect an IP address in your network. Keep DHCP Server Enabled as you will connect the Linux host directly. Use the PC connected to any of your local ports to view the AP events under `EVENT_LOG`. If this PC has both wireless and a wired connection, disable the wireless connection as this will cause “flooding over the Ethernet port”, and you will not be able to connect to the Cisco WLC.

- Step 6** Under WAN Configuration, enter the WLC Controller IP address for the desired PODs, and click **Apply**.



The screenshot shows the Cisco WLC Configuration page. The navigation bar includes HOME, CONFIGURATION, EVENT_LOG, and HELP. The left sidebar lists System, SSID, DHCP, and WAN. The main content area is titled 'Configuration' and contains the following sections:

- Controller**: IP Address field with value 10.1.100.61.
- Uplink IP Configuration**:

Static IP	<input type="checkbox"/>
Domain Name	demo.local
IP Address	10.1.90.115
Subnet Mask:	255.255.255.0
Default Gateway	10.1.90.1
DNS Server	10.1.100.12

©2010 - 2012 Cisco Systems Inc. All rights reserved.

Figure 5. WLC Controller

Note: Unlink IP Configuration is not required

Step 7 Connect your Linux client into **Port 4**.

If everything works successfully, you should see the following under Event_Log:

```
.
.
*Aug 23 17:45:01.129: Change State Event Response from 128.107.255.103
*Aug 23 17:45:01.215: Change State Event Response from 128.107.255.103
*Aug 23 17:45:01.300: Change State Event Response from 128.107.255.103
*Aug 23 17:45:01.397: SSID pctest, WLAN Profile Name: pctest, added to the slot[0], disabled
*Aug 23 17:45:01.765: SSID pctest, WLAN Profile Name: pctest, added to the slot[1], enabled
*Aug 23 17:50:11.914: Re-Tx Count= 0, Max Re-Tx Value=5, NumofPendingMsgs=1

*Aug 23 18:09:46.014: Ethernet Backhaul WLAN ID = 2,qos=3,split-tunnel=0
```

Note: if you keep receiving discovery notifications, it may be possible that the WLC is down, or your DNS may not be working properly.

Linux Host Configuration

Linux Network Configuration

This section provides Linux host network configuration details. The network IP information is provided from Cisco. Below is an example using **centos.demo.local** as the host name and a IP address of 10.1.100.30.

Note: When making changes, ensure that you are root. The IP address, FQDN and nameserver will be provided from Cisco.

Step 1 Edit the host name to include the FQDN use vi or other editor to make changes.

```
vi /etc/hosts
10.1.100.30 centos.demo.local
```

Step 2 Add the **nameserver** to **/etc/resolv.conf**

```
vi /etc/resolv.conf
search demo.local
nameserver 10.1.100.10
```

Step 3 Configure Networking, on Centos, this accomplished via **System->Preferences->Network Connections**. The network connection parameters should be statically defined:

Note: Network Connection setting will vary across Linux platforms.

```
IP Address: 10.1.100.30
Netmask: 255.255.255.0
Gateway: 10.1.100.1
DNS Server: 10.1.100.10
Search Domain: demo.local
```

Step 4 To view network settings for eth0 for Centos-64 enter:

```
cat /etc/sysconfig/network-scripts/ifcfg-eth0
```

Step 5 To test the pxGrid client connection, ping the IP address the Cisco DMZ gateway enter:

```
ping 10.1.100.1
```

Step 6 Ping the IP address of ISE pxGrid server:

```
ping px.demo.local
```

Pinging both the Cisco DMZ gateway and the ISE pxGrid server, ensure that you have a successful remote LAN connection between the Linux host and ISE.

MAC Network Configuration

This section provides the MAC network configuration details. The network IP information is provided from Cisco. Below is an example using **mac.demo.local** as the host name and an IP address of 10.1.100.62.

Note: The IP address, FQDN, and DNS server will be provided by Cisco.

Step 1 Edit the **hostname** to include the **FQDN** using nano:

```
nano /etc/hosts
10.1.100.62 mac.demo.local
```

Step 2 Select **System Preferences-> Network**, to configure networking on the MAC. The network connection parameters should be statically defined:

```
Configure IPv4:Manually
IP Address: 10.1.100.62
Netmask: 255.255.255.0
Router: 10.1.100.1
DNS Server: 10.1.100.10
Search Domains: demo.local
```

Note: By adding the DNS server to the MAC network connection, the nameserver should show up when running `cat /etc/resolv.conf`

Step 3 To test the pxGrid client connection, ping the IP address of the Cisco DMZ gateway:

```
ping 10.1.100.1
```

Step 4 Ping the IP address of ISE pxGrid server

```
ping px.demo.local
```

Pinging both the Cisco DMZ gateway and the ISE pxGrid server, ensure that you have a successful remote LAN connection between the MAC and ISE.

pxGrid SDK Installation

It is assumed that both the Linux operating system and [Java Development Kit](#) are installed. If using a MAC, please download the Mac OS X x64 Oracle Development Kit (`jdk-8u20-macosx-x64.dmg`). This section covers the initial installation of the pxGrid SDK.

When the JDK is installed, the Java Runtime Engine (JRE) will be installed as well as part of the initial install. The JRE path must be included in the export `JAVA_HOME` path. (*Example:* `export JAVA_HOME=/usr/java/jdk1.7.0_51/jre`).

```
export JAVA_HOME=/usr/java/jdk1.7.0_51/jre
```

If running MAC on OSX 10, the JRE path will be:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/jre
```

To check if you have the JDK installed, type the following:

```
java -version
sudo find / -name java
```

You will be required to enter your privileged root password.

Step 1 To untar the file, run

```
tar xzf pxgrid-sdk-1.0.0-alpha-xxx-dist.tar.gz
```

Step 2 You will see the following in the pxgrid-sdk-1.0.0-alpha-xxx-dist:

Lib- contains all the GCL libraries

Samples- contains: bin, certs, conf, lib, src directories

Bin- contain all the sample pxGrid scripts

Certs- contains all the sample keystore and rootSample certificates

Src- contains all the java source code

README.txt- provides definition of sample shell scripts, GCL, openssl and keytool instructions.

Please read the README.txt file for each SDK this will contain important information such as certificate installation on the Linux host.

Installing Identity Certificate on Linux Client

This section exports the ISE identity certificate from the pxGrid ISE server to the Linux host. This process is the same on MACs. This certificate is used for bulk session downloads. Use the Firefox browser from your Linux host.

Step 1 Select **Administration->System->Certificates**

Step 2 Default **Signed-Certificate**.

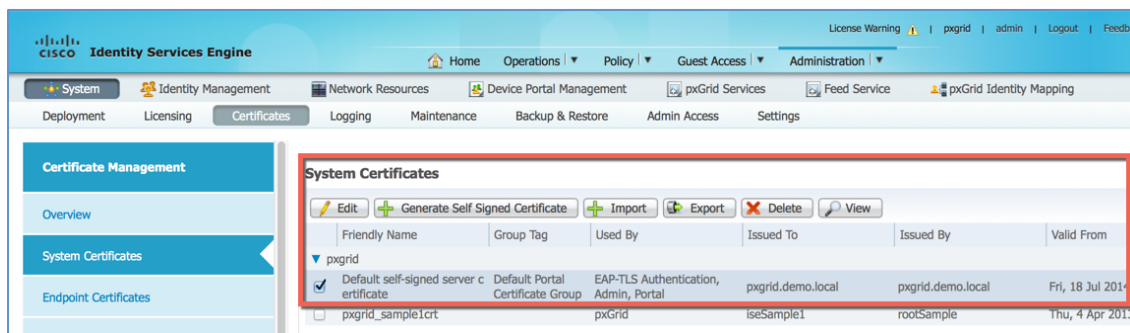


Figure 6. System Certificates

Step 3 Export

Step 4 Export the 'Default self-signed server certificate', export **certificate only**

- Step 5** Save the Defaultselfsignservercerti.fpem file locally.
- Step 6** If using the Linux host client browser, save locally to Linux host and leave in Downloads folder.
- Step 7** Rename or copy the certificatename to pxgridmnt.pem

```
cp Defaultselfsignservercert.pem pxgridmnt.pem
```

- Step 8** Perform the following:

```
$openssl x509 -outform der -in pxgridmnt.pem -out pxgridmnt.der  
$keytool -import -alias mnt -keystore rootSample.jks -file pxgridmnt.der
```

Note: keytool may not appear in the path, as in the case of Centos 64-bit. Append the `../jdk1.7._51/bin` to path. (i.e. `export PATH=/usr//lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/jeppich/bin:/usr/java/jdk1.7.0_51/bin`)

To view the path, type: **echo \$PATH.**

For further reference, please refer to the SDK README.txt file

- Step 9** Enter: **cisco123** for keystore password.
- Step 10** Enter: **yes** for trust this certificate. You should see: Certificate was added to keystore.

You should now have the following: pxgridmnt.der, pxgridmnt.pem, rootSample.jks certificates in your folder.

- Step 11** Copy all four of these certificates to the “../pxgrid-sdk-1.0.0-alpha-xxx/samples/certs” directory.

```
Copy all four of these certificates to the “../pxgrid-sdk-1.0.0-alpha-xxx/samples/certs”  
directory.
```

- Step 12** Copy these certificates and the iseSample1.jks certs to the “../pxgrid-sdk-1.0.0-alpha-xxx/samples/bin” directory:

```
pxgridmnt.der  
pxgridmnt.pem  
iseSample1.jks  
rootSample.jks
```

Testing Authentications using RADIUS Simulator

RADIUS simulator provides 802.1X authentications and allows for the population of basic attributes such as IP, MAC, and identity group information into the Session Directory. RADIUS simulator will be run on the Linux host. Here we define an internal user, user1, which will be used for user authentication. Prior to running RADIUS simulator, the Linux PC running RADIUS simulator will be added to the Network Devices list in ISE.

In addition, RADIUS simulator has command-line arguments as defined in RADIUS simulator PARAMETERS list.

The command-line arguments: `-DUSERNAME`, `-DPASSWORD`, `-DCALLING_STATION_ID`, `-DAUDIT_SESSION_ID`, `-DFRAMED_IP_ADDRESS`, `-DFRAMED_IP_MASK`, `RadiusAccountingStart`, `RadiusAccountingStop`, and `RadiusAuthentication` will be used for multiple user testing. Each user will have their independent session information.

Note: RADIUS simulator command-line arguments are case-sensitive.

Adding Linux Host as a Network Device

- Step 1** Administration->Network Resources->Network Devices->Add enter name and IP address of Linux Host
- Step 2** Enable Authentication Settings->Shared Secret, type: secret
- Step 3** Submit

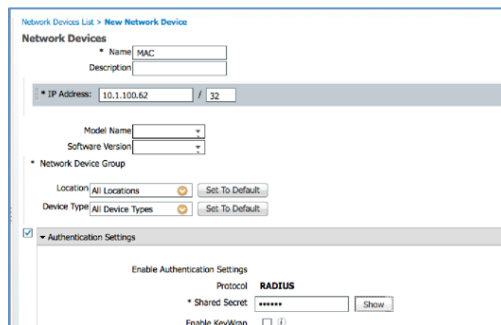


Figure 7. Network Device Configuration

Creating an Internal User

An ISE internal user will be created and assigned to the Employee group which will be used for 802.1X authentication and for obtaining identity group information when running the pxGrid scripts.

- Step 1** Select->Administration->Identity Management->Identities->Users

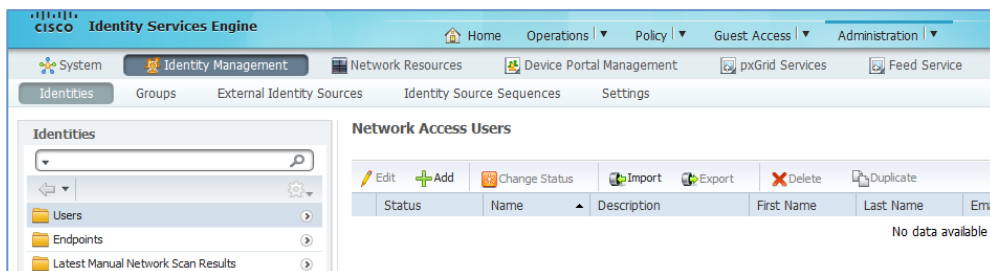


Figure 8. Identity Management

- Step 2** Select -> Add and type: **user1** for the Name, and type: **Aa1234567** for the Password.

Note: Instead of user1, provide a **partner name_User1** name, (i.e. **Eppich_User1**), since the ISE pxGrid server will be used by multiple partners.

Step 3 Assign user1 to the Employee User Group

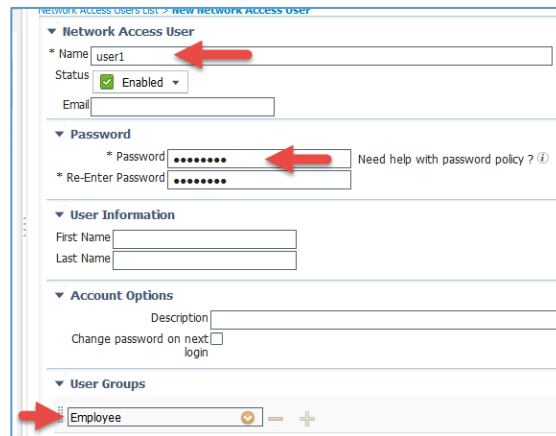


Figure 9. Create User

Step 4 Click **Submit**.

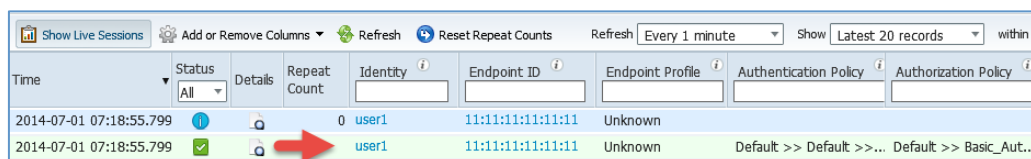
User Authentication Test

Step 1 On the Linux host, type the following:

```
java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -
DCALLING_STATION_ID=11:11:11:11:11:11 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=192.168.1.40 -
DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.98
```

Note: If you only see a blinking cursor after typing the RadiusSimulator command, check Network Devices, and ensure that the host's physical IP address is present.

Step 2 To verify authentications from ISE, **Operations->Authentications**, user1 has successfully authenticated



Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy
2014-07-01 07:18:55.799			0	user1	11:11:11:11:11:11	Unknown		
2014-07-01 07:18:55.799				user1	11:11:11:11:11:11	Unknown	Default >> Default >>... Default >> Basic_Aut...	

Figure 10. User Authorization

Step 3 Click on **Add or Remove Columns** to enable the **IP Address**, **Security Group**, and **Session ID** columns in ISE.

Note: This provides more authenticated session details that will come in handy when running the pxGrid scripts.

Step 4 Click **Save**.

RADIUS simulator PARAMETERS and Defaults

If required, Radius simulator parameters can be set using the following defaults.

Table 1. RADIUS simulator Parameters

Parameters	Defaults
USERNAME	user1
PASSWORD	Lab123
CALLING_STATION_ID	00:11:22:33:44:55
AUDIT_SESSION_ID	101
RADIUS_SECRET	Secret
NAS_IP_ADDRESS	10.0.0.1
FRAMED_IP_ADDRESS	1.2.3.4
FRAMED_IP_MASK	255.255.255.0
NAS_PORT	
NAS_PORT_ID	
NAS_PORT_TYPE	

Note: When using multiple users for testing, please use: -DCALLING_STATION_ID, -DAUDIT_SESSION_ID, -DFRAMED_IP_ADDRESS, -DFRAMED_IP_MASK parameters. By default, if you do not specify the -DFRAMED_IP_ADDRESS, -DFRAMED_IP_MASK parameters, the IP address will remain 1.2.3.4.

pxGrid Script Testing

The pxGrid sample scripts provide a good reference of available session information and available queries through the pxGrid APIs. Developers can modify these scripts to provide to subscribe or query relevant session information.

RADIUS simulator will be used for authenticating user and generating active user sessions. This will be run from the Linux host. For posture, compliant, non-compliant, EPS, endpoint profile session attribute information, the VMware Pod-win7-corp-pc will be used.

Below is a brief description of the sample test scripts:

- Register – registers pxGrid client to the pxGrid controller to an authorized **session** or **EPS** group for bulk session queries/download or published topic subscription.

Note: In this document, the Linux host will be the pxGrid client.

- Session_Subscribe – subscribe to changes in the session state
- Session_Download – download all active sessions from ISE
- Identity_Group_Download – download user and identity groups associated with active sessions in ISE
- Capability – lists all the capabilities or published topics supported by the instance of pxGrid that the pxGrid client will subscribe to.

Note: The pxGrid client will subscribe the SessionDirectory, EndpointProtectionService, TrustsecMetadata capabilities in these examples.

- SecurityGroup_Query- retrieve all Security Groups Tags in ISE
- Session_Query_By_IP – retrieve all active sessions from ISE based on IP address
- EndpointSecurityGroup_Query – retrieve all Trustsec Security Groups configured in ISE
- EndpointProfile_Query – retrieve all endpoint profiles (profiling policies) configured in ISE
- SecurityGroup_Subscribe – subscribe to changes in the Trustsec security groups configured in ISE
- Eps_quarantine – executes the Endpoint Protection Service (EPS) quarantine action on ISE for a given IP address
- Eps_unquarantine - executes the Endpoint Protection Service (EPS) unquarantine action on ISE for a given MAC address

Register

Verification

This test verifies that the 3rd party system can register, i.e. authenticate and be authorized, on the pxGrid.

Definition

PxGrid Client registration connects and registers the 3rd party application, security device, or in this case, the Linux host to the pxGrid controller, to an authorized session as either a **session** or **EPS** group. **Admin** groups are reserved for ISE. **Basic** groups require pxGrid administration approval and then can be manually be moved to an authorized group.

All registered pxGrid clients can be viewed in the in the ISE admin GUI under the pxGrid services view.

pxGrid clients can be publishers or subscribers of information. In ver 1.0 of pxGrid, the pxGrid clients will be subscribers. Throughout this document, all registered pxGrid clients will be subscribers of the ISE published session directory information. One the pxGrid client has successfully registered and authorized the group, the client can then obtain the relevant session information or queries as determined by the pxGrid sample scripts

Example

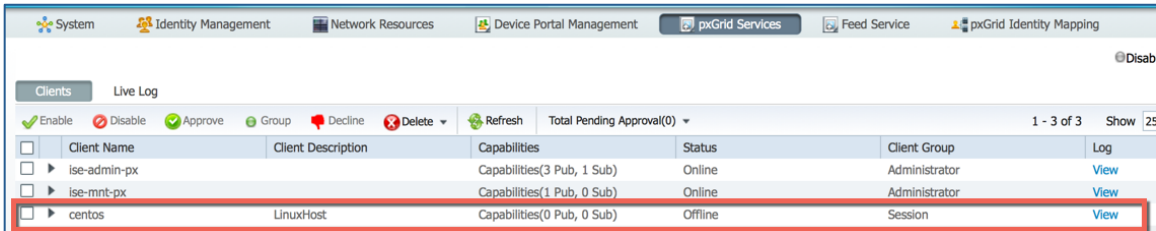
In this example, we will register the Linux host as a pxGrid client with the xgridUsername of centos

Step 1 Type the following on the Linux host:

```
./register.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -truststoreFilename
rootSample.jks -xgridDescription linuxhost -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60 -group Session
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
xgridDescription=linuxhost
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
registering...
connecting...
connected.
done registering.
connection closed
[root@centos bin]#
```

Step 2 Select **Administration->pxGrid Service**

Note the registered pxGrid client has registered to the Session Group in ISE



Client Name	Client Description	Capabilities	Status	Client Group	Log
ise-admin-px		Capabilities(3 Pub, 1 Sub)	Online	Administrator	View
ise-mnt-px		Capabilities(1 Pub, 0 Sub)	Online	Administrator	View
centos	LinuxHost	Capabilities(0 Pub, 0 Sub)	Offline	Session	View

Figure 11. pxGrid

Session_Subscribe

Verification

This test verifies that the 3rd party system is connected to the pxGrid table and it can subscribe to topics of information available in pxGrid. In this case, the pxGrid client will subscribe to updates to the user authentication status.

Definition

Once the pxGrid client has successfully registered and authorized to the session group by the pxGrid controller, the client will subscribe to published topics of information or capabilities, and obtain relevant session information for the authenticated user. The ISE MnT node will publish the ISE Session Directory as a topic to the pxGrid controller, the pxGrid client, Linux host, will subscribe to this capability and obtain the authenticated user's active sessions.

Example

In this example, we will use RADIUS simulator to terminate and start an authenticated user session. We should be able to see the Session state notifications appear in real-time under session notifications under the running script on the Linux host.

Step 1 Type the following on the Linux host:

```
./session_subscribe.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridHostname 10.1.100.60 -
xgridHostname 10.1.100.60 -xgridUsername centos
```

Step 2 You should see the following:

```
--properties--
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
filter=null
-----
connecting...
connected.
```

Step 3 Administration->pxGrid Service

Note that the pxGrid client has successfully subscribed to the Session Directory

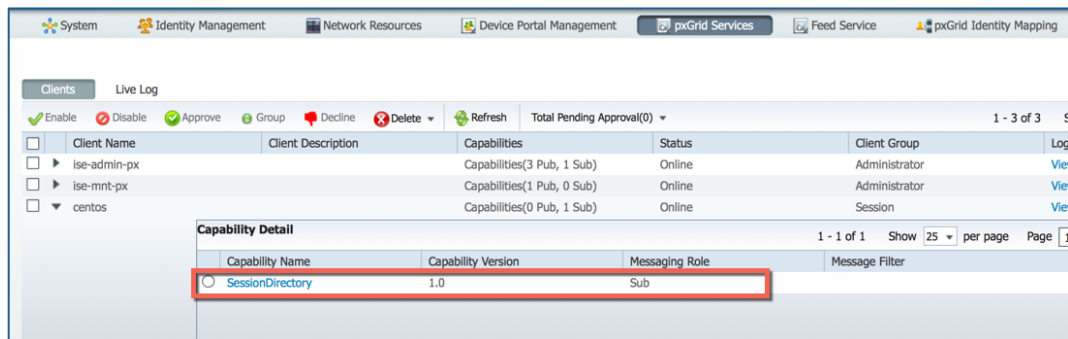


Figure 12. Session Director

Step 4 Open another terminal window on the Linux host.

Step 5 Run RADIUS simulator to start the RADIUS accounting session:

```
java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa1234567 -
DCALLING_STATION_ID=00:11:22:33:44:55 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=10.1.100.99 -
DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStart 10.1.100.60
```

Upon successful completion of RADIUS simulator, you should see:

```
code=5 id=1 length=20
authenticator=928de236dae9f7568750d5757398f99
Attributes={
}
```

Step 6 Run RADIUS simulator to terminate the RADIUS accounting session:

```
java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa1234567 -  
DCALLING_STATION_ID=00:11:22:33:44:55 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=10.1.100.99 -  
DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStop 10.1.100.60
```

Upon successful completion of the RADIUS script, you should see:

```
code=5 id=1 length=20  
authenticator=6da5fdc3c6b9f46261fc7093e9ae333  
Attributes={  
}
```

Step 7 Note the Session states in the below session notifications:

```
session (ip=10.1.100.99, Audit Session Id=1001, User Name=user1, Domain=null, Calling station  
id=00:11:22:33:44:55, Session state= STARTED, Epsstatus=null, Security Group=null, Endpoint  
Profile=Unknown, NAS IP=10.1.100.30, RADIUSAVPairs=[ Acct-Session-Id=123], Posture Status=null,  
Posture Timestamp=, Session Last Update Time=Thu Aug 14 06:03:54 EDT 2014)  
  
session notification:  
session (ip=10.1.100.99, Audit Session Id=1001, User Name=user1, Domain=null, Calling station  
id=00:11:22:33:44:55, Session state= DISCONNECTED, Epsstatus=null, Security Group=null, Endpoint  
Profile=Unknown, NAS IP=10.1.100.30, RADIUSAVPairs=[ Acct-Session-Id=123], Posture Status=null,  
Posture Timestamp=, Session Last Update Time=Thu Aug 14 06:13:55 EDT 2014)
```

Session Download

Verification

This test verifies the ability of the 3rd party system to execute bulk session downloads on context information.

Definition

The session download script downloads bulk session records from the published ISE node.

Example

In this example, the pxGrid client will download session records from the ISE. We are looking for a Security Group Tag of HR as contained in the active user session information. RADIUS simulator will be used to start the RADIUS Session and for user authentication.

Step 1 Type the following on the Linux host to start the RADIUS session:

```
java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa1234567 -  
DCALLING_STATION_ID=00:11:22:33:44:55 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=10.1.100.99 -  
DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStart 10.1.100.60
```

Upon successful completion of RADIUS simulator you should see:

```
java -cp  
code=5 id=1 length=20  
authenticator=928de236dae9f7568750d5757398f99  
Attributes={  
}
```


Step 2 Type the following on the Linux host to authenticate user1

```
java -cp RadiusSimulator -DUSERNAME=user1 -DPASSWORD=Aa1234567 -
DCALLING_STATION_ID=00:11:22:33:44:55 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=10.1.100.99 -
DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 10.1.100.60
```

Upon successful completion of RADIUS simulator you should see:

```
code=2 id=1 length=140
authenticator=ab5e99d2e74619c3845e445471c454e
Attributes={
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:px/196703053/162
  vendorId=9 vsa=[cts:security-group-tag=0002-0,]
  vendorId=9 vsa=[profile-name=Unknown,]
```

Step 3 Open another terminal window

Step 4 Type the following on the Linux host:

```
./ session_download.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridHostname 10.1.100.60 -
xgridUsername centos
```

Upon successful completion of the script you should see:

```
--properties-
version=1.0.0-alpha-144
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
filter=null
-----
connecting...
connected.
started at Thu Aug 14 16:34:49 EDT 2014.
```

Step 5 Note the Security Group Tag of HR for user1 in the below active session

```
session (ip=10.1.100.99, Audit Session Id=1001, User Name=user1, Domain=null, Calling station
id=00:11:22:33:44:55, Session state= AUTHENTICATED, Epsstatus=null, Security Group=HR, Endpoint
Profile=Unknown, NAS IP=10.1.100.30, RADIUSAVPairs=[ Acct-Session-Id=null], Posture Status=null,
Posture Timestamp=, Session Last Update Time=Thu Aug 14 09:04:04 EDT 2014 )

session (ip=10.1.50.201, Audit Session Id=0A0164010000001D101F2A8C, User Name=employee1,
Domain=demo.local, Calling station id=00:50:56:B8:40:0C, Session state= STARTED, Epsstatus=null,
Security Group=null, Endpoint Profile=null, NAS IP=10.1.100.1, NAS Port=GigabitEthernet0/1,
RADIUSAVPairs=[ Acct-Session-Id=00000052], Posture Status=Compliant, Posture Timestamp=Sat Aug 09
08:52:43 EDT 2014)... ending at: Thu Aug 14 16:34:49 EDT 2014

-----

downloaded 2 sessions in 11 milliseconds
-----
```

Identity Group Download

Verification

This test verifies the ability of the 3rd party system to execute a bulk download of user identity information.

Definition

Identity Group download script downloads bulk session records of user group information and user-group mappings from the session directory. These groups include ISE identity groups and profiled groups.

Example

The Identity Group Download script downloads all group information. In this example, we use the identity group download script to download all the group information from the ISE MnT node publisher.

Step 1 Type the following on the Linux host:

```
./identity_group_download.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
starting at Fri Aug 15 14:15:15 EDT 2014...
user=employee1 groups=Workstation
user=00:50:56:B8:40:0C groups=Workstation
user=user1 groups=Unknown
... ending at: Fri Aug 15 14:15:15 EDT 2014

-----
downloaded 3 users in 19 milliseconds
-----

connection closed
[root@centos bin]#
```

EndPointProfileQuery

Verification

This test verifies the ability of the 3rd party system to retrieve all enabled profiles configured in ISE.

Definition

The endpointprofile_query script provides a query method to retrieve all enabled endpoint profiles configured in ISE and provides the endpoint profile id, name and fully qualified name. The subscriber will also be notified if an endpoint profile is added/updated/deleted in ISE.

Example

In this example, the endpointprofile script retrieves all the enabled profiles in ISE.

Step 1 Type the following on the Linux host:

```
./endpointprofile_query.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60 | more
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
Endpoint Profile : id=e5db02e0-1722-11e4-ba29-005056bf2f0a, name=Android, fqname
  Android
Endpoint Profile : id=e62e5300-1722-11e4-ba29-005056bf2f0a, name=Apple-Device, f
qname Apple-Device
Endpoint Profile : id=e68a0790-1722-11e4-ba29-005056bf2f0a, name=Apple-iDevice,
fqname Apple-Device:Apple-iDevice
Endpoint Profile : id=e6be36f0-1722-11e4-ba29-005056bf2f0a, name=Apple-iPad, fq
name Apple-Device:Apple-iPad
Endpoint Profile : id=e6eff550-1722-11e4-ba29-005056bf2f0a, name=Apple-iPhone, f
qname Apple-Device:Apple-iPhone
Endpoint Profile : id=e7281c50-1722-11e4-ba29-005056bf2f0a, name=Apple-iPod, fq
name Apple-Device:Apple-iPod
```

CapabilityQuery

Verification

This test verifies the ability of the 3rd party system to retrieve all the published capabilities in ISE.

Definition

The capability script retrieves all published topics of interest in ISE.

Example

The capability script retrieves information topics or capabilities clients can be publish or subscribe.

Step 1 Type the following on the Linux host:

```
./capability_query.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
capability=TrustSecMetaDataCapability-1.0, version=1.0
capability=EndpointProfileMetaDataCapability-1.0, version=1.0
capability=IdentityGroupCapability-1.0, version=1.0
capability=TDAnalysisServiceCapability-1.0, version=1.0
```

```
capability=NetworkCaptureCapability-1.0, version=1.0
capability=EndpointProtectionServiceCapability-1.0, version=1.0
capability=GridControllerAdminServiceCapability-1.0, version=1.0
capability=SessionDirectoryCapability-1.0, version=1.0
connection closed
```

SecurityGroupQuery

Verification

This test verifies the ability of the 3rd party system to retrieve all Security Groups Tags in ISE.

Definition

The security group query script exposes the Security Group Tags (SGT) configured in ISE through the TrustsecMetaDataCapability topic. It provides a query method to retrieve all the SGTs configured in ISE based on a unique id, security group tag value and description.

Example

In this example, the security group query script will download all the Security Group Tag contextual information

The Security Group Query script retrieves all Trustsec Security Groups session information from ISE. This includes the Trustsec tag name, unique identifier, description and value.

Step 1 Type the following on the Linux host:

```
./securitygroup_query.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
SecurityGroup : id=eabff3a0-1723-11e4-ba29-005056bf2f0a, name=Unknown, desc=Unknown Security
Group, tag=0
SecurityGroup : id=eadb43d0-1723-11e4-ba29-005056bf2f0a, name=ANY, desc=Any Security Group,
tag=65535
SecurityGroup : id=e1166df0-23af-11e4-ba8d-005056b8b604, name=HR, desc=, tag=2
connection closed
[root@centos bin]#
```

Session_Group_Query_BY_IP

Verification

This test verifies the ability of the 3rd party system to execute a directed query regarding a specific IP address via pxGrid.

Definition

The Session Query by IP script obtains the authenticated user's session information by IP Address.

Example

In this example, we obtain employee1's session information or contextual information by entering the IP address of the user

Step 1 Type the following on the Linux host and enter the IP address

```
./session_query_by_ip.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridHostname px.demo.local -
xgridUsername mac_client
----- properties -----
version=1.0.0-alpha-144
xgridHostname=px.demo.local
xgridUsername=mac_client
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
ip address (or <enter> to disconnect): 10.1.50.201

session (ip=10.1.50.201, Audit Session Id=0A01640100000027585C723A, User Name=employee1,
Domain=demo.local, Calling station id=00:50:56:B8:40:0C, Session state= STARTED, Epsstatus=null,
Security Group=null, Endpoint Profile=Windows7-Workstation, NAS IP=10.1.100.1, NAS
Port=GigabitEthernet0/1, RADIUSAVPairs=[ Acct-Session-Id=00000063], Posture Status=Compliant,
Posture Timestamp=Sat Aug 23 09:05:22 EDT 2014, Session Last Update Time=Sat Aug 23 09:05:45 EDT
2014 )
ip address (or <enter> to disconnect):
connection closed
```

EndpointProfile_Subscribe

Verification

This test verifies the ability of the 3rd party system to subscribe to the published Endpoint Profile topic.

Definition

The registered pxGrid client will subscribe to the EndpointProfileMetaData capability to obtain changes or modifications in the global profiling policy. Session notifications will include the Endpoint profile id, name, and fully qualified name.

Example

In this example, a pxGrid EndpointProfile Example policy will be created based on the static MAC address of user's PC. We will see the EndpointProfileChangeNotification session notification appear in the running Linux script.

Step 1 Type the following on the Linux host:

```
./endpointprofile_subscribe.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
```

You should see the following:

```
--properties--
version=1.0.0-alpha-144
```

```
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting
press<enter>to disconnection...
connected.
```

Step 2 Select **Administration->pxGrid Service**

Note the pxGrid client has subscribed to the EndpointProfileMetaData Capability

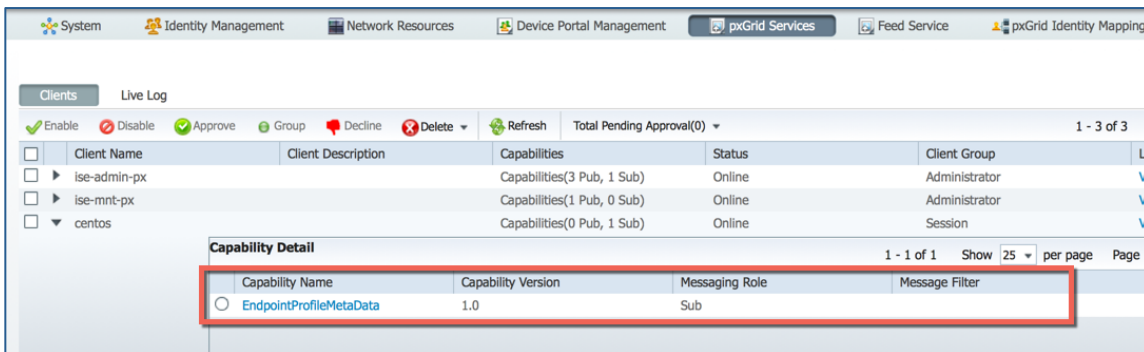


Figure 13. EndpointProfileMetaData Capability

Step 3 Select **Policy->Profiling->Profiling Policies->Add**

Step 4 Enter **Name:** EndpointPolicy_{partnername}

Step 5 Enter **Description:** pxGrid EndpointProfile_Subscribe Example

Step 6 Under **conditions**, create new condition, based on **MACADDRESS:EQUALS:** enter your **MACaddress**

Step 7 Click **Submit.**

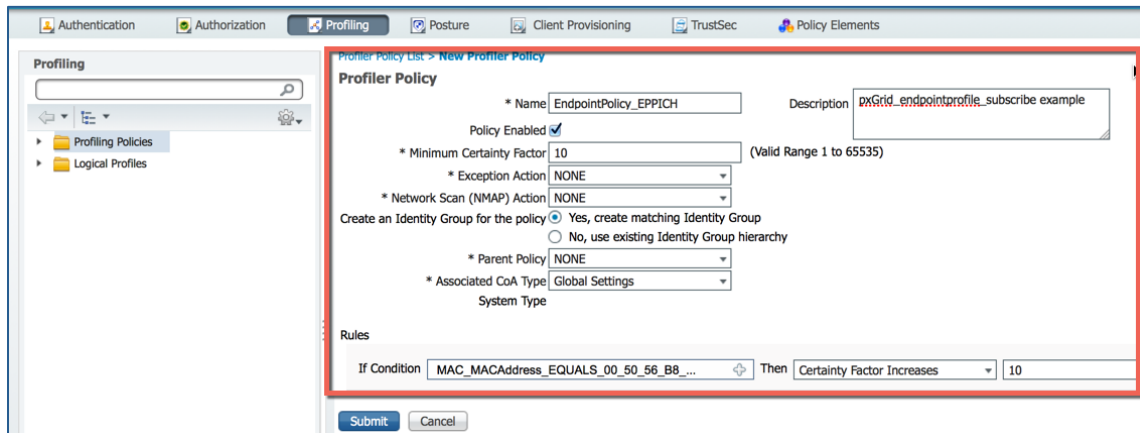


Figure 14. Profile Policy

Step 8 You should see the following appear on the session notification on the Linux host:

```
EndpointProfileChangedNotification (changetype=ADD) Device profile : id=9cd67b00-246b-11e4-ba8d-005056b8b604, name=EndpointPolicy_EPPICH, fqname=EndpointPolicy_EPPICH
```

Security_Group_Subscribe

Verification

This test verifies the ability of the 3rd party system to subscribe to the SecurityGroup topic via pxGrid.

Definition

The security group subscribe script exposes the Security Group Tags (SGT) configured in ISE through the TrustsecMetaDataCapability topic. Security Group Change Notifications will appear in the script session notifications when a security group is added/updated/deleted.

Example

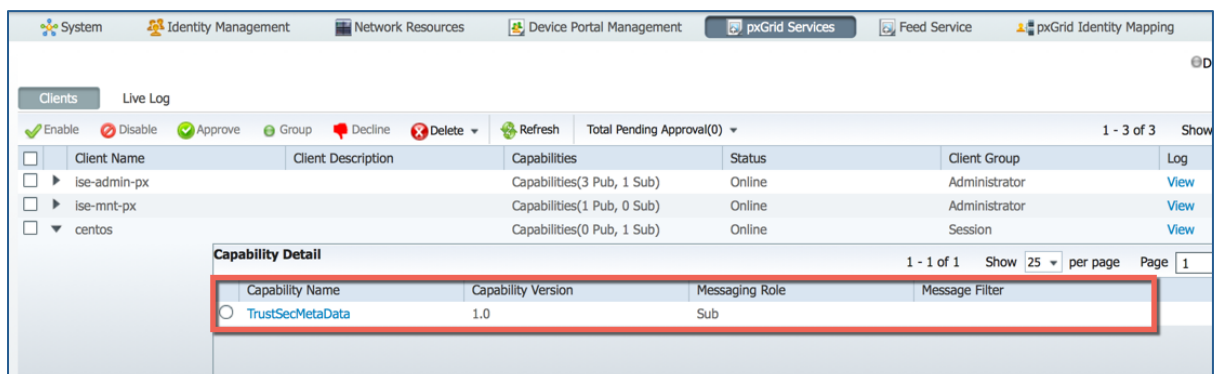
The securitygroup subscribe script subscribe to changes in the ISE Trustsec Policies. In this example, a Security Group Tag of Tech Support will be created. You will see the SecurityGroupChangeNotification appear under session notification in the running script.

Step 1 Type the following on the Linux host:

```
./securitygroup_subscribe.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridHostname px.demo.local -
xgridUsername mac_client
-bash: ./securitgroup_subscribe.sh: No such file or directory
johns-macbook-pro:bin jeppich$
johns-macbook-pro:bin jeppich$ ./securitygroup_subscribe.sh -keystoreFilename iseSample1.jks -
keystorePassword cisco123 -truststoreFilename rootSample.jks -truststorePassword cisco123 -
xgridHostname px.demo.local -xgridUsername centos
----- properties -----
version=1.0.0-alpha-144
xgridHostname=px.demo.local
xgridUsername=mac_client
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
press <enter> to disconnect...
connected.
```

Step 2 Select Administration-pxGrid Services

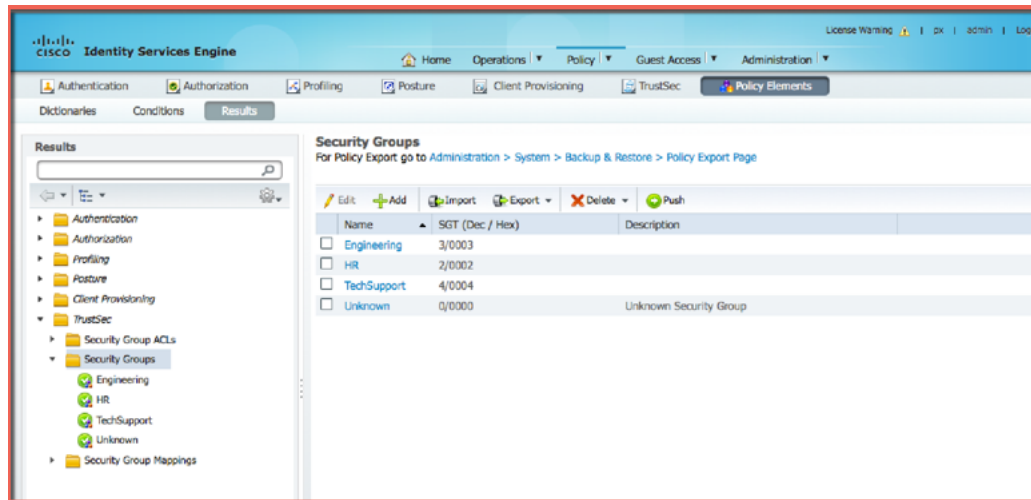
Note that the pxGrid client has successfully registered to the TrustsecMetaData capability



Client Name	Client Description	Capabilities	Status	Client Group	Log
ise-admin-px		Capabilities(3 Pub, 1 Sub)	Online	Administrator	View
ise-mnt-px		Capabilities(1 Pub, 0 Sub)	Online	Administrator	View
centos		Capabilities(0 Pub, 1 Sub)	Online	Session	View

Capability Name	Capability Version	Messaging Role	Message Filter
<input type="radio"/> TrustSecMetaData	1.0	Sub	

Figure 15. TrustsecMetaData

Step 3 Select Policy->Policy Elements->Results->Trustsec->Security Groups and Add Tech Support

Figure 16. Security Group

Step 2 You should see the SecurityGroupChanegNotification appear in the running Linux script

```
SecurityGroupChangeNotification (changetype=ADD) SecurityGroup : id=e547fac0-2ae3-11e4-ba8d-005056b8b604, name=TechSupport, desc=, tag=4
```

Endpoint Protection Service (EPS)

Verification

This test verifies the ability of the 3rd party system to execute a quarantine or network disconnect action on an endpoint on the network.

Definition

The pxGrid client registers to an authorized EPS session group and subscribes to the ISE published EndPointProtection service capability, and quarantines the IP address of the authenticated device, and unquarantines the authenticated device based on the MAC address.

Example

In this example, the client, centos, will register to the authorized EPS group and subscribe to the EndpointProtectionService capability. The eps quarantine script will quarantine jsmith by the IP Address. The eps unquarantine script will unquarantine jsmith by MAC address. The results will be seen on ISE under the Operations Authentication View. Enabling EPS (Endpoint Protection Service) on ISE is a pre-requisite. An authorization profile for quarantine network access will be created and applied to the authorization policy. Authorization profiles provide network access once the user has passed the authentication policies.

Step 1 Type the following on the Linux host

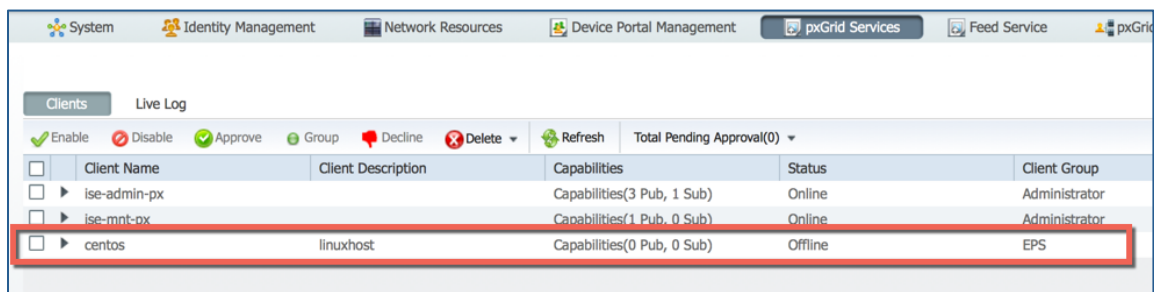
```
./register.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -truststoreFilename rootSample.jks -xgridDescription linuxhost -truststorePassword cisco123 -xgridUsername centos -xgridHostname 10.1.100.60 -group EPS
```

You should see the following:


```
--properties-
version=1.0.0-alpha-144
xgridUsername=centos
xgridDescription=linuxhost
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
-----
registering..
connecting..
connected..
done registering..
connection closed..
```

Step 2 Select Administration->pxGrid Services

Note the pxGrid client has successfully registered to the EPS Group



Client Name	Client Description	Capabilities	Status	Client Group
ise-admin-px		Capabilities(3 Pub, 1 Sub)	Online	Administrator
ise-mnt-px		Capabilities(1 Pub, 0 Sub)	Online	Administrator
centos	linuxhost	Capabilities(0 Pub, 0 Sub)	Offline	EPS

Figure 17. EPS Group

Note: user employee1 may be already by authenticated.

Step 3 Type eps_quarantine on the Linux host and type in the IP address of the host:

```
./eps_quarantine.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
--properties-
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
--connecting-
connected.
ip address (or <enter> to disconnect): 10.1.50.201
ip address (or <enter> to disconnect):
connetion closed
```

Step 4 Select Administration->pxGrid Services.,

Note that the pxGrid client has subscribed to the EndpointProtectionService Capability

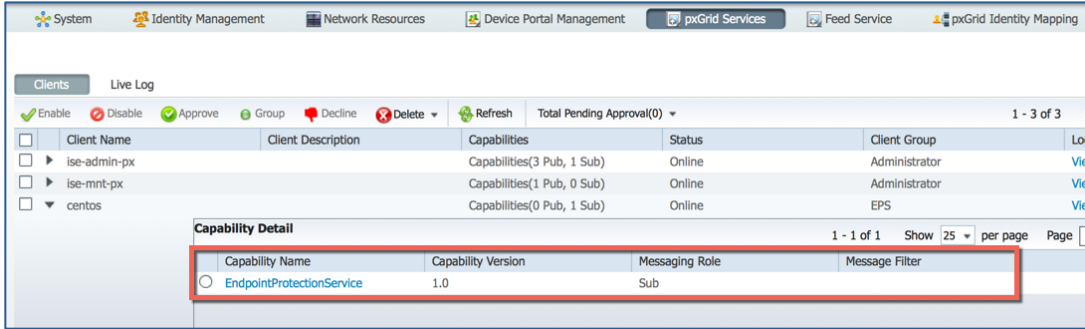


Figure 18.

Step 5 **Operations->Authentications**, the device has been quarantined.

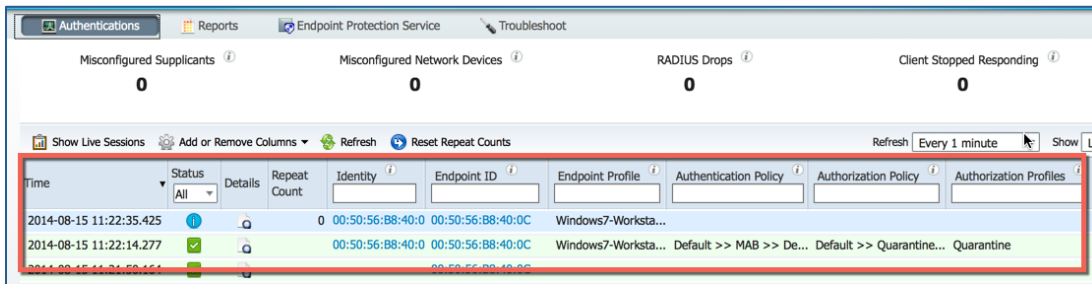


Figure 19. pxGrid Services

Step 6 Type the following and unquarantine the device by entering the MAC address.

```

sssddd./eps_unquarantine.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
-----
connecting...
connected.
mac address (or <enter> to disconnect): 00:50:56:B8:40:0C
mac address (or <enter> to disconnect):
connection closed
    
```

Step 7 **Operations->Authentications**, the device has been unquarantined and has full network access.

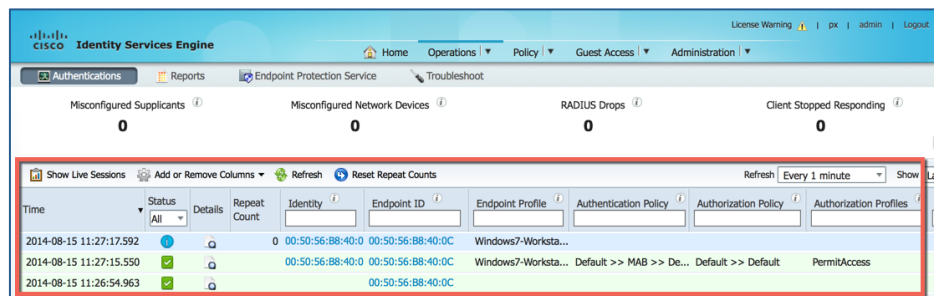


Figure 20. Authentication - unquarantined

Compliance Checks

This section uses the NAC agent and provides a file posture check to verify if the system is compliant. The session_download script will display the Posture State for Compliant and Non-compliant session attributes for employee1 in the downloaded session records.

Compliant

- Step 1** Open the firefox browser on your Linux client and login into ISE
Step 2 Operations->Authentications->Show Live Sessions, you should see the following:

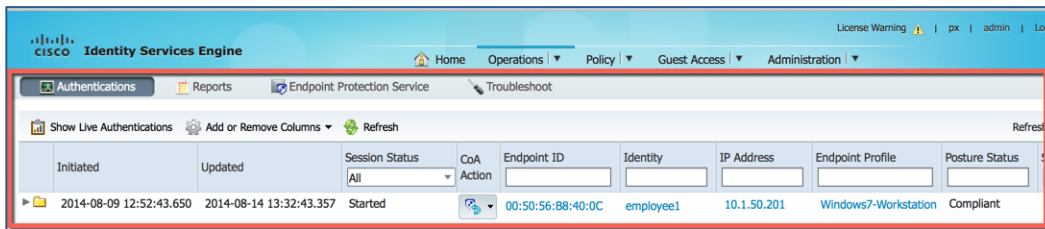


Figure 21. Authentication - Live Session

- Step 3** Click on **CoA**, and Terminate the Session
Step 4 On the Windows p03_w7pc, select PEAP authentication from the Cisco AnyConnect Mobility Client Dropdownbox

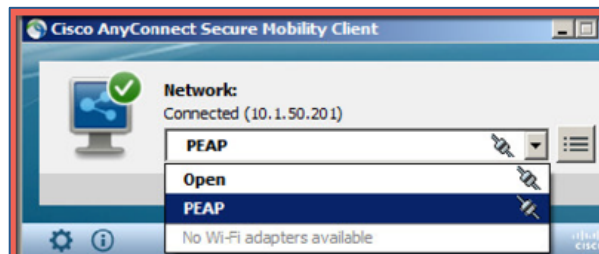


Figure 22. PEAP Authentication

- Step 5** In ISE you should have full access

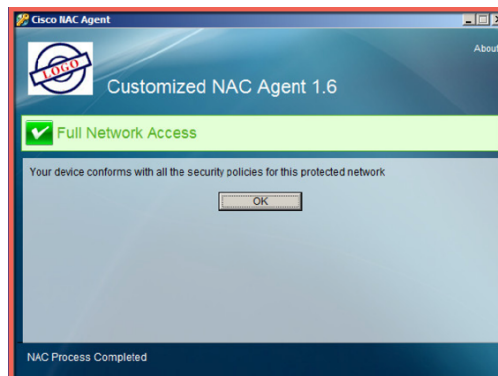
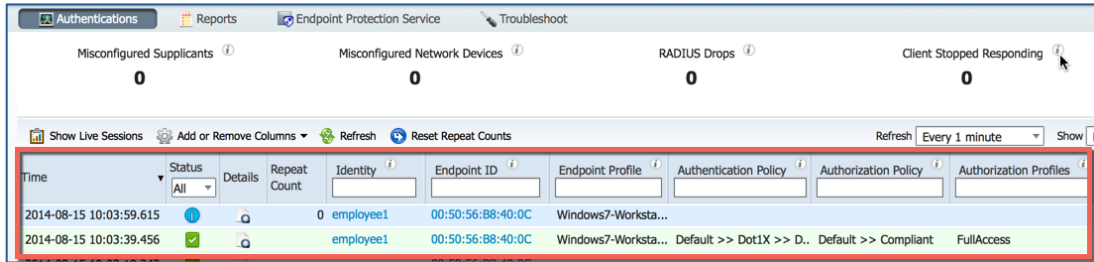


Figure 23. Full Network Access

- Step 6** Select **Operations->Authentications**

Note that employee1 is compliant and has full network access.



Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles
2014-08-15 10:03:59.615			0	employee1	00:50:56:B8:40:0C	Windows7-Worksta...			
2014-08-15 10:03:39.456				employee1	00:50:56:B8:40:0C	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> Compliant	FullAccess

Figure 24. Authentication - compliant user

Step 7 From your Linux host, run the session download script to view compliant session

```
./session_download.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
filter=null
-----
connecting...
connected.
starting at Fri Aug 15 13:42:14 EDT 2014...

session (ip=10.1.50.201, Audit Session Id=0A016401000000202E9E107B, User Name=employee1,
Domain=demo.local, Calling station id=00:50:56:B8:40:0C, Session state= STARTED, Epsstatus=null,
Security Group=null, Endpoint Profile=Windows7-Workstation, NAS IP=10.1.100.1, NAS
Port=GigabitEthernet0/1, RADIUSAVPairs=[ Acct-Session-Id=00000057], Posture Status=Compliant,
Posture Timestamp=Fri Aug 15 06:03:17 EDT 2014, Session Last Update Time=Fri Aug 15 06:03:17 EDT
2014 )... ending at: Fri Aug 15 13:42:14 EDT 2014

-----
downloaded 1 sessions in 31 milliseconds
-----
```

Step 8 Rename c:\pos_check\postest.txt c:\pos_check\old_postest.txt

Step 9 Terminate the session

Step 10 Authenticate select PEAP from the AC dropdown.

The following dialog box appears.

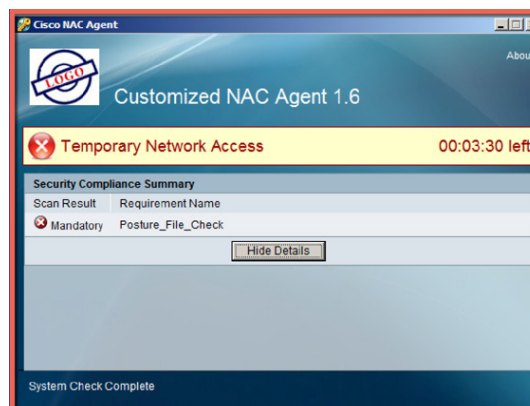
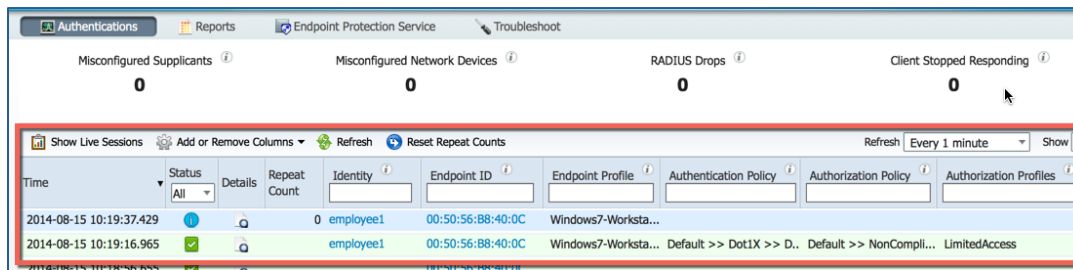


Figure 25. Customized NAE Agent

Step 11 Select Operations->Authentications

Note that employee1 is non-compliant and has limited network access



Time	Status	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles
2014-08-15 10:19:37.429		0	employee1	00:50:56:B8:40:0C	Windows7-Worksta...			
2014-08-15 10:19:16.965			employee1	00:50:56:B8:40:0C	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> NonCompli...	LimitedAccess

Figure 26. Authentication - non-compliant user

Step 12 From your Linux host, run the session download script to view the non-compliant session

```
./session_download.sh -keystoreFilename iseSample1.jks -keystorePassword cisco123 -
truststoreFilename rootSample.jks -truststorePassword cisco123 -xgridUsername centos -
xgridHostname 10.1.100.60
----- properties -----
version=1.0.0-alpha-144
xgridHostname=10.1.100.60
xgridUsername=centos
keystoreFilename=iseSample1.jks
keystorePassword=cisco123
truststoreFilename=rootSample.jks
truststorePassword=cisco123
filter=null
-----
connecting...
connected.
starting at Fri Aug 15 13:50:47 EDT 2014...

session (ip=10.1.50.201, Audit Session Id=0A016401000000212EABE80B, User Name=employee1,
Domain=demo.local, Calling station id=00:50:56:B8:40:0C, Session state= STARTED, Epsstatus=null,
Security Group=null, Endpoint Profile=Windows7-Workstation, NAS IP=10.1.100.1, NAS
Port=GigabitEthernet0/1, RADIUSAVPairs=[ Acct-Session-Id=00000059], Posture Status=Non-compliant,
Posture Timestamp=Fri Aug 15 06:18:54 EDT 2014, Session Last Update Time=Fri Aug 15 06:18:54 EDT
2014 )... ending at: Fri Aug 15 13:50:47 EDT 2014

-----
downloaded 1 sessions in 25 milliseconds
-----

connection closed
[root@centos bin]#
```