

Cisco Identity Services Engine (ISE) 2.0 を使用した Cisco pxGrid との統合の構成 およびテスト

目次

- このドキュメントについて.....7
- pxGrid の操作8
 - 情報のトピック.....8
 - クライアント グループ9
- テスト環境10
 - Cisco Identity Service Engine (ISE 2.0) VM の設定10
 - ISE の初期設定.....11
 - Active Directory ユーザ設定11
 - ネットワーク デバイス14
- pxGrid 向けの ISE の設定.....15
- pxGrid SDK のインストール.....17
 - pxGrid クライアントのテスト用に自己署名証明書を使用する(サンプルの証明書の代替) ...17
 - pxGrid クライアントと ISE pxGrid ノードのテスト22
 - SDK のサンプル証明書を pxGrid のテストに使用する22
 - pxGrid クライアントと ISE pxGrid ノードのテスト.....24
- RADIUS シミュレータ25
 - ISE 内部ユーザの作成25
 - 認証26
 - 認証のテスト26
- pxGrid 2.0 サンプル スクリプト.....28
- RADIUS シミュレータを使用するテスト スクリプト.....30
 - Multigroupclient30
 - 検証.....30
 - 定義.....30
 - 例30
 - セッションのサブスクリブ.....32
 - 検証.....32
 - 定義.....32
 - 例32
 - セッションのダウンロード36
 - 検証.....36

定義.....	36
例.....	36
IP によるセッションのクエリ	37
検証.....	37
定義.....	37
例.....	38
EndpointProfile のサブスクリプション	38
検証.....	38
定義.....	38
例.....	38
ID グループのダウンロード.....	41
検証.....	41
定義.....	41
例.....	41
セキュリティグループのクエリ.....	42
検証.....	42
定義.....	42
例.....	42
セキュリティグループのサブスクリプション	43
検証.....	43
定義.....	43
例.....	43
エンドポイント プロファイルのクエリ.....	46
検証.....	46
定義.....	46
例.....	46
機能.....	47
検証.....	47
定義.....	47
例.....	47
ID グループのクエリ.....	48
検証.....	48
定義.....	48

例	48
ID グループのサブスクリプション	49
検証	49
定義	49
例	49
EPS_Quarantine/EPS_UnQuarantine	51
検証	51
定義	51
例	51
802.1X を使用したサンプル スクリプトのテスト	57
Multigroupclient	57
検証	57
定義	57
例	57
セッションのサブスクリプション	59
検証	59
定義	59
例	59
セッションのダウンロード	61
検証	61
定義	61
例	61
IP によるセッションのクエリ	62
検証	62
定義	62
例	62
EndpointProfile のサブスクリプション	63
検証	63
定義	63
例	63
ID グループのダウンロード	65
検証	65
定義	65

例	65
セキュリティグループのクエリ	66
検証	66
定義	66
例	66
セキュリティグループのサブスクライブ	68
検証	68
定義	68
例	68
エンドポイント プロファイルのクエリ	70
検証	70
定義	70
例	70
機能	71
検証	71
定義	71
例	71
ID グループのクエリ	72
検証	72
定義	72
例	72
ID グループのサブスクライブ	73
検証	73
定義	73
例	73
Adaptive Network Control (ANC) ポリシー	76
ANC 許可ポリシー	76
ANC ポリシー: 検疫	77
エンドポイントの表示/取得/ポリシー適用のための pxGrid ANC 検疫スクリプト	77
ANC の修復	81
ANC のプロビジョニング	84
ANC ポリシーに従うエンドポイントのリスト	86

ダイナミックトピック	88
コアのサブスクリプション	88
Propose_New 機能	89
要約	99
SXP のパブリッシュ	111
TrustSec AAA デバイス	112
TrustSec 向けネットワーク デバイスの構成	112
Cisco Catalyst 3750-x	112
ASA 5505	114
TrustSec 設定の構成	115
セキュリティグループの構成	115
ネットワーク デバイスの許可ポリシーの設定	116
SGACL の定義	116
SACL のマトリックスへの割り当て	116
IP の分散を SGT マッピングから TrustSec 以外のデバイスに許可するように SXP を 構成する	117
静的マッピングの割り当て	117
pxGrid での SXP バインドのパブリッシュ	118
TrustSec ダッシュボード	118
SXP バインドのレポート	120
sxp_download および sxp_subscribe スクリプト	120
トラブルシューティング	122
19:37:39.475 [main] WARN o.a.cxf.phase.PhaseInterceptorChain - Interceptor for {https://ise238.lab6.com:8910/pxgrid/mnt/sd}WebClient has thrown exception, unwinding now	122
参考資料	123
TrustSec デバイス構成	123
TrustSec デバイス構成	123
ASA-5505 向けデバイス構成	123
3750x 向けのデバイス構成	124

このドキュメントについて

このドキュメントには、Cisco Platform Exchange Grid (pxGrid) 向けの ISE 2.0 のインストールの詳細と、関連する SDK、およびサンプルの pxGrid スクリプトが含まれています。これらは、802.1X 以外の環境または 802.1X 環境で実行できます。

pxGrid ISE 2.0 の新機能は、次のとおりです。

- **ダイナミックトピック:** 登録済み/サブスクリプション済みの pxGrid クライアント間でコンテキスト情報を共有できます。pxGrid クライアントは、パブリッシャまたはサブスクリバとして機能してこの情報をパブリッシュしたり使用したりできます。ISE では、この情報を使用できないことに注意してください。
- **Adaptive Network Control (ANC) ポリシー:** サードパーティアプリケーションまたはシスコのセキュリティソリューションによって、ISE ポリシーまたは pxGrid ANC クエリ スクリプトからの検疫、修復、プロビジョニング、ポートバウンス、ポートの閉鎖など、侵害の軽減アクションをカスタマイズします。
- **SXP バインドのパブリッシュ:** これにより、サブスクリバが受信 IP、SGT タグ、送信元、およびピアセッションの情報を取得できるようになります。

802.1X 以外の環境では、Radius シミュレータを使用します。ポスチャ情報、エンドポイント デバイスなどの pxGrid セッション属性をテストするには、802.1X 環境が必要です。

pxGrid ISE 2.0 の機能のテストには 802.1X 環境が必要です。また、SXP をテストする場合は、TrustSec 互換のネットワーク デバイスが必要です。

pxGrid の操作

ISE は、セッション ディレクトリ情報 (pxGrid クライアント、Cisco Security Solution、またはサードパーティ エコシステム パートナーがサブスクリブする ISE のコンテキスト情報を含む) などのトピックをパブリッシュし、イベントについての詳しい情報を提供します。

以下は、成功した 802.1X IEEE 有線認証のエンド ユーザ セッションのサンプルです。ユーザ名、IP アドレス、MAC アドレス、およびデバイス タイプなど、イベントに関連付けられている情報に注目してください。

```
Session={ip=[192.168.1.31], Audit Session Id=0A0000010000002803DBE3C1, User Name=LAB6\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:29:79:02:A8, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Windows7-Workstation, NAS IP=192.168.1.2, NAS Port=GigabitEthernet1/0/12, RADIUSAVPairs=[ Acct-Session-Id=00000053], Posture Status=NonCompliant, Posture Timestamp=Sat Aug 01 15:15:20 EDT 2015, Session Last Update Time=Sat Aug 01 15:15:22 EDT 2015}
```

上記のようなイベント情報を取得したら、組織のセキュリティ ポリシーおよびコンプライアンス要件に基づいてセキュリティ アプリケーションのポリシーを定義できます。たとえば、企業ポリシーに準拠せず、非推奨のデバイスを使って組織のネットワークに接続するエンド ユーザに対して、より制限の厳しいポリシーを適用できます。

また、セキュリティ アプリケーションがデバイスのタイプとユーザのコンテキスト情報を認識する場合は、修復アクションが必要なデバイスのタイプに対して特定のセキュリティ ポリシーを適用できます。

修復アクションは、pxGrid Adaptive Network Control (ANC) 軽減アクションによって実行できます。

情報のトピック

ISE は、以下の情報のトピックをパブリッシュします。

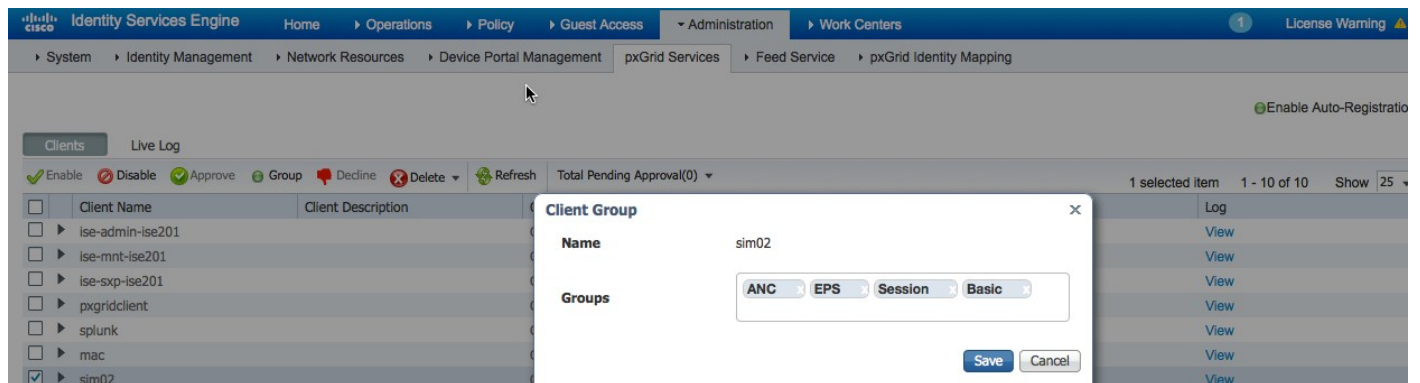
- **GridControllerAdminService**: pxGrid サービスをサブスクリバに提供します。
- **AdaptiveNetworkControl**: 強化された pxGrid ANC 軽減機能をサブスクリバに提供します。
- **Core**: ISE pxGrid ノードのすべての登録済み機能をクエリするための機能を pxGrid クライアントに提供します。
- **EndpointProfileMetada**: ISE からの利用可能なデバイス情報を pxGrid クライアントに提供します。
- **EndpointProtectionService**: ISE 1.3/1.4 からの互換性ある EPS/ANC pxGrid 軽減アクションを提供します。
- **TrustSecMetaData**: 公開されたセキュリティ グループ タグ (SGT) 情報を pxGrid クライアントに提供します。
- **IdentityGroup**: 802.1X 認証では利用できない ID グループ情報を pxGrid クライアントに提供します。
- **SessionDirectory**: ISE によってパブリッシュされたセッション情報または利用可能なセッション オブジェクトを pxGrid クライアントに提供します。

クライアントグループ

pxGrid クライアントは、ISE pxGrid ノードに対して認証、接続、およびクライアントグループへの登録を実行し、これらのトピックへのサブスクライブまたは直接のクエリを発行します。また、pxGrid クライアントは複数のクライアントグループをサブスクライブします。

次の pxGrid クライアントグループがあります。

- **Basic:** ISE pxGrid ノードの接続を提供します。pxGrid の管理者は、登録済みの pxGrid クライアントを手動で他のクライアントグループに移動する必要があります。通常、pxGrid セッション オブジェクトへのアクセスを提供する Session グループに移動します。
- **Administrator:** ISE によってパブリッシュされるノードのクライアント用に予約されています。
- **Session:** pxGrid セッション オブジェクトへのアクセスを提供します。
- **ANC:** ANC ポリシー アクションにアクセスします。
- **EPS:** ISE 1.3/ISE 1.4 の eps_quarantine/eps_unquarantine pxGrid スクリプトとの互換性を備えています。



The screenshot displays the Cisco Identity Services Engine (ISE) Administration interface. The breadcrumb navigation shows: Home > Operations > Policy > Guest Access > Administration > Work Centers. The main navigation includes System, Identity Management, Network Resources, Device Portal Management, pxGrid Services, Feed Service, and pxGrid Identity Mapping. The 'Clients' tab is active, showing a table of client groups. The 'sim02' group is selected. A 'Client Group' configuration modal is open, showing the name 'sim02' and a list of groups: ANC, EPS, Session, and Basic. The 'Session' group is currently selected in the modal. The background table shows columns for Client Name, Client Description, Log, and View.

テスト環境

pxGrid のテスト環境には、次のものがが必要です。

- VMware 5.5 ESX サーバ
- 以下の 3 つ以上の VM が必要です。
 - ISE 2.0 pxGrid ノード
 - Microsoft Active Directory (DNS と NTP も含む) 用の Windows 2008 R2 CA サーバ

(注) CA によって署名された証明書をテストするため、これを CA サーバとして構成することも必要です。

- 802.1x サプリカントを使用する Windows PC クライアント、Cisco AnyConnect NAM、または RADIUS シミュレータ

(注) 802.1X 環境を利用できない場合は、RADIUS シミュレータを使用します。

- 802.1X 環境: Cisco Catalyst 3750-x、Cisco Catalyst 3560-x、Cisco Catalyst 3850。新しい ISE SXP 機能をテストする場合、<http://www.cisco.com/c/en/us/solutions/enterprise-networks/trustsec/trustsec-matrix-archived.html> の TrustSec の互換性マトリックスを参照してください。それ以外の場合は、ネットワーク アクセスデバイスに ISE との互換性が必要です。http://www.cisco.com/c/en/us/td/docs/security/ise/1-4/compatibility/ise_sdt.html#pgfid-198199 を参照してください。
- pxGrid クライアント: Mac または Linux クライアント、Cisco Security Solution、サードパーティ pxGrid パートナー アプリケーション
- ISE 2.0.0.306
- pxGrid SDK 1.0.2.32

Cisco Identity Service Engine (ISE 2.0) VM の設定

ここでは、ESX サーバ VM 作成の初期構成について説明します。

- Linux 5 64 ビット オペレーティング システム
- 100 GB 以上の OS ハードドライブ
- 8 GB RAM
- 2 つの NIC (1 つの NIC を SXP リスナーとして使用する場合)

(注) PC クライアント用に同じ VM のネットワーク NIC を使用しないでください。802.1X 環境では PC クライアントのポートが 802.1X 構成用に設定されるためです。

ISE を構成する前に Active Directory ドメインが動作していることを確認します。ISE 設定の構成では、ホスト名、IP アドレス、ドメイン名、DNS および NTP サーバ名が必要です。

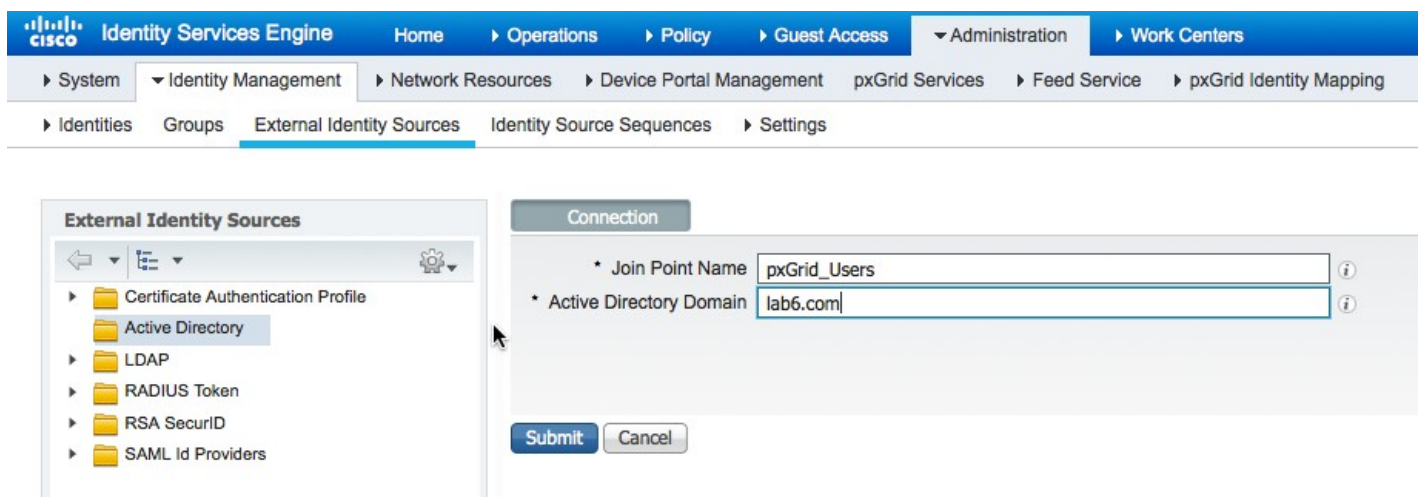
ISE、pxGrid クライアント、および PC クライアントが FQDN で解決できる必要があります。

ISE の初期設定

このセクションでは、エンド ユーザ認証用の Active Directory の設定について説明します。

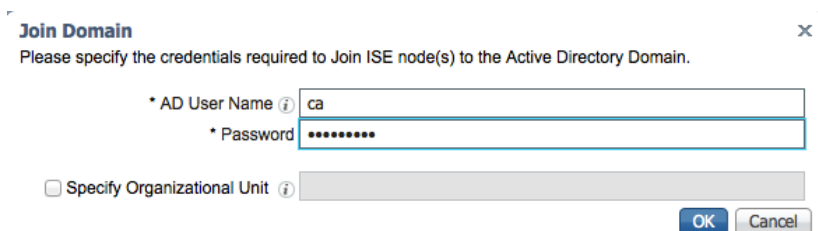
Active Directory ユーザ設定

- ステップ 1** Active Directory 接続を構成します。
 [管理 (Administration)] > [ID 管理 (Identity Management)] > [外部 ID ソース (External Identity Sources)] > [Active Directory] > [追加 (Add)] の順に選択します。
 次の情報を入力します。参加ポイント名: **pxGrid_users**
 Active Directory ドメイン名: **lab6.com**



The screenshot shows the Cisco Identity Services Engine (ISE) Administration console. The breadcrumb navigation is: Administration > External Identity Sources > Add. The 'External Identity Sources' pane on the left shows a tree view with 'Active Directory' selected. The main configuration area is titled 'Connection' and contains two required fields: 'Join Point Name' with the value 'pxGrid_Users' and 'Active Directory Domain' with the value 'lab6.com'. There are 'Submit' and 'Cancel' buttons at the bottom.

- ステップ 2** [送信 (Submit)] を選択し、さらにすべての ISE ノードを Active Directory に追加します。
ステップ 3 ドメインのクレデンシャルを入力します。



The 'Join Domain' dialog box prompts the user to specify credentials for joining ISE nodes to the Active Directory Domain. It contains the following fields: 'AD User Name' with the value 'ca', 'Password' with masked characters, and an unchecked checkbox for 'Specify Organizational Unit'. 'OK' and 'Cancel' buttons are at the bottom right.

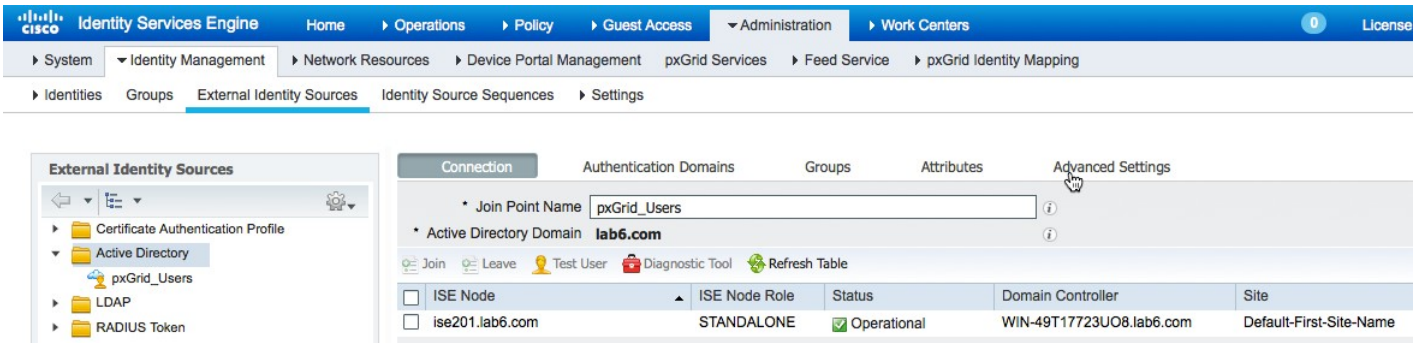
- ステップ 4** [OK] をクリックすると、参加ステータスとして [完了 (Completed)] が表示されます。

Join Operation Status
 Status Summary: Successful

ISE Node	Node Status
ise201.lab6.com	Completed.

(注) ステータスとして [失敗 (Failure)] が表示される場合は、ISE と MS Active Directory の時間が同期しており、FQDN が解決可能であることを確認してください。

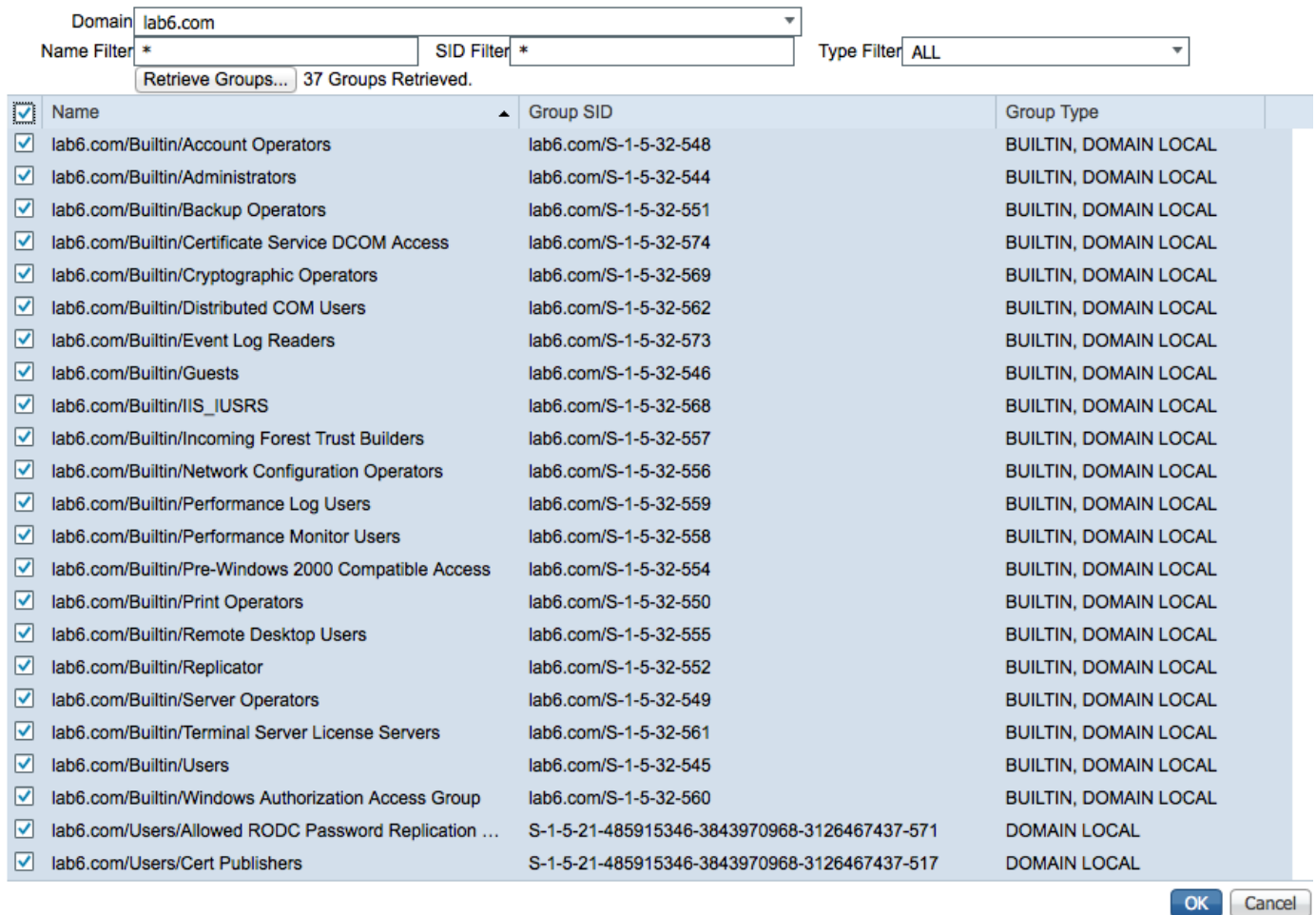
ステップ 5 [閉じる (Close)] を選択すると、次のように表示されます。



ステップ 6 [グループ (Groups)] > [追加 (Add)] > [Active Directory からグループを選択 (Select Groups from Active Directory)] > [グループの取得 (Retrieve groups)] > [すべてを選択 (select all)] > [OK] の順にクリックします。

Select Directory Groups

This dialog is used to select groups from the Directory.



ステップ7 [OK] をクリックします。

The screenshot shows the Cisco Identity Services Engine Administration console. The breadcrumb navigation is: Home > Operations > Policy > Guest Access > Administration > Work Centers. The left sidebar shows the 'External Identity Sources' tree with 'Active Directory' > 'pxGrid_Users' selected. The main content area shows the 'Groups' tab with a table of groups and their SIDs.

Name	SID
<input type="checkbox"/> lab6.com/Builtin/Account Operators	lab6.com/S-1-5-32-548
<input type="checkbox"/> lab6.com/Builtin/Administrators	lab6.com/S-1-5-32-544
<input type="checkbox"/> lab6.com/Builtin/Backup Operators	lab6.com/S-1-5-32-551
<input type="checkbox"/> lab6.com/Builtin/Certificate Service DCOM Access	lab6.com/S-1-5-32-574
<input type="checkbox"/> lab6.com/Builtin/Cryptographic Operators	lab6.com/S-1-5-32-569
<input type="checkbox"/> lab6.com/Builtin/Distributed COM Users	lab6.com/S-1-5-32-562
<input type="checkbox"/> lab6.com/Builtin/Event Log Readers	lab6.com/S-1-5-32-573
<input type="checkbox"/> lab6.com/Builtin/Guests	lab6.com/S-1-5-32-546
<input type="checkbox"/> lab6.com/Builtin/IISS_IUSRS	lab6.com/S-1-5-32-568
<input type="checkbox"/> lab6.com/Builtin/Incoming Forest Trust Builders	lab6.com/S-1-5-32-557
<input type="checkbox"/> lab6.com/Builtin/Network Configuration Operators	lab6.com/S-1-5-32-556
<input type="checkbox"/> lab6.com/Builtin/Performance Log Users	lab6.com/S-1-5-32-559
<input type="checkbox"/> lab6.com/Builtin/Performance Monitor Users	lab6.com/S-1-5-32-558
<input type="checkbox"/> lab6.com/Builtin/Pre-Windows 2000 Compatible Access	lab6.com/S-1-5-32-554
<input type="checkbox"/> lab6.com/Builtin/Print Operators	lab6.com/S-1-5-32-550
<input type="checkbox"/> lab6.com/Builtin/Remote Desktop Users	lab6.com/S-1-5-32-555
<input type="checkbox"/> lab6.com/Builtin/Replicator	lab6.com/S-1-5-32-552
<input type="checkbox"/> lab6.com/Builtin/Server Operators	lab6.com/S-1-5-32-549
<input type="checkbox"/> lab6.com/Builtin/Terminal Server License Servers	lab6.com/S-1-5-32-561
<input type="checkbox"/> lab6.com/Builtin/Users	lab6.com/S-1-5-32-545

ステップ8 [保存(Save)] をクリックします。

ステップ9 [pxGrid_Users] をクリックすると、次のように表示されます。

The screenshot shows the configuration details for the 'pxGrid_Users' group. The breadcrumb navigation is: Home > Operations > Policy > Guest Access > Administration > Work Centers. The left sidebar shows 'Active Directory' > 'pxGrid_Users' selected. The main content area shows the 'Groups' tab with configuration fields and a table of ISE nodes.

Configuration fields:

- Join Point Name: pxGrid_Users
- Active Directory Domain: lab6.com

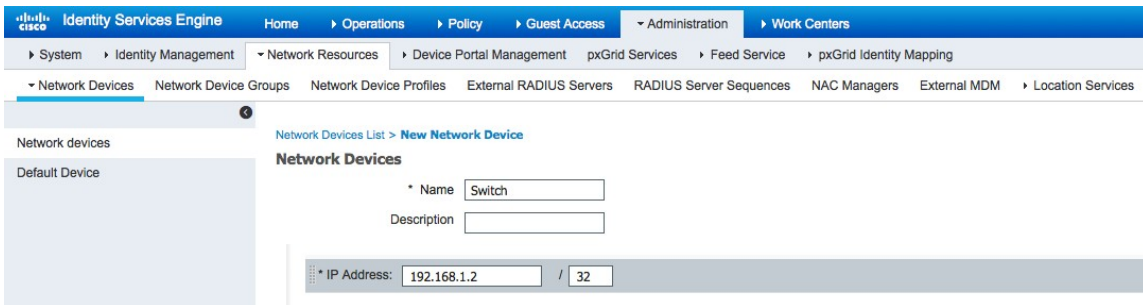
Actions: Join, Leave, Test User, Diagnostic Tool, Refresh Table

ISE Node	ISE Node Role	Status	Domain Controller	Site
<input type="checkbox"/> ise201.lab6.com	STANDALONE	<input checked="" type="checkbox"/> Operational	WIN-49T17723UO8.lab6.com	Default-First-Site-Name

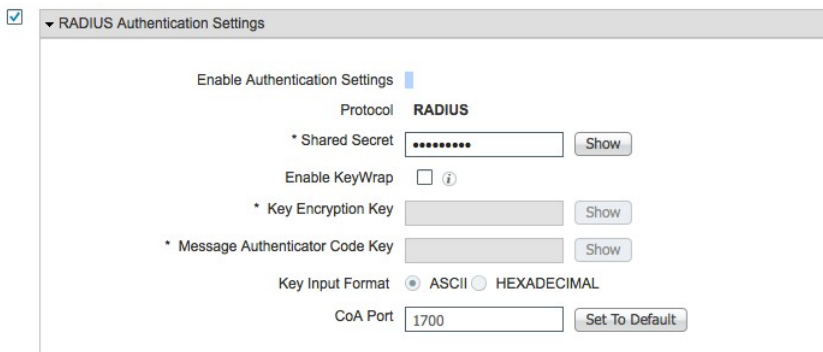
ネットワーク デバイス

ネットワーク デバイス、シスコ スイッチ、および WLAN コントローラを追加します。RADIUS シミュレータを実行している場合は、RADIUS シミュレータを実行する PC クライアントの IP アドレスを入力します。RADIUS シミュレータを使用している場合は、共有秘密として「secret」を使用します。

- ステップ 1** [管理 (Administration)] > [ネットワーク リソース (Network Resources)] > [ネットワーク デバイス (Network Devices)] > [ネットワーク デバイスの追加 (Add Network Device)] の順に選択します。
名前として「Switch」を入力します。
[IP アドレス (IP Address)] に「192.168.1.2」を入力します。

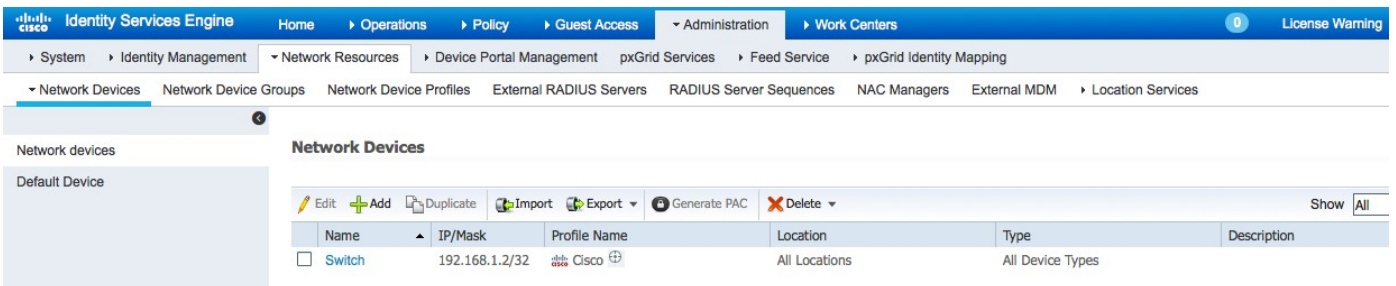


- ステップ 2** [Radius 認証設定 (Radius Authentication Settings)] を有効にして、共有秘密を入力します。



- ステップ 3** [送信 (Submit)] をクリックします。

- ステップ 4** 次のように表示されます。

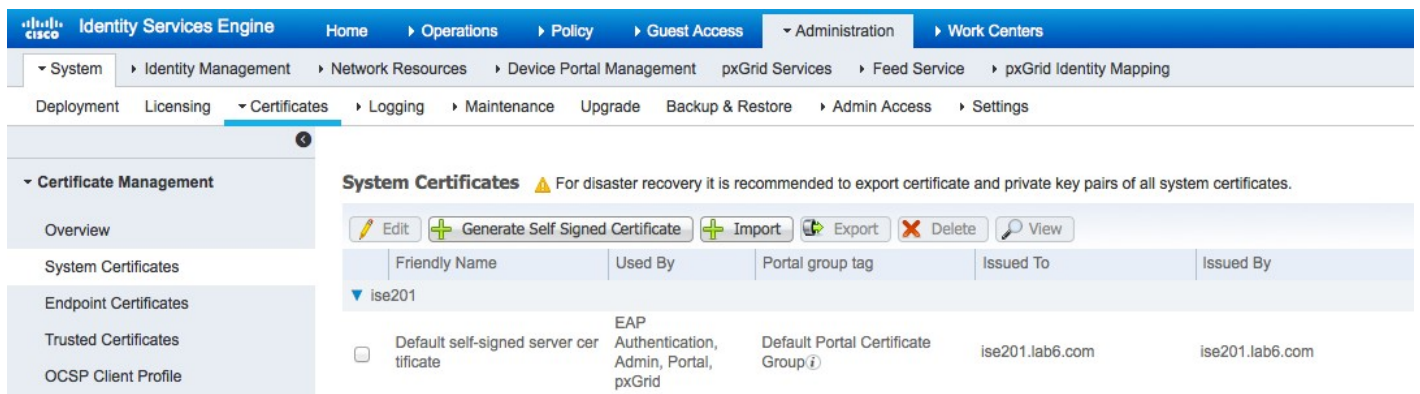


pxGrid 向けの ISE の設定

pxGrid サービスを有効にするために、自己署名の ISE ID 証明書を使用します。

(注) ISE 1.3 および ISE 1.4 では、自己署名の ISE ID 証明書をエクスポートし、Trusted System 証明書ストアにインポートしてから pxGrid サービスを開始する必要がありましたが、この操作は不要になりました。

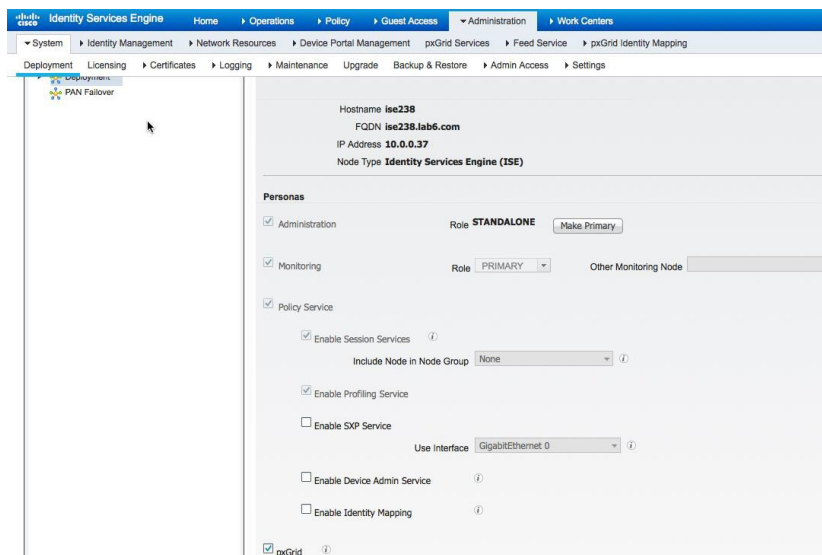
ステップ 1 [管理 (Administration)] > [証明書 (Certificates)] の順に選択し、デフォルトの自己署名証明書を確認します。



System Certificates ⚠ For disaster recovery it is recommended to export certificate and private key pairs of all system certificates.

Friendly Name	Used By	Portal group tag	Issued To	Issued By
Default self-signed server certificate	EAP Authentication, Admin, Portal, pxGrid	Default Portal Certificate Group	ise201.lab6.com	ise201.lab6.com

ステップ 2 pxGrid ペルソナを有効にします。
[管理 (Administration)] > [システム展開 (System Deployment)] > [pxGrid ノードの有効化 (Enable pxGrid node)] の順に選択します。



Hostname **ise238**
 FQDN **ise238.lab6.com**
 IP Address **10.0.0.37**
 Node Type **Identity Services Engine (ISE)**

Personas

- Administration Role **STANDALONE**
- Monitoring Role **PRIMARY** Other Monitoring Node
- Policy Service
 - Enable Session Services Include Node in Node Group **None**
 - Enable Profiling Service
 - Enable SXP Service Use Interface **GigabitEthernet 0**
 - Enable Device Admin Service
 - Enable Identity Mapping
- pxGrid

ステップ 3 MNT ノードから ISE によってパブリッシュされた情報のトピックが表示されます。

(注) これが表示されるまでに数分かかることがあります。

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers License Warning

System Identity Management Network Resources Device Portal Management pxGrid Services Feed Service pxGrid Identity Mapping

Disable Auto-Reg

Clients Live Log

Enable Disable Approve Group Decline Delete Refresh Total Pending Approval(0) 1 - 2 of 2 Show

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View

Capability Detail 1 - 3 of 3 Show 25 per page Page 1

Capability Name	Capability Version	Messaging Role	Message Filter
<input type="radio"/> Core	1.0	Sub	
<input type="radio"/> IdentityGroup	1.0	Pub	
<input type="radio"/> SessionDirectory	1.0	Pub	

ステップ 4 Admin ノードから ISE によってパブリッシュされた情報のトピックが表示されます。

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers License Warning

System Identity Management Network Resources Device Portal Management pxGrid Services Feed Service pxGrid Identity Mapping

Disable Auto-Reg

Clients Live Log

Enable Disable Approve Group Decline Delete Refresh Total Pending Approval(0) 1 - 2 of 2 Show

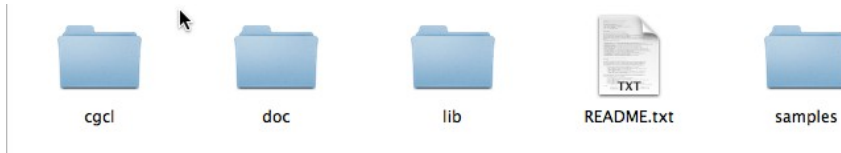
Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View

Capability Detail 1 - 6 of 6 Show 25 per page Page 1

Capability Name	Capability Version	Messaging Role	Message Filter
<input type="radio"/> GridControllerAdminService	1.0	Sub	
<input type="radio"/> AdaptiveNetworkControl	1.0	Pub	
<input type="radio"/> Core	1.0	Sub	
<input type="radio"/> EndpointProfileMetaData	1.0	Pub	
<input type="radio"/> EndpointProtectionService	1.0	Pub	
<input type="radio"/> TrustSecMetaData	1.0	Pub	

pxGrid SDK のインストール

SDK ファイルをダウンロードして解凍すると、次のフォルダが抽出されます。



../samples/cert フォルダには、pxGrid スクリプトを実行するためのサンプル証明書が含まれています。

../samples/bin フォルダには、サンプルの pxGrid 「Java」スクリプトが含まれています。cgcl フォルダには、pxGrid 「C」ライブラリが含まれています。

```

ANCAction_query.sh
alpha.jks
alpha_root.jks
capability_query.sh
common.sh
core_subscribe.sh
endpointprofile_query.sh
endpointprofile_subscribe.sh
eps_quarantine.sh
eps_unquarantine.sh
generic_action_client.properties
generic_client.sh
generic_publisher.properties
generic_subscriber.properties
identity_group_download.sh
identity_group_query.sh
identity_group_subscribe.sh
multigroupclient.sh
propose_capability.sh
securitygroup_query.sh
securitygroup_subscribe.sh
session_download.sh
session_query_by_ip.sh
session_sub_download.sh
session_subscribe.sh
sxp_download.sh
sxp_subscribe.sh

```

これらのスクリプトを実行するには、Oracle 社の Java 開発キットが必要です。

pxGrid クライアントのテスト用に自己署名証明書を使用する(サンプルの証明書の代替)

ISE pxGrid を使用して pxGrid クライアントをテストする場合、自己署名証明書を使用していました。以下は、pxGrid スクリプトのテストで自己署名証明書を使用した場合の手順です。

ステップ 1 pxGrid クライアントの秘密キー(アルファ キー)を生成します。

```

openssl genrsa -out alpha.key 4096

Generating RSA private key, 4096 bit long modulus
.....++
.....++
e is 65537 (0x10001)

```

ステップ 2 自己署名 CSR (alpha.csr) 要求を生成し、チャレンジ パスワードを入力します。

```
openssl req -new -key alpha.key -out alpha.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 []:LAB
```

(注) 便宜上、この文書全体で同じパスワードを使用してください。

ステップ 3 自己署名証明書の公開キー ペアの証明書(alpha.cer)を生成します。

```
openssl req -x509 -days 365 -key alpha.key -in alpha.csr -out alpha.cer
```

ステップ 4 PKCS12 ファイル(alpha.p12)が秘密キーから作成されます。

```
openssl pkcs12 -export -out alpha.p12 -inkey alpha.key -in alpha.cer
```

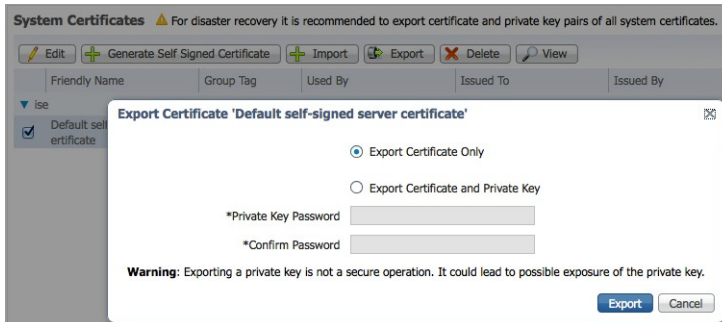
```
Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123
```

ステップ 5 alpha.p12 が ID キーストアにインポートされます(alpha.jks)。キーストアのファイル名は、.jks の拡張子を持つ任意のファイル名にすることができます。これは、pxGrid スクリプト内で keystoreFilename およびその keystorePassword として機能します。

```
keytool -importkeystore -srckeystore alpha.p12 -destkeystore alpha.jks -srcstoretype PKCS12
```

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

- ステップ 6** 公開 ISE ID 証明書のみを pxGrid クライアントにエクスポートします。これは、.pem 形式になります。.pem の拡張子の付いたファイルの名前を識別しやすいものに変更することもできます。この例のファイルは isemnt.pem に変更されています。



- ステップ 7** .pem ファイルを .der 形式に変更します。

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

- ステップ 8** ISE ID 証明書を ID キーストアに追加します。これは、pxGrid セッションダウンロードスクリプトを実行しているときに、ISE MNT ノードからの一括セッションダウンロードを保護するために使用されます。

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

```
Enter keystore password: cisco123
Owner: CN=ise.lab6.com
Issuer: CN=ise.lab6.com
Serial number: 548502f500000000ec27e53c1dd64f46
Valid from: Sun Dec 07 17:46:29 PST 2014 until: Mon Dec 07 17:46:29 PST 2015
Certificate fingerprints:
    MD5: 04:7D:67:04:EC:D2:F5:BC:DC:79:4D:0A:FF:62:09:FD
    SHA1: 5A:7B:02:E4:07:A1:D2:0B:7D:A5:AE:83:27:3B:E7:33:33:30:1E:32
    SHA256:
C4:21:6C:6F:5B:06:F3:2C:D7:26:35:CB:BE:2B:1B:FF:0E:EE:09:91:F6:B6:54:0C:6F:63:CB:43:1F:77:F2:37
Signature algorithm name: SHA1withRSA
Version: 3

Extensions:
```

```
#1: ObjectID: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#2: ObjectID: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  serverAuth
  clientAuth
]

#3: ObjectID: 2.5.29.15 Criticality=false
KeyUsage [
  DigitalSignature
  Key_Encipherment
  Key_Agreement
  Key_CertSign
]

#4: ObjectID: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
  SSL server
]

#5: ObjectID: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: C4 F3 1A 9E 7B 1B 14 4F 51 9E A4 88 33 07 7A AC .....OQ...3.z.
0010: 75 37 36 D4                               u76.
]
]

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

ステップ 9 pxGrid クライアント証明書を ID キーストアにインポートします。

```
keytool -import -alias pxGridclient1 -keystore alpha.jks -file alpha.cer

Enter keystore password:
Certificate already exists in keystore under alias <1>
Do you still want to add it? [no]: n
Certificate was not 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.

ステップ 10 ISE ID 証明書をトラスト キーストア(alpha_root.jks)にインポートします。これは、pxGrid スクリプト用のトラストストアのファイル名およびトラストストアのパスワードとして機能します。

```
keytool -import -alias root1 -keystore alpha_root.jks -file isemnt.der
Enter keystore password:
Re-enter new password:
Owner: CN=ise.lab6.com
Issuer: CN=ise.lab6.com
Serial number: 548502f500000000ec27e53c1dd64f46
Valid from: Sun Dec 07 17:46:29 PST 2014 until: Mon Dec 07 17:46:29 PST 2015
```

```
Certificate fingerprints:
  MD5: 04:7D:67:04:EC:D2:F5:BC:DC:79:4D:0A:FF:62:09:FD
  SHA1: 5A:7B:02:E4:07:A1:D2:0B:7D:A5:AE:83:27:3B:E7:33:33:30:1E:32
  SHA256:
C4:21:6C:6F:5B:06:F3:2C:D7:26:35:CB:BE:2B:1B:FF:0E:EE:09:91:F6:B6:54:0C:6F:63:CB:43:1F:77:F2:37
  Signature algorithm name: SHA1withRSA
  Version: 3

Extensions:

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

#2: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
  serverAuth
  clientAuth
]

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

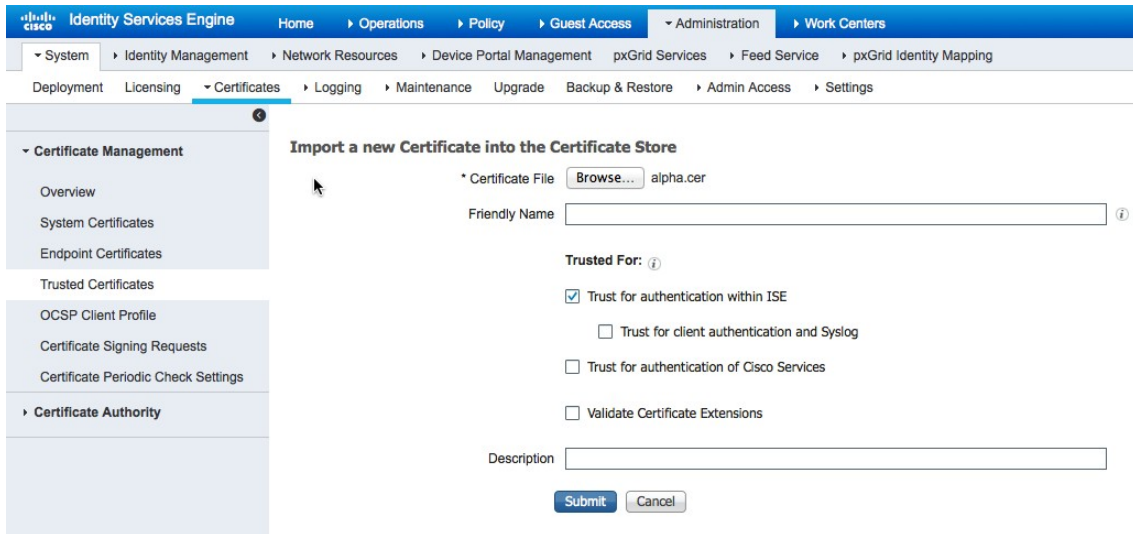
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
  SSL server
]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
0000: C4 F3 1A 9E 7B 1B 14 4F 51 9E A4 88 33 07 7A AC .....OQ...3.z.
0010: 75 37 36 D4                                     u76.
  ]
]

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

ステップ 11 pxGrid クライアントの公開証明書(alpha.cer)を ISE の信頼できる証明書ストアにアップロードします。

ステップ 12 [管理 (Administration)] > [証明書管理 (Certificate Management)] > [信頼できる証明書 (Trusted Certificates)] の順に選択し、alpha.cer を ISE pxGrid ノードにアップロードします。



The screenshot shows the Cisco Identity Services Engine (ISE) web interface. The navigation menu includes Home, Operations, Policy, Guest Access, Administration, and Work Centers. Under Administration, the path is System > Identity Management > Network Resources > Device Portal Management > pxGrid Services > Feed Service > pxGrid Identity Mapping. The left sidebar shows 'Certificate Management' with sub-items: Overview, System Certificates, Endpoint Certificates, Trusted Certificates, OSCP Client Profile, Certificate Signing Requests, Certificate Periodic Check Settings, and Certificate Authority. The main content area is titled 'Import a new Certificate into the Certificate Store'. It contains the following fields and options:

- * Certificate File: Browse... alpha.cer
- Friendly Name: [Text Input Field]
- Trusted For:
 - Trust for authentication within ISE
 - Trust for client authentication and Syslog
 - Trust for authentication of Cisco Services
 - Validate Certificate Extensions
- Description: [Text Input Field]
- Buttons: Submit, Cancel

ステップ 13 ID キーストア (alpha.jks) およびトラスト キーストア (alpha_root.jks) を ../samples/bin/.. フォルダにコピーします。

pxGrid クライアントと ISE pxGrid ノードのテスト

multigroupclient pxGrid スクリプト ファイルを実行して、pxGrid クライアントを ISE pxGrid ノードに登録します。

ステップ 1 pxGrid クライアントを ISE pxGrid ノードに登録します。

```
./multigroupclient.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

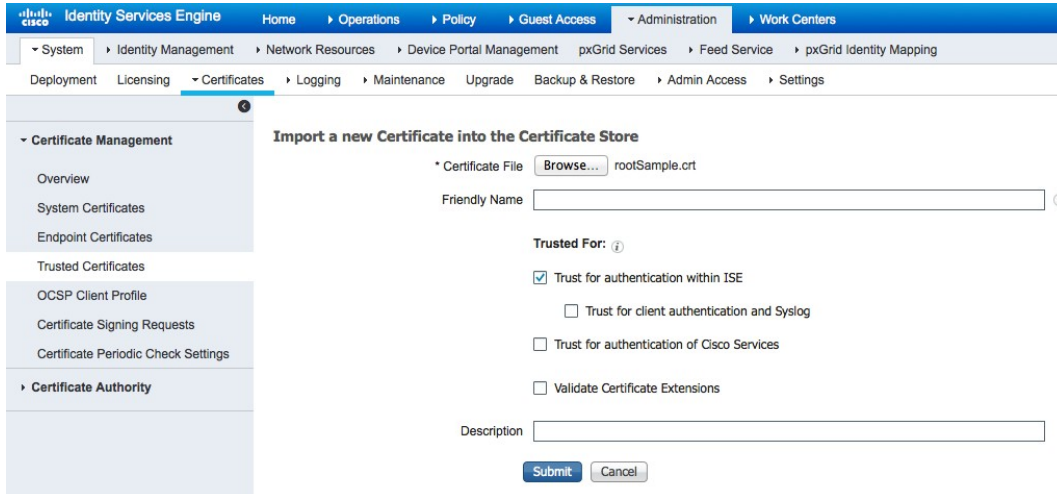
SDK のサンプル証明書を pxGrid のテストに使用する

rootSample.crt を、ISE pxGrid ノードにアップロードします。これは、信頼できる証明書として機能します。また、iseSample1.crt および iseSample1.key ファイルをアップロードします。これは、pxGrid クライアントの ID 証明書として機能します。秘密キーのパスワードは cisco123 です。

ID ストア iseSample1.jks ファイルとトラスト ストア rootSample.jks ファイルは、pxGrid スクリプトから呼び出されます。

(注) これはテスト目的専用であり、実稼働の ISE 環境では使用しません

ステップ 1 rootSample.cert ファイルを、ISE システムトラストストアにアップロードします。
[管理 (Administration)] > [システム (System)] > [証明書管理 (Certificate Management)] > [信頼できる証明書 (Trusted Certificates)] の順に選択し、rootSample.cert ファイルをインポートします。
[ISE 内認証の信頼 (Trust for authentication within ISE)] を有効にします。



ステップ 2 [送信 (Submit)] を選択します。

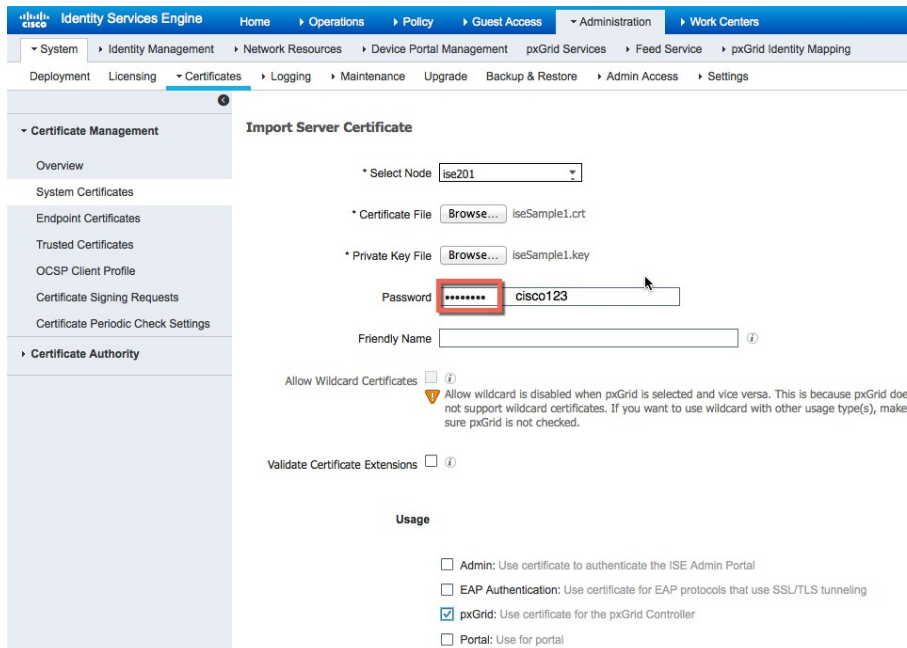
ステップ 3 iseSample1.crt を ISE システムの証明書ストアにアップロードします。

ステップ 4 [管理 (Administration)] > [システム (System)] > [証明書管理 (Certificate Management)] > [システム証明書 (System Certificates)] の順に選択し、iseSample1.crt ファイルをインポートします。

ステップ 5 [管理 (Administration)] > [システム (System)] > [証明書管理 (Certificate Management)] > [システム証明書 (System Certificates)] の順に選択し、iseSample1.key ファイルをインポートします。

ステップ 6 パスワードとして **cisco123** と入力します。

ステップ 7 pxGrid の証明書の使用を有効にします。



ステップ 8 [送信 (Submit)] を選択します。

pxGrid クライアントと ISE pxGrid ノードのテスト

pxGrid multigroupclient スクリプト を実行して、ISE pxGrid ノードを使用して pxGrid クライアントを登録します。

ステップ 1 pxGrid クライアントを ISE pxGrid ノードに登録します。

```
./multigroupclient.sh -a 192.168.1.23 -u SIM01 -k iseSample1.jks -p cisco123 -t rootSample.jks -q cisco123
```


RADIUS シミュレータ

RADIUS シミュレータは IEEE 802.1X 環境がない組織で実行します。

RADIUS シミュレータは 802.1X 認証を提供し、セッション ディレクトリへの IP、MAC、および ID グループ情報などの基本的な属性の入力を可能にします。エンドポイントプロファイル、ポスチャステータスなどのセッション属性は、802.1X を使用する場合にのみ取得できます。

(注) ネイティブのサブリカントまたは AnyConnect NAM は、RADIUS シミュレータを使用する場合は PC に配備しないでください。さらに、RADIUS シミュレータには、RADIUS シミュレータの PARAMETERS リストで定義されているコマンドライン引数があります。

コマンドライン引数 -DUSERNAME、-DPASSWORD、-DCALLING_STATION_ID、- DAUDIT_SESSION_ID、- DACCT_SESSION_ID、-DFRAMED_IP_ADDRESS、-DFRAMED_IP_MASK、RadiusAccountingStart、RadiusAccountingStop、RadiusAuthentication は、複数のエンド ユーザ認証のテストに使用されます。

(注) RADIUS シミュレータのコマンドでは、大文字と小文字が区別されます

RADIUS シミュレータには、Java 開発キットが必要です。RADIUS シミュレータは、pxGrid クライアントまたはクライアント PC で実行できます

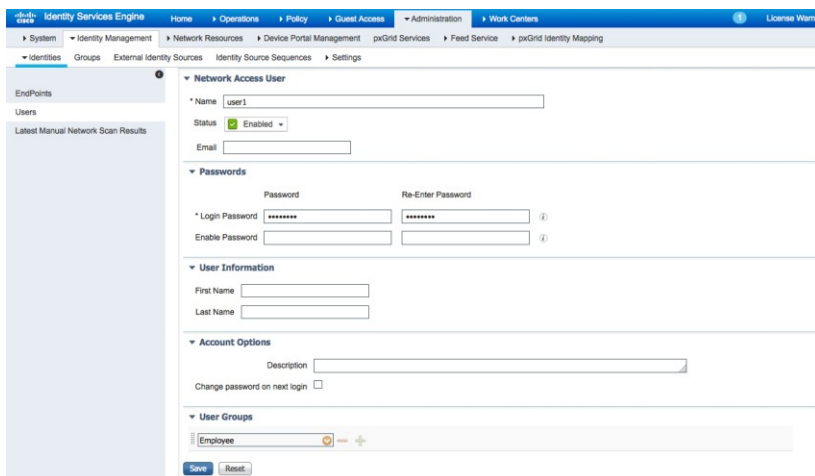
Microsoft Active Directory でユーザを使用していない場合は、ISE 内部ユーザをテストに使用できます。

ISE 内部ユーザの作成

ここでは、いくつかの内部 ISE ユーザを作成します (Active Directory でユーザを設定していない場合)。

ステップ 1 [管理 (Administration)] > [ID 管理 (Identity Management)] > [ID (Identity)] > [ユーザ (Users)] > [追加 (Add)] > [user1] を選択します。

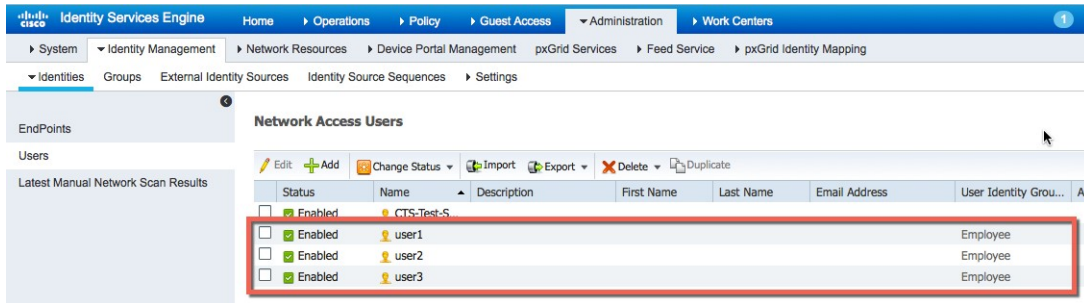
Employee Group に追加するパスワード情報を入力します。



ステップ 2 [保存 (Save)] を選択します。

ステップ 3 user2、user3 にも同じ手順を繰り返します。

ステップ 4 次のように表示されます。



認証

クライアント PC で RADIUS を実行して 802.1X 認証をシミュレートします。

ステップ 1 ユーザ認証をシミュレートします。

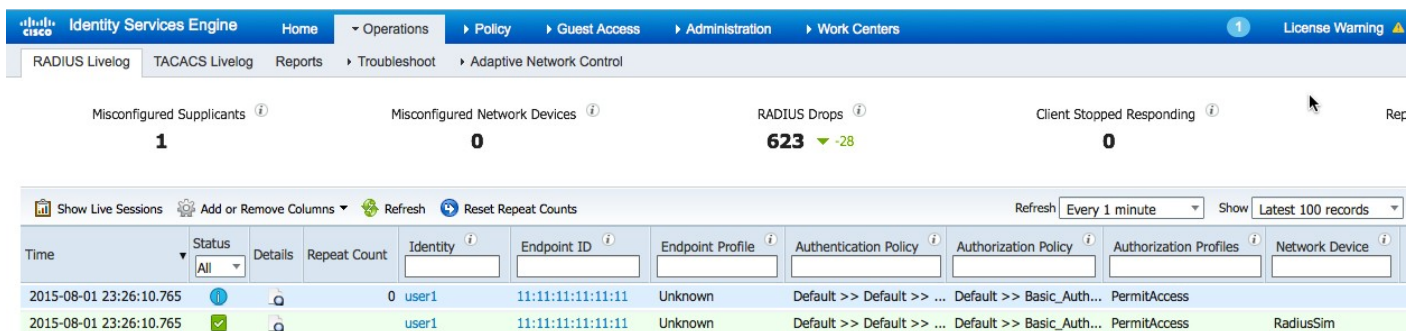
```
java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DCALLING_STATION_ID=11:11:11:11:11:11 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=192.168.1.60 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.98
```

認証のテスト

ステップ 1 ISE のパラメータについて、次の認証情報を入力します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DCALLING_STATION_ID=11:11:11:11:11:11 -DAUDIT_SESSION_ID=1001 -DFRAMED_IP_ADDRESS=192.168.1.100 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.23
AccessAccept code=2 id=1 length=107
authenticator=8e8e3217bee99d3f4bf38c21ba23d3e
Attributes=<
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:ise201/227764484/227
  vendorId=9 vsa=[profile-name=Unknown, ]
>
```

ステップ 2 ISE で認証を確認します。
 [運用 (Operations)] > [RADIUS Livelog] の順に選択します。



RADIUS シミュレータのパラメータ

パラメータ	デフォルト
-DUSERNAME	
-DPASSWORD	
-DCALLING_STATION_ID	
-DAUDIT_SESSION_ID	
-DRADIUS_SECRET	秘密 (Secret)
-DNAS_IP_ADDRESS	
-DFRAMED_IP_ADDRESS	
-DFRAMED_IP_MASK	
RadiusAccountingStop	
RadiusAccountingStart	
RadiusAuthentication	

pxGrid 2.0 サンプル スクリプト

このセクションでは、開発部門によって使用されるユニット テストの実施方法に加え、シスコとのソリューションの検証テストに使用されるテスト ケースの概要を説明します。pxGrid のサンプル スクリプトは、pxGrid を通じて利用可能なセッション情報および利用可能なクエリについて優れた参考情報を提供します。開発者は、これらのスクリプトを変更することで、関連するセッション情報を提供またはクエリすることができます。

このセクションには、次のことに基づいて 2 つのセットのテスト スイートがあります。1) pxGrid SDK から RADIUS シミュレータを使用すること。2) 802.1X が構成された状態で ISE 展開を使用すること。エンドポイント タイプ (モバイル デバイス、プリンタ、ラップトップなど) およびデバイスのセキュリティ ポスチャ (最新のマルウェア対策がインストールされているなど) の特定のために使用されるエンドポイント プロファイリングを可能にすることを含み、完全な ISE 統合機能をテストするには、このドキュメントで後から概略を示す、802.1X テスト スイートを使用してください。システム内でユーザと IP アドレスを関連付けるために単に IP - MAC - ユーザの関連付けだけがが必要な場合、RADIUS シミュレータのテストを使用できます。

802.1X スイートに対してテストを実行する場合は、RADIUS シミュレータを使用する場合と比較すると、テストのスーパーセットとなります。したがって、802.1X テスト スイートを使用する場合は、RADIUS シミュレータ ベースのテスト スイートを実行する必要はありません。

サンプルのテスト スクリプトについての概要を以下に説明します。

Multigroup Client (*pxGrid 1.3/1.4* の *register.sh* を置き換える): 複数のクライアント グループに pxGrid クライアントを接続および登録します。

(注) Register.sh は ISE 2.0 との上位互換性を備えています

Capability: pxGrid のインスタンスによってサポートされており、pxGrid クライアントがサブスクライブしているすべての機能またはパブリッシュされたトピックを一覧表示します。

EPS_Quarantine: レガシーの Endpoint Protection Service (EPS)/Adaptive Network Control (ISE 13/1.4 による、ISE 上の特定の IP アドレスに対する検疫アクション) を実行します。

(注) 登録済みの pxGrid クライアントは EPS クライアント グループに登録し、EndpointProtection Service 機能をサブスクライブします。

EPS_Unquarantine: レガシーの Endpoint Protection Service (EPS)/Adaptive Network Control (ISE 13/1.4 による、ISE 上の特定の MAC アドレスに対する検疫解除アクション) を実行します。

Identity_Group_Download: ISE でアクティブなセッションと関連付けられているユーザおよび ID グループをダウンロードします。

Session_Download: ISE からすべての一括セッション レコードまたはアクティブなセッションをダウンロードします。

Session_Query_By_IP: IP アドレスに基づいて、ISE からすべてのアクティブなセッションを取得します。

Session_Subscribe: セッション状態で変更されたものをサブスクライブします。

EndpointProfile_Query: ISE で構成されたすべてのエンドポイント プロファイル (プロファイリング ポリシー) を取得します。

EndpointSecurityGroup_Query: ISE で構成されたすべての TrustSec セキュリティ グループを取得します。

SecurtiyGroup_Subscribe: ISE で構成された TrustSec セキュリティグループの変更をサブスクライブします。

ANCaction_query: 検疫、修復、プロビジョニング、ポート閉鎖、ポート バウンスなどのカスタマイズされた pxGrid ANC 軽減アクションを提供します。

RADIUS シミュレータを使用するテスト スクリプト

Multigroupclient

検証

このテストは、サードパーティシステムが pxGrid の複数のクライアントグループ (Session、ANC) に登録 (認証および承認) できることを検証します。

定義

PxGrid クライアント登録では、サードパーティアプリケーション、セキュリティ デバイス、また、この場合は Linux ホストを、pxGrid コントローラの承認済みセッションまたは ANC グループに接続および登録します。admin や basic などの追加のグループを使用できますが、Admin グループは ISE 用に予約されており、pxGrid の管理承認が必要な Basic グループは、pxGrid のどの登録例でも使用されません。

すべての登録済み pxGrid クライアントは、[管理 (Administration)] の下の ISE pxGrid サービスで確認できます。

pxGrid クライアントは、ダイナミックピックでも示されるとおり、情報のパブリッシュまたはサブスクライブになることができます。ISE は情報を利用できず、コンテキストの共有は登録済みクライアント間で発生します。pxGrid クライアントが承認されたグループに対して正常に登録されると、pxGrid サンプル スクリプトによって決定されるとおり、クライアントは関連するセッション情報またはクエリを取得できます。

(注) これらの例で、pxGrid クライアントは、SessionDirectory、EndpointProtectionService、および TrustSecMetadata 機能をサブスクライブします。

例

この例では、pxGrid コントローラに対するセッション グループに、Linux ホストを pxGrid クライアントとして登録します。Linux ホストである SIM0 は、pxGrid クライアントのユーザ名です。また、ISE の登録済み pxGrid クライアントを表示します。

ステップ 1 multigroupclient スクリプトを実行します。

```
./multigroupclient.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果:

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session,ANC,
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
```

```
-----
10:33:58.911 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
10:34:03.470 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Create ANC Policy: ANC1438526035992 Result -com.cisco.pxgrid.model.anc.ANCResult@612fc6eb[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancPolicies=<null>
]
Session 1.1.1.2 not found
Connection closed
10:34:04.385 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jeppich$
```

使用法:

```
Usage: ./multigroupclient.sh [options]

Main options
  -a <PXGRID_HOSTNAMES> (comma separated hostnames)
  -u <PXGRID_USERNAME>
  -g <PXGRID_GROUP>
  -d <PXGRID_DESCRIPTION>

The followings are certificates options
  -k <PXGRID_KEYSTORE_FILENAME>
  -p <PXGRID_KEYSTORE_PASSWORD>
  -t <PXGRID_TRUSTSTORE_FILENAME>
  -q <PXGRID_TRUSTSTORE_PASSWORD>
If not specified, it defaults to use clientSample1.jks and rootSample.jks
Specifying values here can override the defaults

Custom config file can fill or override parameters
  -c <config_filename>
Config file are being sourced. Use these variables:
  PXGRID_HOSTNAMES
  PXGRID_USERNAME
  PXGRID_GROUP
  PXGRID_DESCRIPTION
  PXGRID_KEYSTORE_FILENAME
  PXGRID_KEYSTORE_PASSWORD
  PXGRID_TRUSTSTORE_FILENAME
  PXGRID_TRUSTSTORE_PASSWORD
```

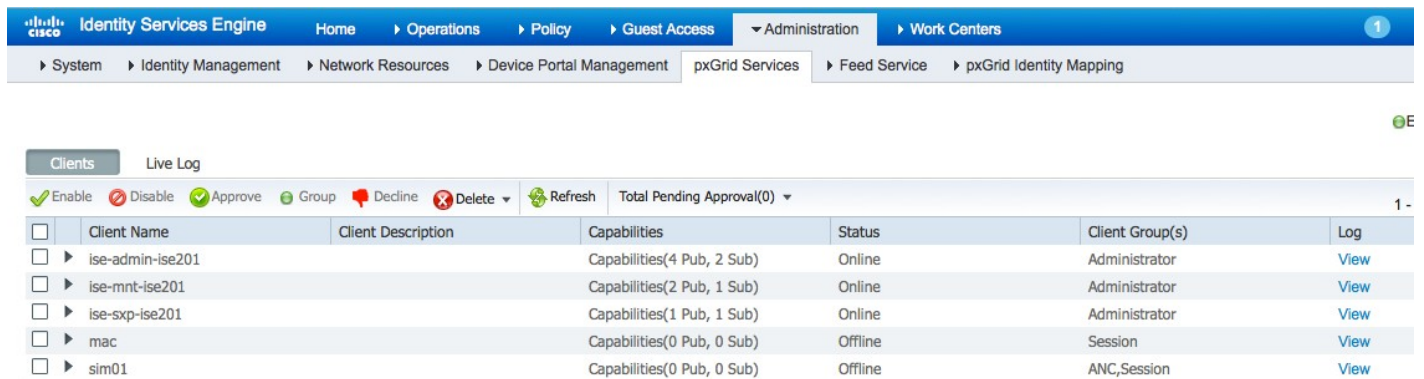
結果:

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session,ANC,Session
description=pxGrid
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
09:35:31.772 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
09:35:35.769 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Create ANC Policy: ANC1437658531354 Result -com.cisco.pxgrid.model.anc.ANCResult@612fc6eb[
```

```

ancStatus=SUCCESS
ancFailure=<null>
failureDescription=<null>
ancEndpoints=<null>
ancpolicies=<null>
]
Session 1.1.1.2 not found
Connection closed
    
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。pxGrid クライアント sim01 を、セッションクライアントグループに登録します。デフォルトで、pxGrid Adaptive Network Control (ANC) 軽減アクションに必要な ANC は、追加されています。



Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-sxp-ise201		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
mac		Capabilities(0 Pub, 0 Sub)	Offline	Session	View
sim01		Capabilities(0 Pub, 0 Sub)	Offline	ANC,Session	View

セッションのサブスクリプション

検証

このテストは、サードパーティシステムが pxGrid コントローラに正常に登録されると、pxGrid クライアントが ISE によってパブリッシュされた Session Directory をサブスクリプションしてリアルタイムで通知を受信することを検証します。

定義

クライアントがセッションおよび ANC グループに対して pxGrid コントローラによって正常に登録および承認されると、クライアントは機能をサブスクリプションして認証されたユーザ向けの関連するセッション情報を取得します。ISE MnT ノードは、ISE Session Directory を pxGrid コントローラに対してトピックとしてパブリッシュします。pxGrid クライアントはこの機能をサブスクリプションし、認証されたユーザのアクティブなセッションおよび通知をリアルタイムで取得します。

例

pxGrid クライアントは、Session Directory をサブスクリプションし、user1、user2、および user3 の認証からリアルタイムで通知を受信し、利用可能なコンテキスト情報を記録します。

ステップ 1 session_subscribe スクリプトを実行します。

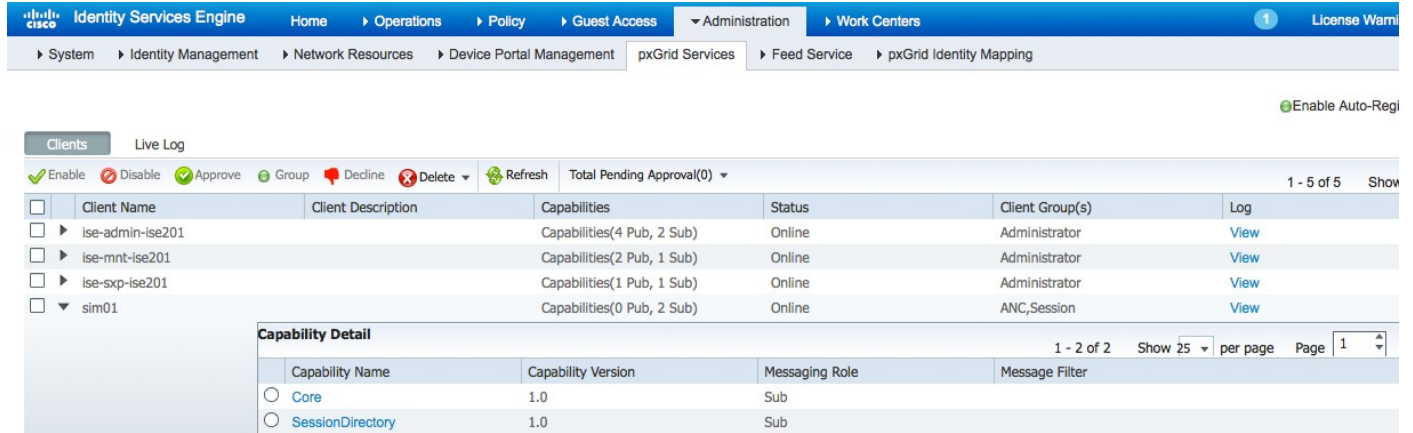
```
./session_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
10:41:17.909 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex.'1.0.0.0/255.0.0.0,1234::/16,...' or <enter> for no filter): 10:41:19.311 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected
Connected
```

ステップ 2 [管理(Administration)] > [pxGrid サービス(pxGrid Services)] の順に選択します。

pxGrid クライアントである SIM01 は Session Directory を正常にサブスクライブしました



Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-sxp-ise201		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
sim01		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

ステップ 3 クライアント PC で RADIUS シミュレータを実行し、user1、user2、および user3 の IEEE 802.1X 認証をシミュレートします。

ステップ 4 RadiusAuthentication から開始する user1 に対して RADIUS シミュレータを実行します。

(注) ユーザ名、audit_session_id、acct_session_id、calling_station_id、framed_ip_address はユーザごとに異なることが重要です。配置の順序が必要です。

また、acct_session_id を含めることも重要です。これを含めないと、以前のユーザのセッションが表示されます。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=1001 -DACCT_SESSION_ID=2001 -DCALLING_STATION_ID=11:11:11:11:11:11 -DFRAMED_IP_ADDRESS=192.168.1.100 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.23
AccessAccept code=2 id=1 length=106
authenticator=dabbd17e2179ce58115dc6cdef1aa73
Attributes=<
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:ise201/227903462/81
  vendorId=9 vsa=[profile-name=Unknown,]
>
```

ステップ 5 RadiusAccountingStart を持つ user1 に RADIUS シミュレータを実行します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=1001 -DACCT_SESSION_ID=2001 -DCALLING_STATION_ID=11:11:11:11:11:11 -DFRAMED_IP_ADDRESS=192.168.1.100 -DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStart 192.168.1.23
AccountingResponse code=5 id=1 length=20
authenticator=a05d59f8e420a7ed47b420f199f5c692
Attributes=<
>
```

ステップ 6 RadiusAuthentication を持つ user2 に RADIUS シミュレータを実行します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user2 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=3001 -DACCT_SESSION_ID=4001 -DCALLING_STATION_ID=22:22:22:22:22:22 -DFRAMED_IP_ADDRESS=192.168.1.101 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.23
AccessAccept code=2 id=1 length=106
authenticator=ce5d7b607e296e47a6199ad2d99dc84
Attributes=<
  UserName=user2
  State=ReauthSession:3001
  Class=CACS:3001:ise201/227903462/75
  vendorId=9 vsa=[profile-name=Unknown,]
>
```

ステップ 7 RadiusAccounting を持つ user2 に RADIUS シミュレータを実行します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user2 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=3001 -DACCT_SESSION_ID=4001 -DCALLING_STATION_ID=22:22:22:22:22:22 -DFRAMED_IP_ADDRESS=192.168.1.101 -DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStart 192.168.1.23
AccountingResponse code=5 id=1 length=20
authenticator=7634b93f66e6308c1ecc7c3056e33a55
Attributes=<
>
```

ステップ 8 RadiusAuthentication を持つ user3 に RADIUS シミュレータを実行します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user3 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=5001 -DACCT_SESSION_ID=5002 -DCALLING_STATION_ID=33:33:33:33:33:33 -DFRAMED_IP_ADDRESS=192.168.1.102 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthentication 192.168.1.23
AccessAccept code=2 id=1 length=106
authenticator=7b9e79da6d6899593d74833752eb8e
Attributes=<
  UserName=user3
  State=ReauthSession:5001
  Class=CACS:5001:ise201/227903462/84
  vendorId=9 vsa=[profile-name=Unknown,]
>
```

ステップ 9 RadiusAccountingStart を持つ user3 に RADIUS シミュレータを実行します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user3 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=5001 -DACCT_SESSION_ID=5002 -DCALLING_STATION_ID=33:33:33:33:33:33 -DFRAMED_IP_ADDRESS=192.168.1.102 -DFRAMED_IP_MASK=255.255.255.0 RadiusAccountingStart 192.168.1.23
AccountingResponse code=5 id=1 length=20
authenticator=6f51ae332ff253622e951bb69dcb918
Attributes=<
>
```

ステップ 10 強調表示されている各ユーザセッションについて、以下の利用可能なコンテキスト情報に注目してください。これらのセッションオブジェクトは、サードパーティアプリケーションで使用して、イベントに関する詳細なコンテキストを取得できます。

```
./session_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:28:19.187 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex. '1.0.0.0/255.0.0.0,1234::/16,...' or <enter> for no filter): 11:28:20.547 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected

press <enter> to disconnect...session notification:
Session={ip=[192.168.1.101], Audit Session Id=3001, User Name=user2, AD User DNS Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=22:22:22:22:22:22, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37, RADIUSAVPairs=[ Acct-Session-Id=4001], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun Aug 02 12:27:12 EDT 2015}

session notification:
Session={ip=[192.168.1.100], Audit Session Id=1001, User Name=user1, AD User DNS Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=11:11:11:11:11:11, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37, RADIUSAVPairs=[ Acct-Session-Id=2001], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun Aug 02 12:30:44 EDT 2015}
```

```
session notification:
Session={ip=[192.168.1.102], Audit Session Id=5001, User Name=user3, AD User DNS Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=33:33:33:33:33:33, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37, RADIUSAVPairs=[ Acct-Session-Id=5002], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun Aug 02 12:35:59 EDT 2015}
```

ステップ 11 イベントを表示するには、[運用 (Operations)] > [RADIUS LiveLog] の順に選択します。

Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles	Network Device
2015-08-02 16:35:59.597	🟢		1	user3	33:33:33:33:33:33	Unknown	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	
2015-08-02 16:34:43.062	🟢			user3	33:33:33:33:33:33		Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	RadiusSim
2015-08-02 16:30:44.458	🟢		1	user1	11:11:11:11:11:11	Unknown	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	
2015-08-02 16:27:12.180	🟢		1	user2	22:22:22:22:22:22	Unknown	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	
2015-08-02 16:26:13.273	🟢			user2	22:22:22:22:22:22	Unknown	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	RadiusSim
2015-08-02 16:24:34.417	🔴			CTS-Test-Server			Default >> Default >> ...			Switch
2015-08-02 16:24:33.184	🔴									
2015-08-02 16:04:56.767	🟢			user1	11:11:11:11:11:11	Unknown	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	RadiusSim

セッションのダウンロード

検証

このテストは、アクティブなユーザセッションの一括セッションダウンロードを実行するためのサードパーティシステムの機能を検証します。

定義

セッションダウンロードスクリプトが、パブリッシュされた ISE ノードから一括セッションレコードをダウンロードします。

例

pxGrid クライアントは、ISE MnT ノードからアクティブなセッションをダウンロードします。

```
./session_download.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:23:49.800 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex.'1.0.0.0/255.0.0.0,1234::/16...' or <enter> for no filter): 12:23:51.043 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected

Start time (ex.'2015-01-31 13:00:00' or <enter> for no start time):
End time (ex.'2015-01-31 13:00:00' or <enter> for no end time):
Session={ip=[192.168.1.31], Audit Session Id=0A0000010000002803DBE3C1, User Name=LAB6\jpeppich, 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:29:79:02:A8, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint
Profile=Windows7-Workstation, NAS IP=192.168.1.2, NAS Port=GigabitEthernet1/0/12, RADIUSAVPairs=[ Acct-
Session-Id=00000053], Posture Status=NonCompliant, Posture Timestamp=Sat Aug 01 15:15:20 EDT 2015, Session
Last Update Time=Sat Aug 01 15:15:22 EDT 2015}
Session={ip=[192.168.1.100], Audit Session Id=1001, User Name=user1, AD User DNS Domain=null, AD Host DNS
Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=11:11:11:11:11:11,
Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37,
RADIUSAVPairs=[ Acct-Session-Id=2001], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun
Aug 02 12:30:44 EDT 2015}
Session={ip=[192.168.1.101], Audit Session Id=3001, User Name=user2, AD User DNS Domain=null, AD Host DNS
Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=22:22:22:22:22:22,
Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37,
RADIUSAVPairs=[ Acct-Session-Id=4001], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun
Aug 02 12:27:12 EDT 2015}
Session={ip=[192.168.1.102], Audit Session Id=5001, User Name=user3, AD User DNS Domain=null, AD Host DNS
Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=33:33:33:33:33:33,
Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37,
RADIUSAVPairs=[ Acct-Session-Id=5002], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun
Aug 02 12:35:59 EDT 2015}
Session count=4
Connection closed
12:23:59.504 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jpeppich$
```

IPによるセッションのクエリ

検証

このテストは、pxGridを経由し、特定のIPアドレスに関連する指示されたクエリを実行し、ユーザからコンテキスト情報を返すサードパーティシステムの能力を検証します。

定義

Session Query by IP スクリプトは、IPアドレスにより、認証されたユーザのセッション情報を取得します。

例

この例では、エンドユーザの IP アドレス(192.168.1.100)を入力して、エンドユーザのセッション情報を取得します。

ステップ 1 session_query_by_ip スクリプトを実行します。

```
./session_query_by_ip.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:30:45.610 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:30:46.935 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
IP address (or <enter> to disconnect): 192.168.1.100
Session={ip=[192.168.1.100], Audit Session Id=1001, User Name=user1, AD User DNS Domain=null, AD Host DNS
Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station id=11:11:11:11:11:11,
Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown, NAS IP=192.168.1.37,
RADIUSAVPairs=[ Acct-Session-Id=2001], Posture Status=null, Posture Timestamp=, Session Last Update Time=Sun
Aug 02 12:30:44 EDT 2015}
IP address (or <enter> to disconnect):
```

EndpointProfile のサブスクリプション

検証

このテストは、パブリッシュされた Endpoint Profile トピックをサブスクリプションするためのサードパーティシステムの能力を検証します。

定義

登録済みの pxGrid クライアントは EndpointProfileMetaData 機能にサブスクリプションし、グローバル プロファイル ポリシーの変更または修正を取得します。セッション通知には、エンドポイント プロファイル ID、名前、および完全修飾名が含まれます

例

この例では、ユーザの PC の静的 MAC アドレスに基づいて、pxGrid の EndpointProfile Example ポリシーが作成されます。pxGrid クライアントが EndpointprofileMetadata 機能をサブスクリプションし、ISE プロファイリング ポリシーに何らかの修正が発生したとき、リアルタイムで実行中の Linux スクリプトのセッション通知を確認します。

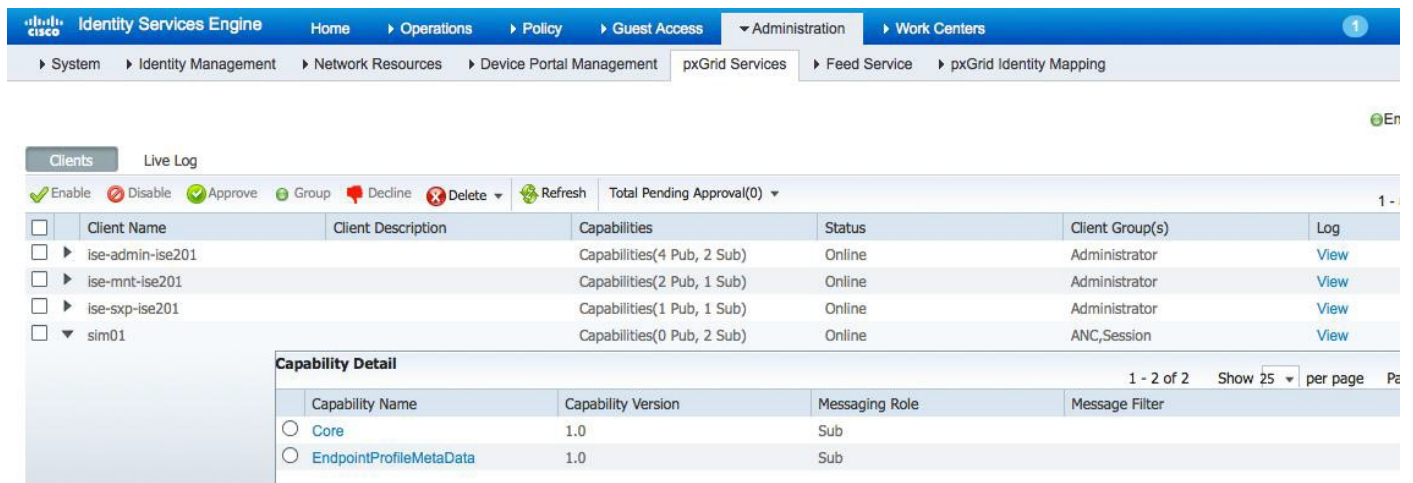
ステップ 1 endpointprofile_subscribe スクリプトを実行します。

```
./endpointprofile_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:41:22.280 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:41:23.552 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...
```

ステップ 2 [管理 (Administrations)] > [pxGrid サービス (pxGrid Services)] の順に選択します。 pxGrid クライアントは EndpointProfileMetaData 機能をサブスクライブしています



Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
<input type="checkbox"/> ▶ ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
<input type="checkbox"/> ▶ ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
<input type="checkbox"/> ▶ ise-sxp-ise201		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
<input type="checkbox"/> ▼ sim01		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

ステップ 3 [ポリシー (Policy)] > [プロファイリング (Profiling)] > [追加 (Add)] の順に選択します。 ポリシーの名前と説明を入力します。 [If 条件 (If Condition)] > [新しい条件の作成 (Create New Condition)] > [IP] > {ネットワークにアクセスするデバイスの IP アドレスを入力} [送信 (Submit)] を選択します。

Identity Services Engine Home > Operations > Policy > Guest Access > Administration > Work Centers

Authentication Authorization Profiling Posture Client Provisioning > Policy Elements

Profiler Policy List > **New Profiler Policy**

Profiler Policy

* Name: Add_Device Description: trigger_endpointprofile_subscript_pxGrid

Policy Enabled:

* Minimum Certainty Factor: 10 (Valid Range 1 to 65535)

* Exception Action: NONE

* Network Scan (NMAP) Action: NONE

Create an Identity Group for the policy: Yes, create matching Identity Group
 No, use existing Identity Group hierarchy

* Parent Policy: NONE

* Associated CoA Type: Global Settings

System Type

Rules

If Condition: Conditions Then Certainty Factor Increases 10

Condition Name	Expression
	IP:ip CONTAINS 192.168.1.100

ステップ 4 先ほど作成し、追加されたプロファイリング ポリシー エンドポイントプロファイルのサブスクリプション通知を受信します。

```
./endpointprofile_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:41:22.280 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:41:23.552 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...EndpointProfileChangedNotification (changetype=ADD) Device profile :
id=8c8f42b0-393f-11e5-ac86-000c297fb12a, name=Add_Device, fqname=Add_Device
```


ID グループのダウンロード

検証

このテストは、ユーザ ID 情報の一括ダウンロードを実行するためのサードパーティシステムの機能を検証します。

定義

Identity Group ダウンロード スクリプトは、ユーザ グループ情報の一括セッションレコードとユーザ グループ マッピングをセッション ディレクトリからダウンロードします。これらのグループには、ISE ID グループおよびプロファイル済みグループが含まれています。

例

Identity Group ダウンロード スクリプトを使用して ISE MnT Node パブリッシュャからすべてのグループ情報をダウンロードします。

ステップ 1 identity_group_download スクリプトを実行します。

```
./identity_group_download.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:01:21.977 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:01:23.242 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
user=host/jeppich-PC.lab6.com groups=Workstation
user=LAB6\jeppich groups=Workstation
user=user1 groups=User Identity Groups:Employee,Unknown
user=user2 groups=User Identity Groups:Employee,Unknown
user=user3 groups=User Identity Groups:Employee
User count=5
Connection closed
```

セキュリティグループのクエリ

検証

このテストは、ISE のすべてのセキュリティグループ タグを取得するためのサードパーティシステムの能力を検証します。

定義

セキュリティグループ クエリ スクリプトは、TrustSecMetadata 機能トピックを介して ISE で設定されたセキュリティグループ タグ (SGT) を公開します。これにより、一意の ID、セキュリティグループ タグの値、および記述に基づいて ISE で設定されたすべての SGT を取得するためのクエリ メソッドが提供されます。

例

この例では、セキュリティグループクエリ スクリプトがすべてのセキュリティグループ タグのコンテキスト情報をダウンロードします。このスクリプトは、ISE からすべての TrustSec セキュリティグループ セッション情報を取得します。これには、TrustSec タグの名前、固有識別子、記述、および値が含まれます。

セキュリティグループ タグについてのクエリを送信します。

ステップ 1 securitygroup_query スクリプトを実行します。

```
./securitygroup_query.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:04:24.807 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:04:26.071 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
SecurityGroup : id=65fddc70-2a34-11e5-82cb-005056bf2f0a, name=Unknown, desc=Unknown Security Group, tag=0
SecurityGroup : id=660aadb0-2a34-11e5-82cb-005056bf2f0a, name=ANY, desc=Any Security Group, tag=65535
SecurityGroup : id=669e6230-2a34-11e5-82cb-005056bf2f0a, name=SGT_Auditor, desc=Auditor Security Group,
tag=9
SecurityGroup : id=66bdd110-2a34-11e5-82cb-005056bf2f0a, name=SGT_BYOD, desc=BYOD Security Group, tag=15
SecurityGroup : id=66dd3ff0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Contractor, desc=Contractor Security
Group, tag=5
SecurityGroup : id=66fcd5e0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Developer, desc=Developer Security Group,
tag=8
SecurityGroup : id=671a21e0-2a34-11e5-82cb-005056bf2f0a, name=SGT_DevelopmentServers, desc=Development
Servers Security Group, tag=12
SecurityGroup : id=673c9e00-2a34-11e5-82cb-005056bf2f0a, name=SGT_Employee, desc=Employee Security Group,
tag=4
SecurityGroup : id=6759ea00-2a34-11e5-82cb-005056bf2f0a, name=SGT_Guest, desc=Guest Security Group, tag=6
SecurityGroup : id=6775d670-2a34-11e5-82cb-005056bf2f0a, name=SGT_NetworkServices, desc=Network Services
Security Group, tag=3
```

```
SecurityGroup : id=67959370-2a34-11e5-82cb-005056bf2f0a, name=SGT_PCIServers, desc=PCI Servers Security
Group, tag=14
SecurityGroup : id=67b3a2c0-2a34-11e5-82cb-005056bf2f0a, name=SGT_PointOfSale, desc=PointOfSale Security
Group, tag=10
SecurityGroup : id=67d50d70-2a34-11e5-82cb-005056bf2f0a, name=SGT_ProductionServers, desc=Production Servers
Security Group, tag=11
SecurityGroup : id=67f16f10-2a34-11e5-82cb-005056bf2f0a, name=SGT_ProductionUser, desc=Production User
Security Group, tag=7
SecurityGroup : id=680df7c0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Quarantine, desc=Quarantine Security
Group, tag=255
SecurityGroup : id=682a5960-2a34-11e5-82cb-005056bf2f0a, name=SGT_TestServers, desc=Test Servers Security
Group, tag=13
SecurityGroup : id=68461ec0-2a34-11e5-82cb-005056bf2f0a, name=SGT_TrustSecDevices, desc=TrustSec Devices
Security Group, tag=2
SecurityGroup : id=1bea1190-37f8-11e5-aeb1-000c297fb12a, name=3750x, desc=, tag=16
SecurityGroup : id=e855d7c0-3805-11e5-aeb1-000c297fb12a, name=ASA5505, desc=, tag=17
SecurityGroup : id=c0e5a9d0-381a-11e5-aeb1-000c297fb12a, name=Mobile_Users, desc=, tag=18
Connection closed
13:04:26.450 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jeppich$
```

セキュリティグループのサブスクリプション

検証

このテストは、pxGrid を介して SecurityGroup トピックをサブスクリプションするためのサードパーティシステムの能力を検証します。

定義

セキュリティグループ スクリプトは、TrustsecMetaDataCapability トピックを介して ISE で設定されたセキュリティグループ タグ (SGT) を公開します。セキュリティグループが追加/更新/削除されると、セキュリティグループの変更通知がスクリプト セッション通知に表示されます。

例

セキュリティグループ サブスクリプション スクリプトは、ISE TrustSec ポリシーの変更をサブスクリプションします。ISE にセキュリティグループ タグを追加します。pxGrid クライアントが TrustSecMetadataCapability トピックをサブスクリプションしているため、通知を受信します。

ステップ 1 security_subscribe スクリプトを実行します。

```
./securitygroup_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

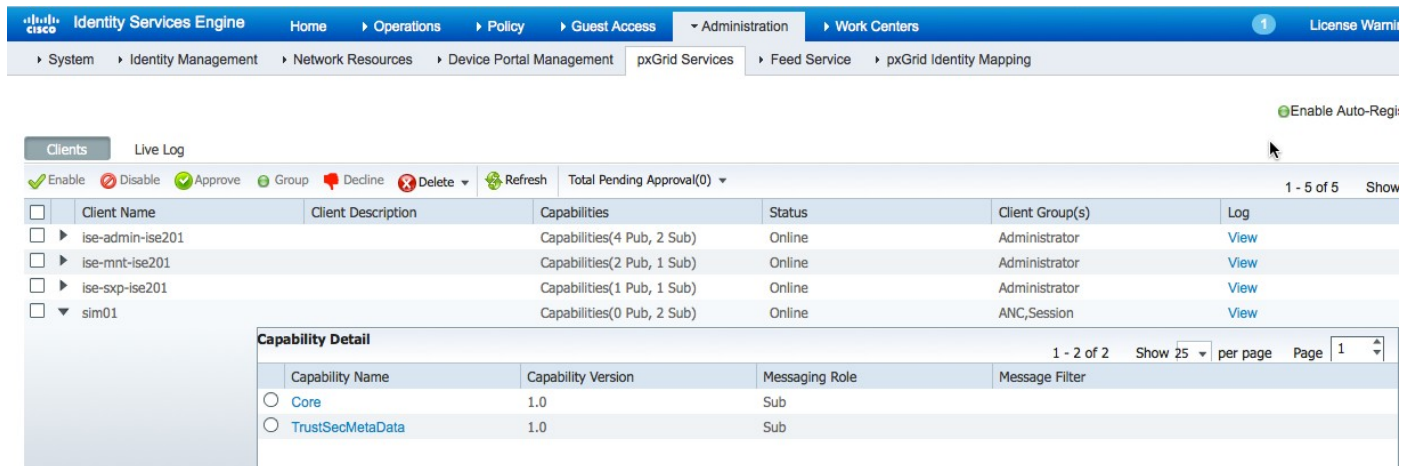
```

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----

13:07:12.322 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:07:13.613 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...

```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Service)] の順に選択します。
smc01 が TrustSecMetadata 機能に登録されたことを確認します。



The screenshot shows the Cisco Identity Services Engine (ISE) Administration console. The breadcrumb navigation is: Home > Operations > Policy > Guest Access > Administration > Work Centers > pxGrid Services > pxGrid Identity Mapping. The 'Clients' tab is selected, showing a table of clients. The client 'sim01' is expanded, and a 'Capability Detail' window is open for it.

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-sxp-ise201		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
sim01		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

ステップ 3 [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [セキュリティ グループ (Security Groups)] > [新しいセキュリティグループ (New Security Group)] > [SMC01] の順に選択します。

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers

TrustSec Device Administration

Overview Authentication Policy Authorization Policy Components Policy SXP Reports Settings

Security Groups List > New Security Group

Security Groups

* Name
SIM01

* Icon

Description

Security Group Tag (Dec / Hex): 19/0013
Generation Id: 0

Submit Cancel

ステップ 4 セキュリティグループ タグ通知が表示されます。

```
./securitygroup_subscribe.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:07:12.322 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:07:13.613 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...SecurityGroupChangeNotification (changetype=ADD) SecurityGroup : id=994e2140-3941-11e5-ac86-000c297fb12a, name=SIM01, desc=, tag=19
```

エンドポイント プロファイルのクエリ

検証

このテストは、ISE で設定された、すべての有効にされたプロファイルを取得するためのサードパーティシステムの能力を検証します。

定義

endpointprofile_query スクリプトは、ISE で設定された、すべての有効にされたエンドポイント プロファイルを取得するためのクエリ メソッドを提供します。また、エンドポイント プロファイル ID、名前、および完全修飾名を提供します。また、サブスクリイバは、エンドポイント プロファイルが ISE で追加/更新/削除された場合にも通知されます。

例

The endpointprofile クエリ スクリプトは、ISE のすべての有効にされたプロファイルを取得します。

ステップ 1 endpointprofile_query スクリプトを実行します。

```
./endpointprofile_query.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:14:11.358 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:14:12.631 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Endpoint Profile : id=8c8f42b0-393f-11e5-ac86-000c297fb12a, name=Add_Device, fqname Add_Device
Endpoint Profile : id=4d852be0-2a33-11e5-82cb-005056bf2f0a, name=Android, fqname Android
Endpoint Profile : id=4dc7b320-2a33-11e5-82cb-005056bf2f0a, name=Apple-Device, fqname Apple-Device
Endpoint Profile : id=4e190770-2a33-11e5-82cb-005056bf2f0a, name=Apple-iDevice, fqname Apple-Device:Apple-iDevice
Endpoint Profile : id=4e452080-2a33-11e5-82cb-005056bf2f0a, name=Apple-iPad, fqname Apple-Device:Apple-iPad
Endpoint Profile : id=4e6f8be0-2a33-11e5-82cb-005056bf2f0a, name=Apple-iPhone, fqname Apple-Device:Apple-iPhone
```

機能

検証

このテストは、ISE でパブリッシュされたすべての機能を取得するためのサードパーティシステムの能力を検証します。

定義

機能スクリプトは、ISE でパブリッシュされたすべての関心のあるトピックを取得します。

例

機能スクリプトは、情報トピックまたはクライアントがパブリッシュまたはサブスクライブできる機能を取得します。

ステップ 1 機能スクリプトを実行します。

```
./capability_query.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t
```

結果

```
alpha_root.jks -q cisco123
----- properties -----
  version=1.0.2-30-SNAPSHOT
  hostnames=192.168.1.23
  username=SIM01
  group=null
  description=null
  keystoreFilename=alpha.jks
  keystorePassword=cisco123
  truststoreFilename=alpha_root.jks
  truststorePassword=cisco123
-----
13:16:57.359 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:16:58.607 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
capability=SessionDirectory, version=1.0
capability=GridControllerAdminService, version=1.0
capability=EndpointProtectionService, version=1.0
capability=IdentityGroup, version=1.0
capability=EndpointProfileMetaData, version=1.0
capability=TrustSecMetaData, version=1.0
capability=AdaptiveNetworkControl, version=1.0
capability=Core, version=1.0
Connection closed
13:16:58.659 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jeppich$
```

ID グループのクエリ

検証

このテストは、指定されたユーザから ISE ID グループ情報を取得するためのサードパーティシステムの能力を検証します。

定義

ID グループ クエリ スクリプトは、ISE ID グループ情報を取得します。

例

user1、user2、および user3 は、ISE ID グループ情報のためにクエリされます

ステップ 1 identity_group_query_script を実行します。

```
./identity_group_query.sh -a 192.168.1.23 -u SIM01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM01
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:18:59.446 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:19:00.755 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
user name (or <enter> to disconnect): user1
group=User Identity Groups:Employee,Unknown
user name (or <enter> to disconnect): user2
group=User Identity Groups:Employee,Unknown
user name (or <enter> to disconnect): user3
group=User Identity Groups:Employee
user name (or <enter> to disconnect):
```


ID グループのサブスクライブ

検証

このテストは、ISE によってパブリッシュされた ID トピックをサブスクライブするため、および通知を受信するためのサードパーティシステムの能力を検証します。

定義

ID グループ トピックをサブスクライブすると、pxGrid クライアントは、802.1X 以外のイベントについて通知を受信できません。

例

ISE に内部ネットワークのユーザが作成され、イベントをトリガーするゲスト ポータルをテストするために使用されます

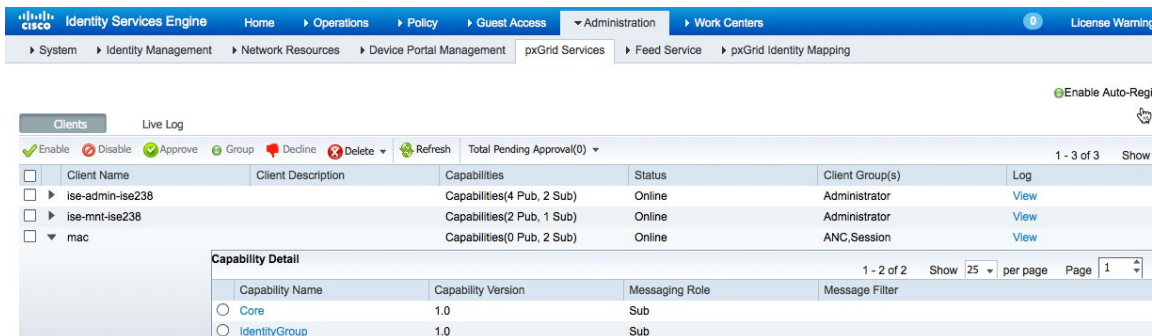
ステップ 1 identity_group_subscribe スクリプトを実行します。

```
/identity_group_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:20:22.839 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:20:24.468 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択して、サブスクライブされた ID グループ セッションを表示します。



Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
mac		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

ステップ 3 ゲスト ポータルに使用して従業員をトリガーするために ISE ID ユーザを作成します。

ステップ 4 デフォルトのセルフ サービス ポータルのテストを使用して、ユーザおよび関連付けられた ID グループをリアルタイムで検証するため、[ゲスト アクセス (Guest Access)] > [構成 (Configure)] > [ゲスト ポータル (Guest Portals)] > [ポータル (Portal)] の順に選択して、URL をテストします。

ステップ 5 [ポータル (Portal)] テストをクリックして、入力した ID グループのユーザ値を入力します。

ステップ 6 [サインオン (Sign On)] をクリックします。

ステップ 7 ID ユーザおよびグループ通知が表示されます。

```
./identity_group_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:20:22.839 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:20:24.468 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...user=jsmith
group=Employee
```

EPS_Quarantine/EPS_UnQuarantine

検証

このテストは、ネットワーク上のエンドポイントの検疫またはネットワーク切断アクションを実行するためのサードパーティシステムの機能を検証します。これはまた、MAC アドレスによってエンドポイントの検疫を解除するためのサードパーティシステムの機能を検証します。

定義

pxGrid クライアントは、承認済みの EPS セッション グループに登録し、ISE によってパブリッシュされる EndPointProtection サービス機能をサブスクライブし、認証されたデバイスの IP アドレスを検疫し、認証されたデバイスの検疫を、MAC アドレスに基づいて解除します。

例

クライアント user1 は、承認済みの EPS グループに登録し、EndpointProtectionService 機能をサブスクライブします。eps 検疫スクリプトが、IP アドレスで user1 を検疫します。認可変更 (CoA) をシミュレートするために DynAuthListener が使用され、検疫/検疫解除軽減アクションが実行されます。エンドポイントの IP アドレスを検疫するために eps_quarantine スクリプトが実行されます。MAC アドレスによってエンドポイントの検疫を解除するため、eps_unquarantine スクリプトが実行されます。pxGrid クライアントは、EndPointProtection サービス機能をサブスクライブしていることに注意してください。

ステップ 1 multigroupclient スクリプトを実行します。

```
./multigroupclient.sh -a 192.168.1.23 -u SIM02 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -g EPS -d RADIUSIMEPS Tests
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM02
group=Session,ANC,EPS
description=RADIUSIMEPS
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:54:57.950 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:54:59.800 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Create ANC Policy: ANC1438538097569 Result - com.cisco.pxgrid.model.anc.ANCResult@612fc6eb[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=<null>
]
Session 1.1.1.2 not found
Connection closed
13:55:00.434 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jeppich$
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。

pxGrid クライアントが EPS クライアント グループに登録します。

The screenshot shows the Cisco Identity Services Engine Administration console. The navigation path is: Administration > pxGrid Services. The 'Clients' tab is active, displaying a table of registered clients. The 'sim02' client is highlighted with a red box.

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-sxn-ise201		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
sim02	RADIUSIMEPS	Capabilities(0 Pub, 0 Sub)	Offline	ANC, EPS, Session	View
mac		Capabilities(0 Pub, 0 Sub)	Offline	Session	View
sim01		Capabilities(0 Pub, 0 Sub)	Offline	ANC, Session	View

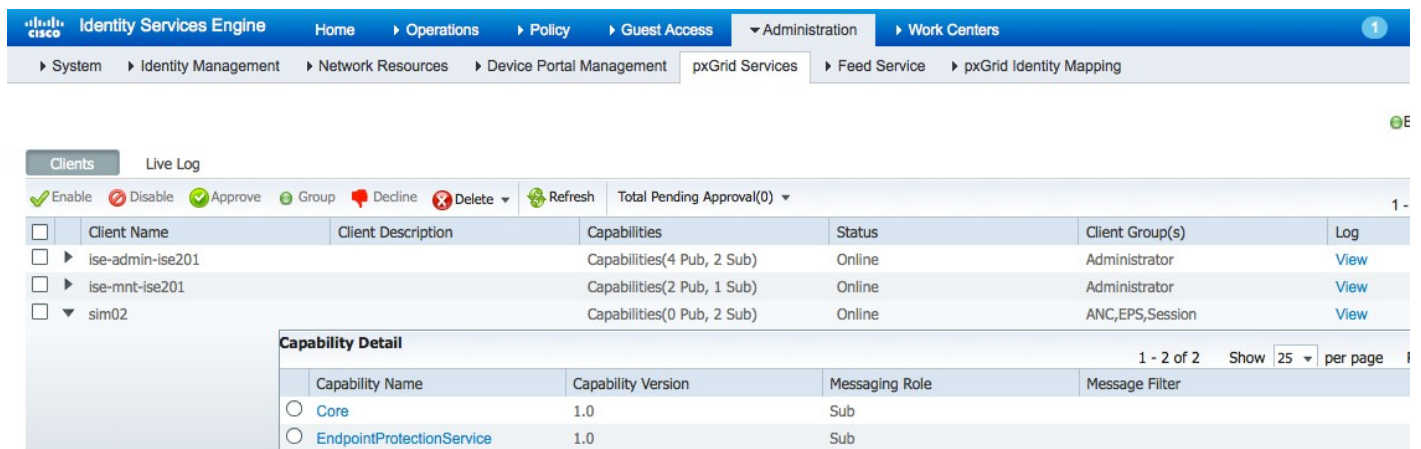
ステップ 3 PC で DynAuthListener を実行します。

```
java -cp RadiusSimulator.jar DynAuthListener
```

次が表示されます。

```
C:\sim>java -cp RadiusSimulator.jar DynAuthListener
DynAuthListener listening
```

ステップ 4 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。 pxGrid クライアントは EndPointProtection サービス機能をサブスクライブしています



Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
sim02		Capabilities(0 Pub, 2 Sub)	Online	ANC, EPS, Session	View

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

ステップ 5 eps_quarantine スクリプトを実行します。

```
./eps_quarantine.sh -a 192.168.1.23 -u SIM02 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM02
group=EPS
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
14:04:41.263 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
14:04:42.619 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
IP address (or <enter> to disconnect): 192.168.1.100
IP address (or <enter> to disconnect):
```

ステップ 6 DynAuthListener が検疫イベントを受信します。

```
C:\sim>java -cp RadiusSimulator.jar DynAuthListener
DynAuthListener listening
Received from /192.168.1.23:38085
DisconnectRequest code=40 id=1 length=104
authenticator=8216c5c449b45310a0317bfe5c1f12
Attributes=<
  NASIPAddress=192.168.1.37
  CallingStationID=11:11:11:11:11:11
  Unknown code=49 length=4
  EventTimestamp=Sun Aug 02 15:02:55 EDT 2015
  MessageAuthenticator=c74125fc42845e8facb673086525446
  vendorId=9 vsa=[audit-session-id=1001, ]
>
```

ステップ 7 PC 上で別の CMD ウィンドウを開き、RADIUS シミュレータを実行して user1 を認証します。

```
C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=1001 -DACCT_SESSION_ID=2001 -DCALLING_STATION_ID=11:11:11:11:11:11 -DFRAMED_IP_ADDRESS=192.168.1.100 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthenticat
tion 192.168.1.23
AccessAccept code=2 id=1 length=146
authenticator=2cff72c97b6b1cbd6839a224ae566af0
Attributes=<
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:ise201/227903462/89
  vendorId=9 vsa=[cts:security-group-tag=0014-0, ]
  vendorId=9 vsa=[profile-name=Add_Device, ]
>
```

ステップ 8 DynAuthListener が検疫イベントを受信します。

```
Received from /192.168.1.23:38085
DisconnectRequest code=40 id=2 length=104
authenticator=24151f8209cc58244112d2747aae92
Attributes=<
  NASIPAddress=192.168.1.37
  CallingStationID=11:11:11:11:11:11
  Unknown code=49 length=4
  EventTimestamp=Sun Aug 02 15:22:24 EDT 2015
  MessageAuthenticator=4cb295ea4fd8333c97bf9e21b04454
  vendorId=9 vsa=[audit-session-id=1001, ]
>
```

ステップ 9 [運用 (Operations)] > [RADIUS Livelog] を選択します。

ユーザは検疫されています

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers License Warning

RADIUS Livelog TACACS Livelog Reports Troubleshoot Adaptive Network Control

Misconfigured Supplicants 0 Misconfigured Network Devices 0 RADIUS Drops 45 Client Stopped Responding 0

Show Live Sessions Add or Remove Columns Refresh Reset Repeat Counts Refresh Every 1 minute Show Latest 100 records

Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles	Network Device
2015-08-02 19:17:00.214	✖			CTS-Test-Server			Default >> Default >> ...			Switch
2015-08-02 19:15:27.365	ⓘ		0	user1	11:11:11:11:11:11	Add_Device	Default >> Default >> ...	Default >> EPS_Legacy	Quarantine	
2015-08-02 19:15:27.365	✔			user1	11:11:11:11:11:11	Add_Device	Default >> Default >> ...	Default >> EPS_Legacy	Quarantine	RadiusSim
2015-08-02 19:02:55.195	✔				11:11:11:11:11:11					RadiusSim

ステップ 10 eps_unquarantine スクリプトを実行します。

```
Johns-MacBook-Pro:bin jeppich$ ./eps_unquarantine.sh -a 192.168.1.23 -u SIM02 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=SIM02
group=EPS
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
14:24:07.282 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
14:24:10.852 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
MAC address (or <enter> to disconnect): 11:11:11:11:11:11
MAC address (or <enter> to disconnect):
```

ステップ 11 RADIUS シミュレータを使用して user1 を認証します。

```
authenticator=2cff72c97b6b1cbd6839a224ae566af0
Attributes=<
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:ise201/227903462/89
  vendorId=9 vsa=[cts:security-group-tag=0014-0,]
  vendorId=9 vsa=[profile-name=Add_Device,]
>

C:\sim>java -cp RadiusSimulator.jar -DUSERNAME=user1 -DPASSWORD=Aa123456 -DAUDIT_SESSION_ID=1001 -DACCT_SESSION_ID=2001 -DCALLING_STATION_ID=11:11:11:11:11:11 -DFRAMED_IP_ADDRESS=192.168.1.100 -DFRAMED_IP_MASK=255.255.255.0 RadiusAuthenticat
ion 192.168.1.23
AccessAccept code=2 id=1 length=109
authenticator=3ed59313ec8ceec6e349fbe6f23f444
Attributes=<
  UserName=user1
  State=ReauthSession:1001
  Class=CACS:1001:ise201/227903462/92
  vendorId=9 vsa=[profile-name=Add_Device,]
>

C:\sim>
```

ステップ 12 DynAuthListener が検疫イベントを受信します。

```

Received from /192.168.1.23:38085
DisconnectRequest code=40 id=2 length=104
authenticator=24151f8209cc58244112d2747aae92
Attributes=<
  NASIPAddress=192.168.1.37
  CallingStationID=11:11:11:11:11:11
  Unknown code=49 length=4
  EventTimestamp=Sun Aug 02 15:22:24 EDT 2015
  MessageAuthenticator=4cb295ea4fd8333c97bf9e21b04454
  vendorId=9 usa=[audit-session-id=1001,]
>
    
```

ステップ 13 [運用 (Operations)] > [RADIUS Livelog] を選択します。

Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles	Network Device
2015-08-02 19:24:01.804	i		0	user1	11:11:11:11:11:11	Add_Device	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	
2015-08-02 19:24:01.804	✓			user1	11:11:11:11:11:11	Add_Device	Default >> Default >> ...	Default >> Basic_Auth...	PermitAccess	RadiusSim
2015-08-02 19:22:24.856	✓				11:11:11:11:11:11					RadiusSim

802.1X を使用したサンプル スクリプトのテスト

Multigroupclient

検証

このテストは、サードパーティシステムが pxGrid の複数のクライアントグループ (Session、ANC) に登録 (認証および承認) できることを検証します。

定義

PxGrid クライアント登録では、サードパーティアプリケーション、セキュリティ デバイス、また、この場合は Linux ホストを、pxGrid コントローラの承認済みセッションまたは ANC グループに接続および登録します。admin や basic などの追加のグループを使用できますが、Admin グループは ISE 用に予約されており、pxGrid の管理承認が必要な Basic グループは、pxGrid のどの登録例でも使用されません。

すべての登録済み pxGrid クライアントは、[管理 (Administration)] の下の ISE pxGrid サービスで確認できます。

pxGrid クライアントは、ダイナミックピックでも示されるとおり、情報のパブリッシャまたはサブスクリバになることができます。ISE は情報を利用できず、コンテキストの共有は登録済みクライアント間で発生します。pxGrid クライアントが承認されたグループに対して正常に登録されると、pxGrid サンプル スクリプトによって決定されるとおり、クライアントは関連するセッション情報またはクエリを取得できます。

例

この例では、pxGrid コントローラに対するセッション グループに、Linux ホストを pxGrid クライアントとして登録します。Linux ホストである mac は、pxGrid クライアントのユーザ名です。また、ISE の登録済み pxGrid クライアントを表示します。

ステップ 1 multigroupclient スクリプトを実行します。

```
./multigroupclient.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -g Session -d pxGrid Client
```

使用法:

```
Usage: ./multigroupclient.sh [options]

Main options
-a <PXGRID_HOSTNAMES> (comma separated hostnames)
-u <PXGRID_USERNAME>
-g <PXGRID_GROUP>
-d <PXGRID_DESCRIPTION>

The followings are certificates options
-k <PXGRID_KEYSTORE_FILENAME>
-p <PXGRID_KEYSTORE_PASSWORD>
-t <PXGRID_TRUSTSTORE_FILENAME>
-q <PXGRID_TRUSTSTORE_PASSWORD>
```

If not specified, it defaults to use clientSample1.jks and rootSample.jks
Specifying values here can override the defaults

Custom config file can fill or override parameters

-c <config_filename>

Config file are being sourced.Use these variables:

- PXGRID_HOSTNAMES
- PXGRID_USERNAME
- PXGRID_GROUP
- PXGRID_DESCRIPTION
- PXGRID_KEYSTORE_FILENAME
- PXGRID_KEYSTORE_PASSWORD
- PXGRID_TRUSTSTORE_FILENAME
- PXGRID_TRUSTSTORE_PASSWORD

結果:

```

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session,ANC,Session
description=pxGrid
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
09:35:31.772 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
09:35:35.769 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Create ANC Policy: ANC1437658531354 Result -com.cisco.pxgrid.model.anc.ANCResult@612fc6eb[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=<null>
]
Session 1.1.1.2 not found
Connection closed
    
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。
pxGrid クライアント mac を、セッションクライアントグループに登録します。デフォルトで、pxGrid Adaptive Network Control (ANC) 軽減アクションに必要な ANC は、追加されています。

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
mac	pxGrid	Capabilities(0 Pub, 0 Sub)	Offline	ANC,Session	View

セッションのサブスクライブ

検証

このテストは、サードパーティシステムが登録可能になると、pxGrid 上で利用可能な情報のトピックにクライアントがサブスクライブできる pxGrid に接続されることを検証します。この場合、pxGrid クライアントはユーザ認証ステータスの更新をサブスクライブします。

定義

クライアントがセッションおよび ANC グループに対して pxGrid コントローラによって正常に登録および承認されると、クライアントは機能をサブスクライブして認証されたユーザ向けの関連するセッション情報を取得します。ISE MnT ノードは、ISE Session Directory を pxGrid コントローラに対してトピックとしてパブリッシュします。pxGrid クライアントはこの機能をサブスクライブし、認証されたユーザのアクティブなセッションまたは通知をリアルタイムで取得します。

例

pxGrid クライアントは、SessionDirectory 機能をサブスクライブし、リアルタイムで通知を受信します。

ステップ 1 session_subscribe スクリプトを実行します。

```
./session_subscribe.sh -a 10.0.0.37 -u mac_session -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac_session
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:00:10.800 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex. '1.0.0.0/255.0.0.0,1234::/16,...' or <enter> for no filter): 13:00:12.205 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。
pxGrid クライアントは SessionDirectory トピックをサブスクライブしています

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers License Warning

System Identity Management Network Resources Device Portal Management pxGrid Services Feed Service pxGrid Identity Mapping

Enable Auto-Regis

Clients Live Log

Enable Disable Approve Group Decline Delete Refresh Total Pending Approval(0) 1 - 9 of 9 Show

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
ise-sxp-ise238		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
mac_session		Capabilities(0 Pub, 2 Sub)	Online	Session	View

Capability Detail 1 - 2 of 2 Show 25 per page Page 1

Capability Name	Capability Version	Messaging Role	Message Filter
Core	1.0	Sub	
SessionDirectory	1.0	Sub	

ステップ 3 クライアント PC に対してログオフおよびログオンして、次の通知をリアルタイムで確認します。

```

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac_session
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
06:58:07.070 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex.'1.0.0.0/255.0.0.0,1234::/16,...' or <enter> for no filter): 06:58:08.835 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected

press <enter> to disconnect...session notification:
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F006EE7E0, User Name=host/jeppich-PC.lab6.com, AD
User DNS Domain=null, AD Host DNS Domain=lab6.com, AD User NetBIOS Name=null, AD Host NETBIOS Name=LAB6,
Calling station id=00:0C:29:79:02:A8, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint
Profile=Add Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-
Id=00000009], Posture Status=null, Posture Timestamp=, Session Last Update Time=Tue Jul 28 07:57:25 EDT
2015}

session notification:
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F006EE7E0, User Name=LAB6\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:29:79:02:A8, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint
Profile=Add Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-
Id=00000009], Posture Status=null, Posture Timestamp=, Session Last Update Time=Tue Jul 28 07:57:56 EDT
2015}

session notification:
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F006EE7E0, User Name=host/jeppich-PC.lab6.com, AD
User DNS Domain=null, AD Host DNS Domain=lab6.com, AD User NetBIOS Name=null, AD Host NETBIOS Name=LAB6,
Calling station id=00:0C:29:79:02:A8, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint
Profile=Add Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-
Id=00000009], Posture Status=null, Posture Timestamp=, Session Last Update Time=Tue Jul 28 07:59:17 EDT
2015}
    
```

セッションのダウンロード

検証

このテストは、アクティブなユーザ セッションの一括セッション ダウンロードを実行するためのサードパーティシステムの機能を検証します。

定義

セッション ダウンロード スクリプトが、パブリッシュされた ISE ノードから一括セッション レコードをダウンロードします。

例

この例では、pxGrid クライアントは、ISE MnT ノードからアクティブなセッションをダウンロードします。

ステップ 1 セッション ダウンロード スクリプトを実行します。

```
./session_download.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:30:38.687 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
Filters (ex.'1.0.0.0/255.0.0.0,1234::/16...' or <enter> for no filter): 12:30:40.056 [Thread-1] INFO
com.cisco.pxgrid.ReconnectionManager - Connected

Start time (ex.'2015-01-31 13:00:00' or <enter> for no start time):
End time (ex.'2015-01-31 13:00:00' or <enter> for no end time):
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F004BE344, User Name=jepich, 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:29:79:02:A8, Session state=AUTHENTICATED, ANCstatus=null, Security Group=null, Endpoint
Profile=Add Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-
Id=00000009], Posture Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 13:42:25 EDT
2015} Session={ip=[10.0.0.37], Audit Session Id=0A0000020000000E004156F4, User Name=00:0C:29:87:8D:1F, AD
User DNS
Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station
id=00:0C:29:87:8D:1F, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=VMWare-
Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/37, RADIUSAVPairs=[ Acct-Session-Id=00000005], Posture
Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 09:41:25 EDT 2015}
Session={ip=[10.0.0.3], Audit Session Id=0A0000020000000D00036A42, User Name=18:E7:28:2E:29:CB, AD User DNS
Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station
id=18:E7:28:2E:29:CB, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Cisco-
Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/37, RADIUSAVPairs=[ Acct-Session-Id=00000007], Posture
Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 09:43:42 EDT 2015}
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F004BE344, User Name=18:E7:28:2E:29:CC, AD User DNS
```

```
Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station
id=18:E7:28:2E:29:CC, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Cisco-
Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-Id=0000000A], Posture
Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 13:42:25 EDT 2015}
Session={ip=[10.0.0.33], Audit Session Id=0A0000020000000C0003610A, User Name=68:05:CA:12:7C:78, AD User DNS
Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station
id=68:05:CA:12:7C:78, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Unknown,
NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/23, RADIUSAVPairs=[ Acct-Session-Id=00000006], Posture
Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 09:43:42 EDT 2015}
Session count=5
Connection closed
```

IP によるセッションのクエリ

検証

このテストは、pxGrid を経由し、特定の IP アドレスに関連する指示されたクエリを実行するためのサードパーティシステムの能力を検証します。

定義

Session Query by IP スクリプトは、IP アドレスにより、認証されたユーザのセッション情報を取得します。

例

エンドユーザの IP アドレスを入力して、エンドユーザのセッション情報を取得します。

ステップ 1 session_query_by_ip スクリプトを実行します。

```
./session_query_by_ip.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:50:33.356 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:50:34.961 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
IP address (or <enter> to disconnect): 10.0.0.15
Session={ip=[10.0.0.15], Audit Session Id=0A0000020000000F004BE344, User Name=18:E7:28:2E:29:CC, AD User DNS
Domain=null, AD Host DNS Domain=null, AD User NetBIOS Name=null, AD Host NETBIOS Name=null, Calling station
id=18:E7:28:2E:29:CC, Session state=STARTED, ANCstatus=null, Security Group=null, Endpoint Profile=Cisco-
Device, NAS IP=10.0.0.2, NAS Port=GigabitEthernet1/0/43, RADIUSAVPairs=[ Acct-Session-Id=0000000A], Posture
Status=null, Posture Timestamp=, Session Last Update Time=Thu Jul 23 13:42:25 EDT 2015}
IP address (or <enter> to disconnect
```

EndpointProfile のサブスクリプション

検証

このテストは、パブリッシュされた Endpoint Profile トピックをサブスクリプションするためのサードパーティシステムの能力を検証します。

定義

登録済みの pxGrid クライアントは EndpointProfileMetadata 機能にサブスクリプションし、グローバル プロファイル ポリシーの変更または修正を取得します。セッション通知には、エンドポイント プロファイル ID、名前、および完全修飾名が含まれます。

例

この例では、ユーザの PC の静的 MAC アドレスに基づいて、pxGrid の EndpointProfile Example ポリシーが作成されます。pxGrid クライアントが EndpointprofileMetadata 機能をサブスクリプションし、ISE プロファイリング ポリシーに何らかの修正が発生したとき、リアルタイムで実行中の Linux スクリプトのセッション通知を確認します。

ステップ 1 endpointprofile_subscribe スクリプトを実行します。

```
./endpointprofile_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
10:14:02.627 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
10:14:04.268 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
mac		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

ステップ 3 [ポリシー (Policy)] > [プロファイリング (Profiling)] > [追加 (Add)] の順に選択します。
 ポリシーの名前と説明を入力します。

[If 条件 (If Condition)] > [新しい条件の作成 (Create New Condition)] > [IP] > {ネットワークにアクセスする IP アドレスを入力} および選択 > [送信 (Submit)] の順に選択します。

ステップ 4 先ほど作成し、追加されたプロファイリング ポリシー エンドポイント プロファイルのサブスクリプション 通知を受信します。

```
./endpointprofile_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
10:14:02.627 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
10:14:04.268 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...EndpointProfileChangedNotification (changetype=ADD) Device profile :
id=a5469840-3150-11e5-9b58-000c29878d1f, name=Add_Device, ffname=Add_Device
```

ID グループのダウンロード

検証

このテストは、ユーザ ID 情報の一括ダウンロードを実行するためのサードパーティシステムの機能を検証します。

定義

Identity Group ダウンロード スクリプトは、ユーザ グループ情報の一括セッションレコードとユーザ グループ マッピングをセッション ディレクトリからダウンロードします。これらのグループには、ISE ID グループおよびプロファイル済みグループが含まれています。

例

この例では、Identity Group ダウンロード スクリプトを使用して ISE MnT Node パブリッシャからすべてのグループ情報をダウンロードします。

ステップ 1 identity_group_download スクリプトを実行します。

```
./identity_group_download.sh -a 192.168.1.23 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
20:36:26.820 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
20:36:28.397 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
user=host/jeppich-PC.lab6.com groups=Workstation
```

```

user=LAB6\jpeppich groups=Workstation
user=user1 groups=User Identity Groups:Employee,Add_Device
user=user2 groups=User Identity Groups:Employee,Unknown
user=user3 groups=User Identity Groups:Employee
user=00:0C:29:79:02:A8 groups=Workstation
User count=6
Connection closed
20:36:30.882 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jpeppich$

```

結果

```

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
20:36:26.820 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
20:36:28.397 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
user=host/jpeppich-PC.lab6.com groups=Workstation
user=LAB6\jpeppich groups=Workstation
user=user1 groups=User Identity Groups:Employee,Add_Device
user=user2 groups=User Identity Groups:Employee,Unknown
user=user3 groups=User Identity Groups:Employee
user=00:0C:29:79:02:A8 groups=Workstation
User count=6
Connection closed
20:36:30.882 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jpeppich$

```

セキュリティグループのクエリ

検証

このテストは、ISE のすべてのセキュリティグループ タグを取得するためのサードパーティシステムの能力を検証します。

定義

セキュリティグループ クエリ スクリプトは、TrustSecMetadata 機能トピックを介して ISE で設定されたセキュリティグループ タグ (SGT) を公開します。これにより、一意の ID、セキュリティグループ タグの値、および記述に基づいて ISE で設定されたすべての SGT を取得するためのクエリ メソッドが提供されます。

例

この例では、セキュリティグループクエリ スクリプトがすべてのセキュリティグループ タグのコンテキスト情報をダウンロードします。このスクリプトは、ISE からすべての TrustSec セキュリティグループ セッション情報を取得します。これには、TrustSec タグの名前、固有識別子、記述、および値が含まれます。

セキュリティグループ タグについてのクエリを送信します。

ステップ 1 securitygroup_query スクリプトを実行します。

```
./securitygroup_query.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:53:11.474 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:53:12.897 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
SecurityGroup : id=65fddc70-2a34-11e5-82cb-005056bf2f0a, name=Unknown, desc=Unknown Security Group, tag=0
SecurityGroup : id=660aadb0-2a34-11e5-82cb-005056bf2f0a, name=ANY, desc=Any Security Group, tag=65535
SecurityGroup : id=669e6230-2a34-11e5-82cb-005056bf2f0a, name=SGT_Auditor, desc=Auditor Security Group,
tag=9
SecurityGroup : id=66bdd110-2a34-11e5-82cb-005056bf2f0a, name=SGT_BYOD, desc=BYOD Security Group, tag=15
SecurityGroup : id=66dd3ff0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Contractor, desc=Contractor Security
Group, tag=5
SecurityGroup : id=66fcd5e0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Developer, desc=Developer Security Group,
tag=8
SecurityGroup : id=671a21e0-2a34-11e5-82cb-005056bf2f0a, name=SGT_DevelopmentServers, desc=Development
Servers Security Group, tag=12
SecurityGroup : id=673c9e00-2a34-11e5-82cb-005056bf2f0a, name=SGT_Employee, desc=Employee Security Group,
tag=4
SecurityGroup : id=6759ea00-2a34-11e5-82cb-005056bf2f0a, name=SGT_Guest, desc=Guest Security Group, tag=6
SecurityGroup : id=6775d670-2a34-11e5-82cb-005056bf2f0a, name=SGT_NetworkServices, desc=Network Services
Security Group, tag=3
SecurityGroup : id=67959370-2a34-11e5-82cb-005056bf2f0a, name=SGT_PCIServers, desc=PCI Servers Security
Group, tag=14
SecurityGroup : id=67b3a2c0-2a34-11e5-82cb-005056bf2f0a, name=SGT_PointOfSale, desc=PointOfSale Security
Group, tag=10
SecurityGroup : id=67d50d70-2a34-11e5-82cb-005056bf2f0a, name=SGT_ProductionServers, desc=Production Servers
Security Group, tag=11
SecurityGroup : id=67f16f10-2a34-11e5-82cb-005056bf2f0a, name=SGT_ProductionUser, desc=Production User
Security Group, tag=7
SecurityGroup : id=680df7c0-2a34-11e5-82cb-005056bf2f0a, name=SGT_Quarantine, desc=Quarantine Security
Group, tag=255
SecurityGroup : id=682a5960-2a34-11e5-82cb-005056bf2f0a, name=SGT_TestServers, desc=Test Servers Security
Group, tag=13
SecurityGroup : id=68461ec0-2a34-11e5-82cb-005056bf2f0a, name=SGT_TrustSecDevices, desc=TrustSec Devices
Security Group, tag=2
Connection closed
11:53:13.235 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager- Stopped
```

セキュリティグループのサブスクリプション

検証

このテストは、pxGrid を介して SecurityGroup トピックをサブスクリプションするためのサードパーティシステムの能力を検証します。

定義

セキュリティグループ スクリプトは、TrustsecMetaDataCapability トピックを介して ISE で設定されたセキュリティグループ タグ (SGT) を公開します。セキュリティグループが追加/更新/削除されると、セキュリティグループの変更通知がスクリプト セッション通知に表示されます。

例

セキュリティグループ サブスクリプション スクリプトは、ISE TrustSec ポリシーの変更をサブスクリプションします。この例では、jsmith のセキュリティグループ タグ情報を含む .cvs ファイルを生成および作成します。この情報には、セキュリティ、タグ名、値、説明が入力されます。このファイルは、ISE にアップロードされます。このファイルがアップロードされると、Linux ホストで実行中の securitygroup_subscribe スクリプトに SecurityGroupChange 通知セッション通知が表示されます。これは pxGrid クライアントが TrustsecMetaDataCapability をサブスクリプションしたときに発生します。

ステップ 1 securitygroup_subscribe スクリプトを実行します。

```
./securitygroup_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:12:22.902 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス(pxGrid services)] の順に選択します。
pxGrid クライアントは TrustSecMetadata 機能をサブスクライブしています

The screenshot shows the Identity Services Engine Administration interface. The breadcrumb trail is: Home > Operations > Policy > Guest Access > Administration > pxGrid Services. The main content area shows a 'Clients' section with a 'Live Log' tab. There are action buttons: Enable, Disable, Approve, Group, Decline, Delete, and Refresh. A table lists clients:

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
mac		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

Below the table is a 'Capability Detail' section with a table:

Capability Name	Capability Version	Messaging Role	Message Filter
Core	1.0	Sub	
TrustSecMetaData	1.0	Sub	

ステップ 3 [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [セキュリティグループ (Security Groups)] リストの順に選択して、MAC_Group を追加します。

The screenshot shows the Identity Services Engine TrustSec Device Administration interface. The breadcrumb trail is: Home > Operations > Policy > Guest Access > Administration > Work Centers > TrustSec > Device Administration. The main content area shows the 'Security Groups List > MAC_Group' configuration page. The form includes:

- * Name: MAC_Group
- * Icon: A grid of icons with a computer monitor icon selected.
- Description: An empty text area.
- Security Group Tag (Dec / Hex): 16/0010
- Generation Id: 0
- Buttons: Save, Reset

ステップ 4 セキュリティグループの変更通知が以下のように反映されます。

```
./securitygroup_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:12:22.902 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:12:24.320 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...SecurityGroupChangeNotification (changetype=MODIFY) SecurityGroup :
id=af3c6ac0-315d-11e5-9b58-000c29878d1f, name=MAC_Group, desc=, tag=16
```

エンドポイントプロファイルのクエリ

検証

このテストは、ISE で設定された、すべての有効にされたプロファイルを取得するためのサードパーティシステムの能力を検証します。

定義

endpointprofile_query スクリプトは、ISE で設定された、すべての有効にされたエンドポイント プロファイルを取得するためのクエリメソッドを提供します。また、エンドポイント プロファイル ID、名前、および完全修飾名を提供します。また、サブスクリイバは、エンドポイント プロファイルが ISE で追加/更新/削除された場合にも通知されます。

例

この例では、endpointprofile スクリプトは ISE の有効にされたすべてのプロファイルを取得します。

ステップ 1 endpointprofile_query スクリプトを実行します。

```
./endpointprofile_query.sh -a 192.168.1.23 -u pxGrid02 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGrid02
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
```

```
-----
17:57:04.103 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
17:57:05.681 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Endpoint Profile : id=8c8f42b0-393f-11e5-ac86-000c297fb12a, name=Add Device, fqname Add Device
Endpoint Profile : id=4d852be0-2a33-11e5-82cb-005056bf2f0a, name=Android, fqname Android
Endpoint Profile : id=4dc7b320-2a33-11e5-82cb-005056bf2f0a, name=Apple-Device, fqname Apple-Device
Endpoint Profile : id=4e190770-2a33-11e5-82cb-005056bf2f0a, name=Apple-iDevice, fqname Apple-Device:Apple-iDevice
Endpoint Profile : id=4e452080-2a33-11e5-82cb-005056bf2f0a, name=Apple-iPad, fqname Apple-Device:Apple-iPad
```

機能

検証

このテストは、ISE でパブリッシュされたすべての機能を取得するためのサードパーティシステムの能力を検証します。

定義

機能スクリプトは、ISE でパブリッシュされたすべての関心のあるトピックを取得します。

例

機能スクリプトは、情報トピックまたはクライアントがパブリッシュまたはサブスクライブできる機能を取得します。

ステップ 1 capability_query スクリプトを実行します。

```
./capability_query.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=null
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
09:57:07.306 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
09:57:09.199 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
capability=SessionDirectory, version=1.0
capability=GridControllerAdminService, version=1.0
capability=EndpointProtectionService, version=1.0
capability=IdentityGroup, version=1.0
capability=EndpointProfileMetaData, version=1.0
capability=TrustSecMetaData, version=1.0
capability=AdaptiveNetworkControl, version=1.0
capability=Core, version=1.0
Connection closed
09:57:09.254 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
```

ID グループのクエリ

検証

このテストは、指定されたユーザから ISE ID グループ情報を取得するためのサードパーティシステムの能力を検証します。

定義

ID グループ クエリ スクリプトは、ISE ID グループ情報を取得します。

例

エンドユーザから取得したエンドユーザの ID グループ情報

ステップ 1 identity_group_query スクリプトを実行します。

```
./identity_group_query.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
10:58:54.937 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
10:58:56.869 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
user name (or <enter> to disconnect): jeppich
group=Profiled
```


ID グループのサブスクライブ

検証

このテストは、ISE によってパブリッシュされた ID トピックをサブスクライブするため、および通知を受信するためのサードパーティシステムの能力を検証します。

定義

ID グループ トピックをサブスクライブすると、pxGrid クライアントは、802.1X 以外のイベントについて通知を受信できません。

例

ISE に内部ネットワークのユーザが作成され、イベントをトリガーするゲスト ポータルをテストするために使用されます

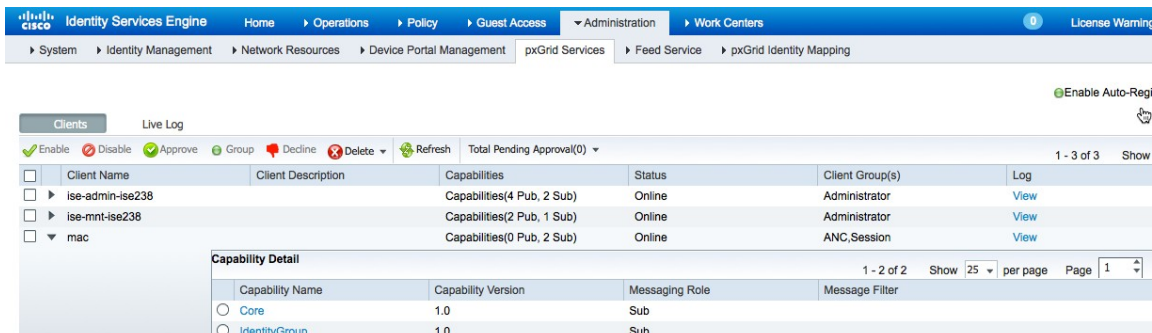
ステップ 1 identity_group_subscribe スクリプトを実行します。

```
/identity_group_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:20:22.839 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:20:24.468 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect...
```

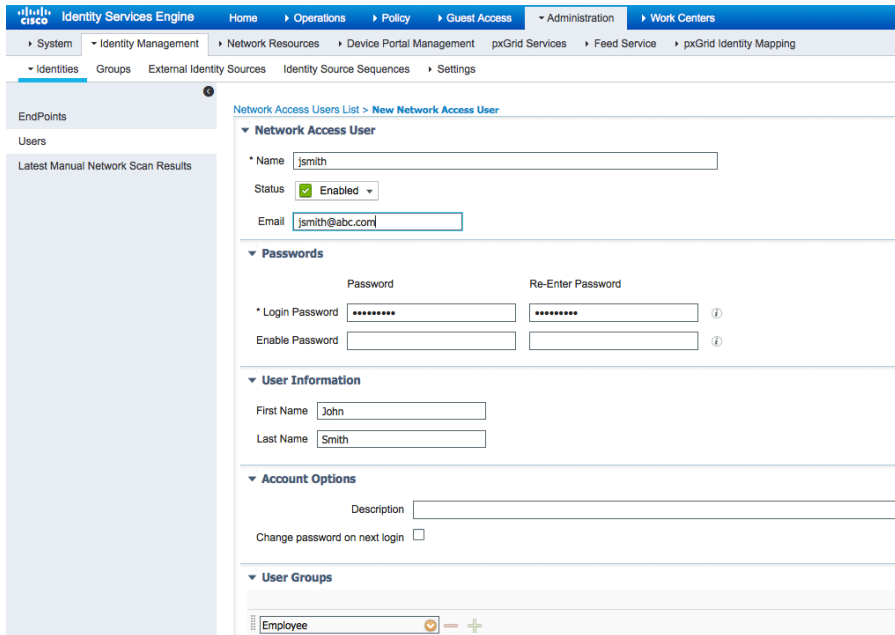
ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択して、サブスクライブされた ID グループ セッションを表示します。



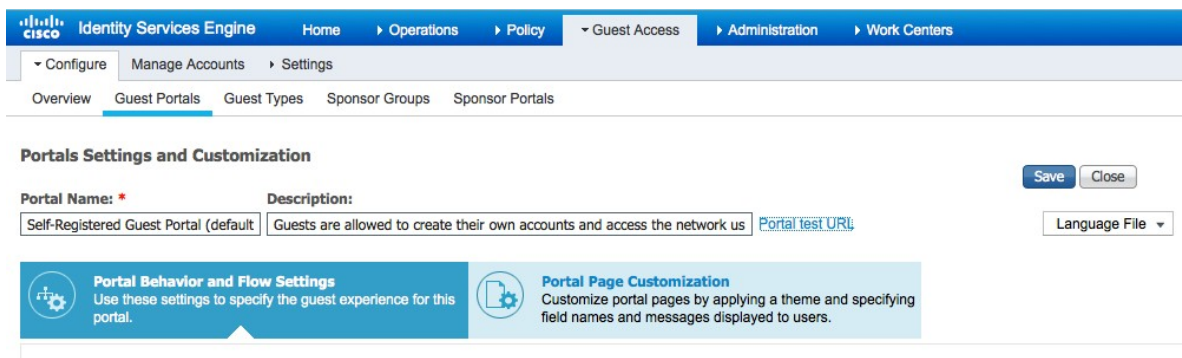
Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
mac		Capabilities(0 Pub, 2 Sub)	Online	ANC,Session	View

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

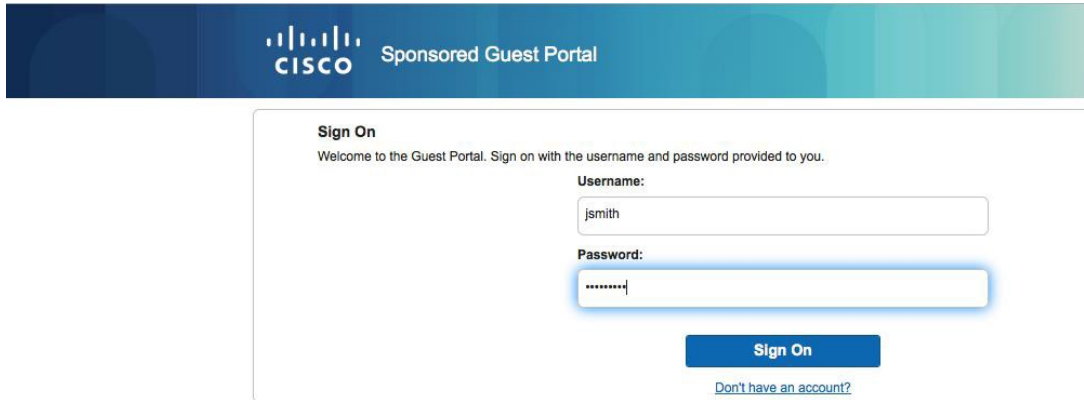
ステップ 3 ゲストポータルに使用して従業員をトリガーするために ISE ID ユーザを作成します。



ステップ 4 デフォルトのセルフ サービスポータルのテストを使用して、ユーザおよび関連付けられた ID グループをリアルタイムで検証するため、 [ゲストアクセス (Guest Access)] > [構成 (Configure)] > [ゲストポータル (Guest Portals)] > [ポータル (Portal)] の順に選択して、URL をテストします。



ステップ 5 [ポータル (Portal)] テストをクリックして、入力した ID グループのユーザ値を入力します。



ステップ 6 [サインオン(Sign On)] をクリックします。

ステップ 7 ID ユーザおよびグループ通知が表示されます。

```
./identity_group_subscribe.sh -a 10.0.0.37 -u mac -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
```

結果

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:20:22.839 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:20:24.468 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Press <enter> to disconnect... user=jsmith
group=Employee
```

Adaptive Network Control(ANC)ポリシー

Adaptive Network Control Policies (ANC) pxGrid 軽減ポリシーは、サードパーティアプリケーションまたはシスコのセキュリティソリューションに対して、検疫、修復、プロビジョニング、port_bounce、port_shutdown などのカスタマイズされたアクションを実施することで、企業のセキュリティポリシーのよりカスタマイズされたきめ細かな適用方法を提供します。エンドポイントの検疫を解除するには、clear コマンドを発行します。ANC ポリシーは、ISE 上で Session:ANCpolicy という承認条件規則とともに構成されます。MAC または IP アドレスによって、手動でエンドポイントの軽減アクションを適用することもできます。

ISE 2.0 には、ISE 1.3 または Adaptive Network Control (ANC) にあった、ANC による軽減を機能させるために ISE で有効化が必要な Endpoint Protection サービスがありません。この機能はデフォルトで有効になっています。

ANCAction_query スクリプトは、認証済みの 802.1X エンドユーザとともに実行されるため、読者は、ANC 軽減スクリプトの次の呼び出しが容易になります。

- 検疫によって認証された 802.1X エンドポイント
- エンドポイントの検疫解除 (clear)
- トリガーされた ANC ポリシーに基づくエンドポイントのリストの提供
- 修復およびプロビジョニング通知を受信するための、ANC 機能へのサブスクライブ

ANC 許可ポリシー

ANC 許可ポリシーは、ANC ポリシー条件規則の結果となるネットワークアクションです。

ステップ 1 ANC 許可を作成します。

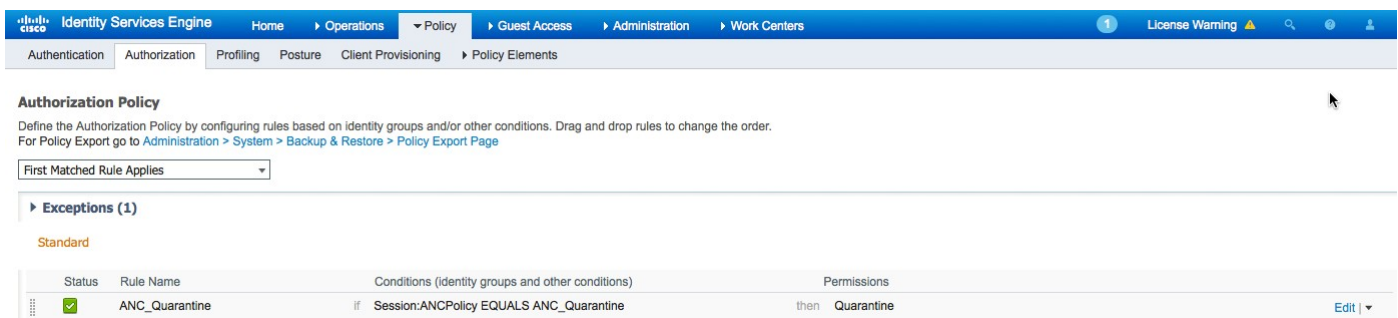
ステップ 2 [ポリシー (Policy)] > [認証 (Authorization)] の順に選択し、三角形の上でクリックして新しい規則を挿入します。

次の規則を追加します。

規則名: **ANC_Quarantine:**

新しい条件の作成: **Session:ANCpolicy:ANC_Quarantine**

セキュリティグループ: **Quarantine**



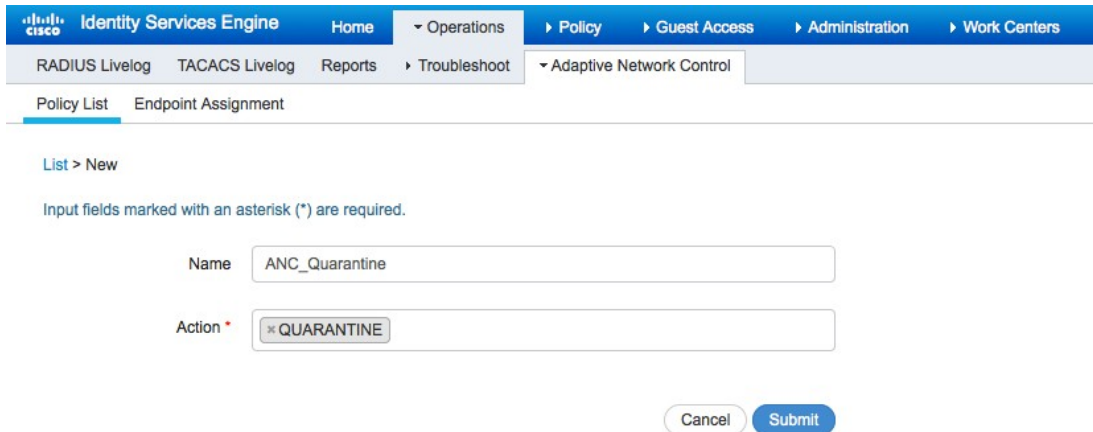
Status	Rule Name	Conditions (identity groups and other conditions)	Permissions
✓	ANC_Quarantine	if Session:ANCPolicy EQUALS ANC_Quarantine	then Quarantine

ステップ 3 [完了 (Done)] > [保存 (Save)] の順にクリックします。

ANC ポリシー: 検疫

ANC ポリシーは、実行される ANC pxGrid 検疫軽減アクションを定義します。

ステップ 1 [運用 (Operations)] > [Adaptive Network Control] > [ポリシー リスト (Policy List)] > [名前 (Name)] > [ANC_Quarantine] の順に選択します。



Identity Services Engine Home Operations Policy Guest Access Administration Work Centers

RADIUS Livelog TACACS Livelog Reports Troubleshoot Adaptive Network Control

Policy List Endpoint Assignment

List > New

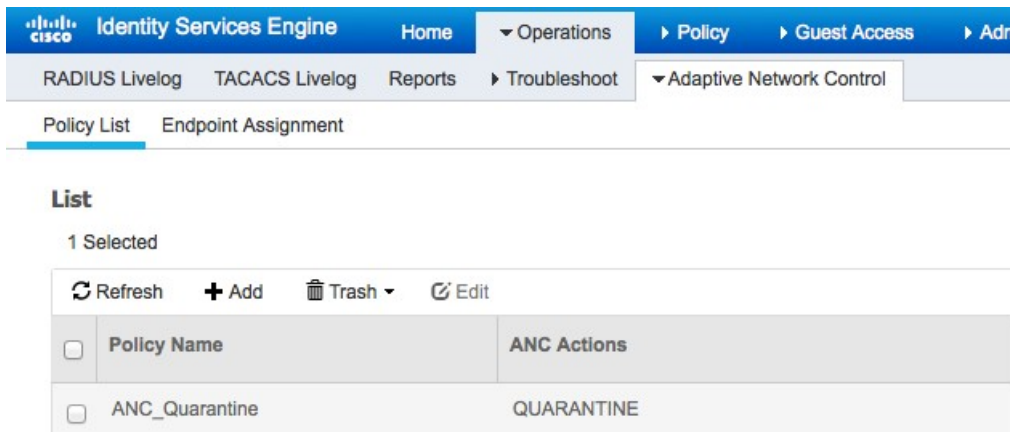
Input fields marked with an asterisk (*) are required.

Name: ANC_Quarantine

Action*: *QUARANTINE

Cancel Submit

ステップ 2 [送信 (Submit)] を選択します。
次が表示されます。



Identity Services Engine Home Operations Policy Guest Access Administration

RADIUS Livelog TACACS Livelog Reports Troubleshoot Adaptive Network Control

Policy List Endpoint Assignment

List

1 Selected

Refresh Add Trash Edit

<input type="checkbox"/>	Policy Name	ANC Actions
<input type="checkbox"/>	ANC_Quarantine	QUARANTINE

エンドポイントの表示/取得/ポリシー適用のための pxGrid ANC 検疫スクリプト

この例では、ANC クエリ スクリプトを実行して、ANC_Quarantine ポリシーを取得し、エンドポイントに適用します。

ステップ 1 ANCAction_query スクリプトを実行します。

```
./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
```

```
truststorePassword=cisco123
-----
21:27:57.849 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
21:28:00.252 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect):
```

ステップ 2 10 を選択し、ポリシー名を入力します。

```
Enter number (or <enter> to disconnect): 10
Policy name (or <enter> to disconnect): ANC_Quarantine
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@11758f2a[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=[com.cisco.pxgrid.model.anc.ANCPolicy@74ad1f1f[
    name=ANC_Quarantine
    actions=[QUARANTINE]
  ]]
]
```

ステップ 3 14 を選択し、ポリシー名を入力します。

```
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 14
Policy name (or <enter> to disconnect): ANC_Quarantine
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@66d1af89[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=[com.cisco.pxgrid.model.anc.ANCEndpoint@8646db9[
    policyName=ANC_Quarantine
    macAddress=00:0C:29:79:02:A8
    ipAddress=<null>
  ]]
]
```

ステップ 4 3 を選択し、ポリシー名を入力します。

```

Operation selection:
1. ApplyEndpointPolicyByMAC
2. ClearEndpointPolicyByMAC
3. ApplyEndpointPolicyByIP
4. ClearEndpointPolicyByIP
5. GetEndpointByIP
6. Subscribe
7. CreatePolicy
8. UpdatePolicy
9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 3
Policy name (or <enter> to disconnect): ANC_Quarantine
IP address (or <enter> to disconnect): 192.168.1.38
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@462d5aee[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=<null>
]
    
```

ステップ 5 [運用(Operations)] > [RADIUS Livelog] の順に選択します。認証された IP アドレスは検疫されていることに注意してください

Time	Status	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles
2015-08-03 02:40:22.644	All	0	LAB6\jeppich	00:0C:29:79:02:A8	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> ANC_Quarantine	Quarantine
2015-08-03 02:40:22.549	✓		#CTSREQUEST#					
2015-08-03 02:40:22.530	✓		LAB6\jeppich	00:0C:29:79:02:A8	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> ANC_Quarantine	Quarantine
2015-08-03 02:40:22.128	✓			00:0C:29:79:02:A8				

ステップ 6 検疫を解除する (clear を実行する) には、4 を選択し、MAC アドレスを入力します。

```

Operation selection:
1. ApplyEndpointPolicyByMAC
2. ClearEndpointPolicyByMAC
3. ApplyEndpointPolicyByIP
4. ClearEndpointPolicyByIP
5. GetEndpointByIP
6. Subscribe
7. CreatePolicy
8. UpdatePolicy
9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 2
MAC address (or <enter> to disconnect): 00:0C:29:79:02:A8
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@11758f2a[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancPolicies=<null>
    
```

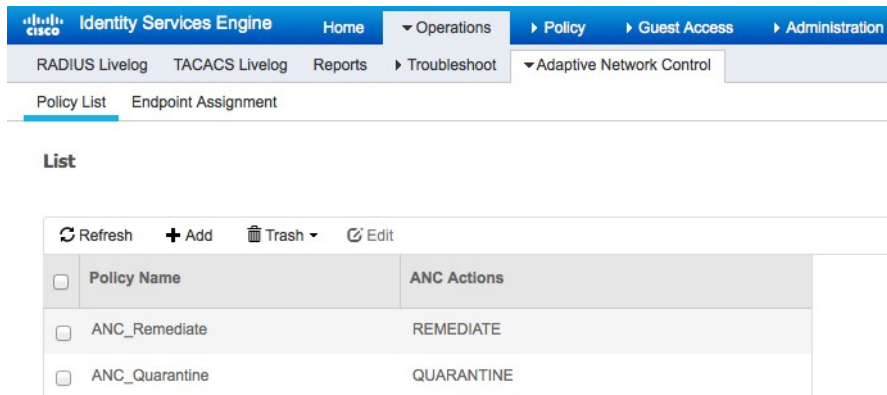
ステップ 7 [運用 (Operations)] > [RADIUS Livelog] を選択します。
 エンドユーザは、検疫が解除されています

Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles	Network Device
2015-08-03 03:23:53.087			0	LAB6\jpeppich	00:0C:29:79:02:A8	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> Compliant	Compliant	
2015-08-03 03:23:52.766				#ACSACL#-IP-PEI						Switch
2015-08-03 03:23:52.734				LAB6\jpeppich	00:0C:29:79:02:A8	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> Compliant	Compliant	Switch
2015-08-03 03:23:52.603					00:0C:29:79:02:A8					Switch
2015-08-03 03:23:24.526				#CTSREQUEST#						Switch
2015-08-03 03:23:24.432				#ACSACL#-IP-Po:						Switch
2015-08-03 03:23:24.412				LAB6\jpeppich	00:0C:29:79:02:A8	Windows7-Worksta...	Default >> Dot1X >> D..	Default >> Posture	Posture,SGT_Employee	Switch

ANC の修復

ANC の修復の軽減アクションは、サブスクリイバに修復アクションを提供します。

ステップ 1 [運用 (Operations)] > [Adaptive Network Control]、[ANC_Remediate] の順に選択してから、[REMEDiate (修復)] アクションを選択します。



Policy Name	ANC Actions
ANC_Remediate	REMEDiate
ANC_Quarantine	QUARANTINE

ステップ 2 ANCQuery スクリプトを実行し、6(サブスクリイブ)を選択します。

```
Johns-MacBook-Pro:bin jeppich$ ./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123
-t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:42:49.269 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:42:52.131 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 6
Press <enter> to disconnect:
```

ステップ 3 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択すると、pxGrid クライアントは、ANC グループに接続します。

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise201		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise201		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
pxgridcremediate		Capabilities(0 Pub, 2 Sub)	Online	ANC	View
pxgridclient		Capabilities(0 Pub, 2 Sub)	Online	ANC, EPS	View

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

ステップ 4 別のシェルを開き、次のスクリプトを実行します。

```

./ANCAction_query.sh -a 192.168.1.23 -u pxGridCRemediate -k alpha.jks -p cisco123 -t alpha_root.jks -g
cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridCRemediate
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:49:35.734 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:49:37.043 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 3
Policy name (or <enter> to disconnect): ANC_Remediate
IP address (or <enter> to disconnect): 192.168.1.41
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@11758f2a[
 ancStatus=SUCCESS
 ancFailure=<null>
 failureDescription=<null>
 ancEndpoints=<null>
 ancpolicies=<null>
]
    
```

```
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect):
```

ステップ 5 元のサブスクリプト スクリプトに通知が表示されます。

```
./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:48:17.245 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:48:18.563 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 6
Press <enter> to disconnect:
Apply Endpoint Policy Notification:
Policy=ANC_Remediate IP Address=192.168.1.41
```

ANC のプロビジョニング

ANC のプロビジョニングの軽減アクションは、サブスクリイバに修復アクションを提供します。

ステップ 1 ANCAction クエリ スクリプトを実行し、**6**(サブスクリイブ)を選択します。

```
Johns-MacBook-Pro:bin jeppich$ ./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123
-t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:42:49.269 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:42:52.131 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 6
Press <enter> to disconnect:
```

ステップ 2 clear または検疫解除を実行するには、エンドポイントに ANC プロビジョニング ポリシーを適用します。

```
12:03:43.784 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 4
IP address (or <enter> to disconnect): 192.168.1.41
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@111758f2a[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=<null>
]
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 3
Policy name (or <enter> to disconnect): ANC Provisioning
IP address (or <enter> to disconnect): 192.168.1.41
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@74ad1f1f[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=<null>
  ancpolicies=<null>
]
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect):
```

ステップ 3 サブスクライバは ANC Provisioning ポリシー通知を受信します。

```
./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:04:19.804 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:04:21.292 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 6
Press <enter> to disconnect:
Apply Endpoint Policy Notification:
Policy=ANC_Provisioning IP Address=192.168.1.41
```

ANC ポリシーに従うエンドポイントのリスト

この例では、適用される ANC のポリシーがあるエンドポイントのリストについて説明します。たとえば、エンドポイントのリストに対して ANC 検疫ポリシーを適用できます。

ステップ 1 ANC_Action クエリ スクリプトを実行し、**14** を選択してポリシー名 (**ANC_Provisioning**) を選択します。ANC_Provisioning ポリシーが適用された MAC アドレスのリストが表示されます。

```
./ANCAction_query.sh -a 192.168.1.23 -u pxGridClient -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=pxGridClient
group=ANC
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
13:32:53.702 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
13:32:54.973 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
```

```
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect): 14
Policy name (or <enter> to disconnect): ANC_Provisioning
ANCResult=com.cisco.pxgrid.model.anc.ANCResult@11758f2a[
  ancStatus=SUCCESS
  ancFailure=<null>
  failureDescription=<null>
  ancEndpoints=[com.cisco.pxgrid.model.anc.ANCEndpoint@74ad1f1f[
    policyName=ANC_Provisioning
    macAddress=00:0C:29:79:02:A8
    ipAddress=<null>
  ]]
  ancpolicies=<null>
]
Operation selection:
 1. ApplyEndpointPolicyByMAC
 2. ClearEndpointPolicyByMAC
 3. ApplyEndpointPolicyByIP
 4. ClearEndpointPolicyByIP
 5. GetEndpointByIP
 6. Subscribe
 7. CreatePolicy
 8. UpdatePolicy
 9. DeletePolicy
10. GetPolicyByName
11. GetAllPolicies
12. GetEndPointByMAC
13. GetAllEndpoints
14. GetEndpointByPolicy
Enter number (or <enter> to disconnect):
```

ダイナミックトピック

ダイナミックトピックを使用すると、ISE pxGrid ノードに接続されている pxGrid クライアントが情報トピックに関するパブリッシュ、サブスクライブ、アクションを行うことができます。ダイナミックトピックは、次のもので構成されます。

- トピック セットアップ

トピック、クエリ項目、およびアクション項目は、「propose_capability.sh」を使用して定義されます

- トピックのパブリッシュ

パブリッシャは、「generic_client -c publisher.properties」によって定義されます。ここで、パブリッシャのプロパティはコンフィギュレーション ファイルであり、トピック名、パブリッシャのクライアント モード、およびその他の項目など、トピック情報を記述したものです。

- トピックへのサブスクライブ

サブスクライバは、「generic_client -c subscriber.properties」によって定義されます。ここで、サブスクライバのプロパティはコンフィギュレーション ファイルであり、トピック名やその他の項目、サブスクライバのクライアント モードおよびクエリやアクション名のセット、その他の項目など、トピック情報を記述したものです。読み取り専用クエリ名セットは、特定のアクセストピック情報をサブスクライバに提供します。

アクション項目は、情報トピックをサブスクライブしないでトピックにクエリを発行したいサブスクライバ用のものです。

この例では、パブリッシュされたトピックまたは機能は Auction およびオークション サービスです。sdk-01-pub pxGrid クライアントは Auction トピックをパブリッシュし、sdk-01-sub pxGrid クライアントがこのトピックにサブスクライブすると、「get inventory services」および「get current bids」についてクエリを実行できます。別の pxGrid クライアント sdk-01-act は、このトピックにサブスクライブせず、通知も一切受信しませんが、このクライアントは、「bid on items」(アクションを実行すること)だけが可能です。

コアのサブスクライブ

pxGrid クライアントが「core」トピックにサブスクライブしているとき、機能のトピック通知のリストを提供します。

ステップ 1 次の内容を実行します。

```
./core_subscribe.sh -a 10.0.0.37 -u core_user-01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -g Session -d pxGrid Client
```

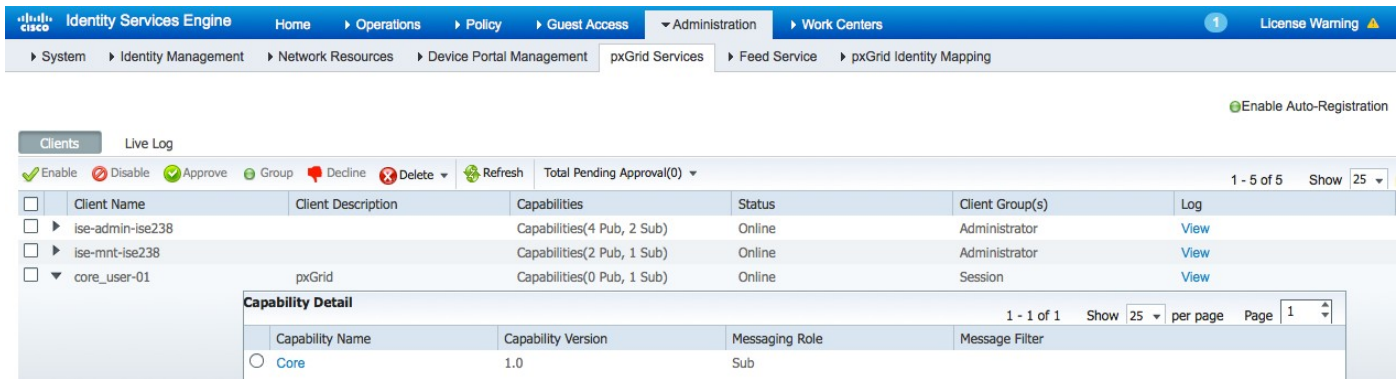
利用できる機能または情報のトピックのリストを取得します。

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=core_user-01
group=Session
description=pxGrid
keystoreFilename=alpha.jks
```



```
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:38:47.850 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:38:50.611 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
getList: status=CREATED capability=TrustSecMetaData, version=1.0
getList: status=CREATED capability=EndpointProfileMetaData, version=1.0
getList: status=CREATED capability=IdentityGroup, version=1.0
getList: status=CREATED capability=GridControllerAdminService, version=1.0
getList: status=CREATED capability=SessionDirectory, version=1.0
getList: status=CREATED capability=AdaptiveNetworkControl, version=1.0
getList: status=CREATED capability=EndpointProtectionService, version=1.0
getList: status=CREATED capability=Core, version=1.0
Capability name [, version] to query (or <enter> to quit) :
```

ステップ 2 pxGrid クライアントがコア機能をサブスクライブしていることを確認します。
[管理 (Administration)] > [pxGrid サービス (pxGrid Services)] を選択します。



Propose_New 機能

新しいトピック情報を pxGrid ノードに対して定義します。または、機能名、バージョン、説明、プラットフォーム、クエリ、およびアクション項目などを入力することで、既存のトピックを修正できます。このトピックは、pxGrid の管理者がトピックを承認するまで保留状態のままになります。

ステップ 1 次の内容を実行します。

```
./propose_capability.sh -a 10.0.0.37 -u sdk01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -g -d
pxGrid New Publisher
```

情報の入力を求められた場合、機能の情報が必要になります。

```
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=sdk01
group=Basic
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
```

```
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:02:07.373 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:02:08.779 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
New capability?(y/n): y
Enter capability name: Auction
Enter capability version: 1.0
Enter capability description: Auction Service
Enter vendor platform: ABC Auction Service
Enter query name (<enter> to continue): GetInventoryItems
Enter query name (<enter> to continue): GetCurrentBids
Enter query name (<enter> to continue):
Enter action name (<enter> to continue): BidOnItems
Enter action name (<enter> to continue):
Proposing new capability...
Press <enter> to disconnect...
Connection closed
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] > [機能ごとに表示 (View by Capabilities)] の順に選択します。
「保留状態」の「Auction」機能が表示されます。

Capability Name	Capability Description	Vendor Platform	Capability Vers...	Status	Publisher Count	Subscriber Co...	Supported Filter Type	Created By
GridControllerAdminService			1.0	Enabled	0	1	N/A	
AdaptiveNetworkControl			1.0	Enabled	1	0	N/A	
Auction	Auction Service	ABC Auction Service	1.0	Pending create	0	0	N/A	sdk01@xgrid.cisco.com
Core			1.0	Enabled	0	4	N/A	
EndpointProfileMetaData			1.0	Enabled	1	0	N/A	

ステップ 3 トピックを選択して[承認 (Approve)] を選択します。

ステップ 4 pxGrid 管理者がトピックを承認します。

Capability Name	Capability Description	Vendor Platform	Capability Vers...	Status	Publisher Count	Subscriber Co...	Supported Filter Type	Created By
GridControllerAdminService			1.0	Enabled	0	1	N/A	
AdaptiveNetworkControl			1.0	Enabled	1	0	N/A	
Auction	Auction Service	ABC Auction Service	1.0	Pending create	0	0	N/A	sdk01@xgrid.cisco.com
Core			1.0	Enabled	0	4	N/A	
EndpointProfileMetaData			1.0	Enabled	1	0	N/A	

ステップ 5 「Auction」トピックが正常に作成されます。

Capability Name	Capability Description	Vendor Platform	Capability Vers...	Status	Publisher Count	Subscriber Co...	Supported Filter Type	Created By
GridControllerAdminService			1.0	Enabled	0	1	N/A	
AdaptiveNetworkControl			1.0	Enabled	1	0	N/A	
Auction	Auction Service	ABC Auction Service	1.0	Enabled	0	0	N/A	sdk01@xgrid.cisco.com
Core			1.0	Enabled	0	4	N/A	
EndpointProfileMetaData			1.0	Enabled	1	0	N/A	

ステップ 6 以下で強調表示されているように、pxGrid クライアントが「core_subscribed」になっている場合、新しいトピック通知が表示されます。

```

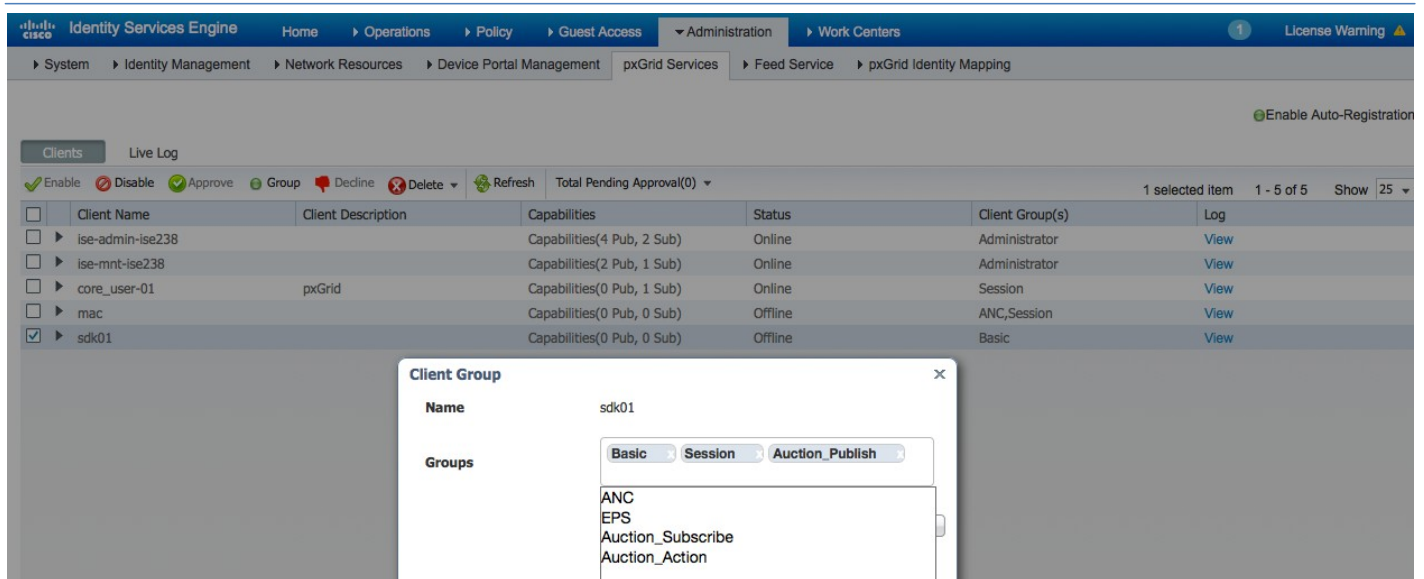
/core_subscribe.sh -a 10.0.0.37 -u core_user-01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -g
Session -d pxGrid Client
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=core_user-01
group=Session
description=pxGrid
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
11:48:41.155 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
11:48:42.946 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
getList: status=CREATED capability=TrustSecMetaData, version=1.0
getList: status=CREATED capability=EndpointProfileMetaData, version=1.0
getList: status=CREATED capability=IdentityGroup, version=1.0
getList: status=CREATED capability=GridControllerAdminService, version=1.0
getList: status=CREATED capability=SessionDirectory, version=1.0
getList: status=CREATED capability=AdaptiveNetworkControl, version=1.0
getList: status=CREATED capability=EndpointProtectionService, version=1.0
getList: status=CREATED capability=Core, version=1.0
Capability name [, version] to query (or <enter> to quit) : notification: status=CREATED capability=Auction,
version=1.0
    
```

ステップ 7 [ライブ ログ (Live Log)] を選択して、Auction トピックのセットアップの記録を確認します。

Client Name	Capability Name	Event Type	Timestamp	Other Attributes
sdk01@xgrid.cisco.com		Client offline	5:21:26 PM UTC, Jul 24 2015	
sdk01@xgrid.cisco.com	Core-1.0	Client unsubscribed	5:21:26 PM UTC, Jul 24 2015	
sdk01@xgrid.cisco.com	Auction-1.0	Topic create completed	5:21:25 PM UTC, Jul 24 2015	
sdk01@xgrid.cisco.com	Auction-1.0	Group created	5:21:25 PM UTC, Jul 24 2015	group Auction_Action
sdk01@xgrid.cisco.com	Auction-1.0	Group created	5:21:25 PM UTC, Jul 24 2015	group Auction_Subscribe
sdk01@xgrid.cisco.com	Auction-1.0	Group created	5:21:25 PM UTC, Jul 24 2015	group Auction_Publish
sdk01@xgrid.cisco.com	Auction-1.0	Topic create pending	5:01:59 PM UTC, Jul 24 2015	

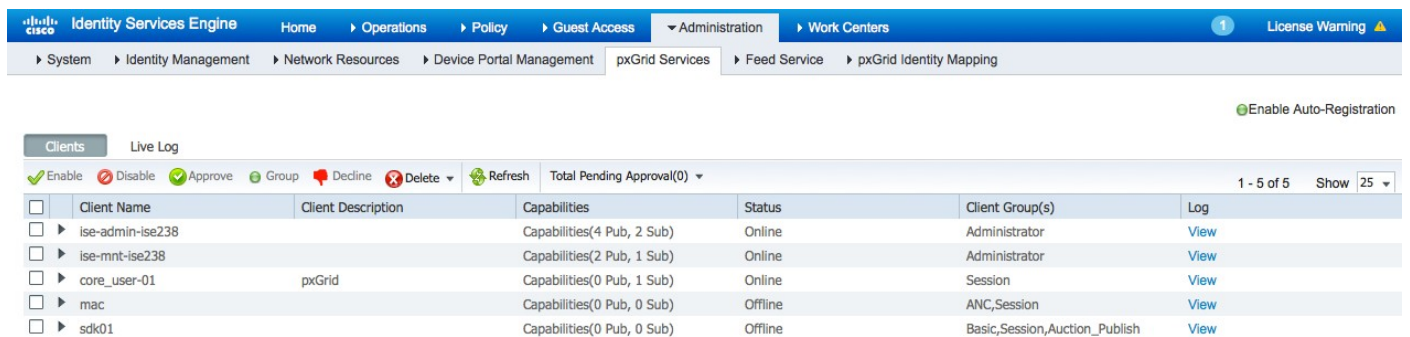
ステップ 8 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] > [sdk01] > [グループ (Group)] > [ベーシック、セッション、アクションのパブリッシュ (Basic, Session, Action Publish)] > [保存 (Save)] の順に選択します。

(注) 管理者は、トピックを「Basic」グループから他のグループに割り当てる必要があります。「Basic」グループは、pxGrid の接続のみのグループです。



The screenshot shows the Cisco Identity Services Engine (ISE) Administration console. The breadcrumb navigation is: Home > Operations > Policy > Guest Access > Administration > Work Centers > pxGrid Services > Feed Service > pxGrid Identity Mapping. The 'Clients' tab is active, showing a table of clients. The client 'sdk01' is selected, and a 'Client Group' dialog box is open. The dialog shows the 'Name' as 'sdk01' and the 'Groups' dropdown menu is open, displaying the following options: Basic, Session, Auction_Publish, ANC, EPS, Auction_Subscribe, and Auction_Action. The 'Basic' group is currently selected.

ステップ 9 [sdk01] の横の [表示 (View)] をクリックします。
パブリッシュされた Auction トピックが表示されます。



The screenshot shows the Cisco Identity Services Engine (ISE) Administration console. The breadcrumb navigation is: Home > Operations > Policy > Guest Access > Administration > Work Centers > pxGrid Services > Feed Service > pxGrid Identity Mapping. The 'Clients' tab is active, showing a table of clients. The client 'sdk01' is selected, and the 'Log' link is visible in the 'Log' column.

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
core_user-01	pxGrid	Capabilities(0 Pub, 1 Sub)	Online	Session	View
mac		Capabilities(0 Pub, 0 Sub)	Offline	ANC,Session	View
sdk01		Capabilities(0 Pub, 0 Sub)	Offline	Basic,Session,Auction_Publish	View

ステップ 10 イベントをパブリッシュするパブリッシャを決定する必要があります。publisher.conf ファイルを編集します。

```

conf — vim — 80x24
GENERIC_TOPIC_NAME="One"
GENERIC_CLIENT_MODE="publisher"
GENERIC_QUERY_NAME_SET=""
GENERIC_ACTION_NAME_SET=""
GENERIC_PUBLISH_DATA_SET="pub-notif-001, pub-notif-002, pub-notif-003"
GENERIC_REQUEST_DATA_SET=""
GENERIC_RESPONSE_DATA_SET="resp-001, resp-002, resp-003, resp-004"
GENERIC_SLEEP_INTERVAL="500"
GENERIC_ITERATIONS="20"
~
~
~
~
~
~
~
~
~
~
~
"generic_publisher.properties" 9L, 324C
    
```

ステップ 11 GENERIC_TOPIC_NAME=”AUCTION”とGENERIC_CLIENT_MODE=“PUBLISHER”に変更すると、データセットと応答データセットがパブリッシュされます。

```

GENERIC_TOPIC_NAME="Auction"
GENERIC_CLIENT_MODE="publisher"
GENERIC_QUERY_NAME_SET=""
GENERIC_ACTION_NAME_SET=""
GENERIC_PUBLISH_DATA_SET="pub-notif-001, pub-notif-002, pub-notif-003"
GENERIC_REQUEST_DATA_SET=""
GENERIC_RESPONSE_DATA_SET=" resp-001, resp-002, resp-003, resp-004"
GENERIC_SLEEP_INTERVAL="2000"
GENERIC_ITERATIONS="20"
~
~
~
~
~
~
~
~
~
~
~
"generic_publisher.properties" 9L, 329C
    
```

ステップ 12 パブリッシャ向けの汎用クライアント スクリプトを実行します。

```

./generic_client.sh -a 10.0.0.37 -u sdk01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -c
generic_publisher.properties
    
```

結果

```

Initialized : GenericClient:
  topicName=Auction
  clientMode=PUBLISHER
  sleepInterval=2000
  iterations=20
  queryNameSet=[]
  actionNameSet=[]
  publishDataSet=[pub-notif-001, pub-notif-002, pub-notif-003]
  requestDataSet=[]
  responseDataSet=[resp-001, resp-002, resp-003, resp-004]
----- properties -----
version=1.0.2-30-SNAPSHOT
    
```

```
hostnames=10.0.0.37
username=sdk01
group=Auction_Publish
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
14:12:59.548 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
14:13:00.921 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847981189]pub-notif-001
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847983193]pub-notif-002
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847985194]pub-notif-003
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847987195]pub-notif-001
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847989196]pub-notif-002
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
  capabilityName=Auction
  operationName=sampleNotification
  body:
    content:
      contentTags=[NOTIF-TAG-201]
      contentType=PLAIN_TEXT
      value=NOTIFICATION[1437847991197]pub-notif-003
Publishing notification: GenericMessage:
  messageType=NOTIFICATION
```

```
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437847993199]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437847995200]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437847997201]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437847999202]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437848001203]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437848003207]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437848005209]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
```

```
value=NOTIFICATION[1437848007210]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848009211]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848011213]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848013214]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848015216]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848017217]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
content:
contentTags=[NOTIF-TAG-201]
contentType=PLAIN_TEXT
value=NOTIFICATION[1437848019218]pub-notif-002
Press <enter> to disconnect...
```


ステップ 13 pxGrid クライアント sdk01 が Auction トピックをパブリッシュします。

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
▼ sdk01		Capabilities(1 Pub, 1 Sub)	Online	Basic,Session,Auction_Publish	View

Capability Detail				
Capability Name	Capability Version	Messaging Role	Message Filter	
<input type="radio"/> Auction	1.0	Pub		
<input type="radio"/> Core	1.0	Sub		

ステップ 14 サブスクライバがパブリッシュされた Auction トピックに対して、直接のクエリ「GetInventoryItems」、 「GetCurrentBids」でクエリを行うように構成する必要があります

```

GENERIC_TOPIC_NAME="Auction"
GENERIC_CLIENT_MODE="subscriber"
GENERIC_QUERY_NAME_SET="GetInventoryItems,GetCurrentBids,BidOnItems"
GENERIC_ACTION_NAME_SET=""
GENERIC_PUBLISH_DATA_SET=""
GENERIC_REQUEST_DATA_SET="req-001, req-002, req-003"
GENERIC_RESPONSE_DATA_SET=""
GENERIC_SLEEP_INTERVAL="500"
GENERIC_ITERATIONS="20"
~
~
    
```

ステップ 15 サブスクライバ用の汎用クライアント スクリプトを実行します。サブスクライバは、クエリトピックの GetInventoryItems、 GetCurrentBid に対してアクセス権があり、 BidOnItems にはないことに注意してください。 BidOnItems は、 Query トピックとして定義されていません。

```

./generic_client.sh -a 10.0.0.37 -u sdk01-sub -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -c
    
```

結果

```

Initialized : GenericClient:
  topicName=Auction
  clientMode=SUBSCRIBER
  sleepInterval=500
  iterations=20
  queryNameSet=[GetInventoryItems, GetCurrentBids, BidOnItems]
  actionNameSet=[]
  publishDataSet=[]
  requestDataSet=[req-001, req-002, req-003]
  responseDataSet=[]

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=sdk01-sub
group=Auction_Subscribe
description=null
    
```

```
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
15:51:33.423 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
15:51:36.123 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
Sending request: GenericMessage:
  messageType=REQUEST
  capabilityName=Auction
  operationName=GetInventoryItems
  body:
    content:
      contentTags=[QUERY-TAG-301]
      contentType=PLAIN_TEXT
      value=QUERY[1437853896264]req-001
Received response: GenericMessage:
  messageType=RESPONSE
  capabilityName=Auction
  operationName=GetInventoryItems
  body:
    content:
      contentTags=[RESP-TAG-101]
      contentType=PLAIN_TEXT
      value=RESPONSE[1437853896285]resp-003 - for request[QUERY[1437853896264]req-001]
Sending request: GenericMessage:
  messageType=REQUEST
  capabilityName=Auction
  operationName=GetCurrentBids
  body:
    content:
      contentTags=[QUERY-TAG-301]
      contentType=PLAIN_TEXT
      value=QUERY[1437853896885]req-002
Received response: GenericMessage:
  messageType=RESPONSE
  capabilityName=Auction
  operationName=GetCurrentBids
  body:
    content:
      contentTags=[RESP-TAG-101]
      contentType=PLAIN_TEXT
      value=RESPONSE[1437853896945]resp-004 - for request[QUERY[1437853896885]req-002]
Sending request: GenericMessage:
  messageType=REQUEST
  capabilityName=Auction
  operationName=BidOnItems
  body:
    content:
      contentTags=[QUERY-TAG-301]
      contentType=PLAIN_TEXT
      value=QUERY[1437853897457]req-003
Received response: GenericMessage:
  messageType=RESPONSE
  capabilityName=null
  operationName=null
  body:
    error=not authorized
Sending request: GenericMessage:
  messageType=REQUEST
  capabilityName=Auction
  operationName=GetInventoryItems
  body:
    content:
      contentTags=[QUERY-TAG-301]
      contentType=PLAIN_TEXT
      value=QUERY[1437853898077]req-001
Received response: GenericMessage:
  messageType=RESPONSE
```

```
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853898428]resp-001 - for request[QUERY[1437853898077]req-001]
```

ステップ 16 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。
pxGrid クライアント sdk01-sub は、Auction トピックにサブスクライブしています

The screenshot shows the Identity Services Engine Administration console. The breadcrumb navigation is: System > Identity Management > Network Resources > Device Portal Management > pxGrid Services > Feed Service > pxGrid Identity Mapping. The 'Clients' tab is active, showing a table of clients. The client 'sdk01-sub' is expanded to show its 'Capability Detail'.

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
▼ sdk01-sub		Capabilities(0 Pub, 2 Sub)	Online	Auction_Subscribe	View

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

要約

ステップ 1 Publisher (sdk01) は Auction トピックをパブリッシュします。

```
./generic_client.sh -a 10.0.0.37 -u sdk01 -k alpha.jks -p cisco123 -t alpha_root.jks -q cisco123 -c
generic_publisher.properties
Initialized : GenericClient:
  topicName=Auction
  clientMode=PUBLISHER
  sleepInterval=2000
  iterations=20
  queryNameSet=[]
  actionNameSet=[]
  publishDataSet=[pub-notif-001, pub-notif-002, pub-notif-003]
  requestDataSet=[]
  responseDataSet=[resp-001, resp-002, resp-003, resp-004]

----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=10.0.0.37
username=sdk01
group=Auction_Publish
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
15:47:52.196 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
15:47:53.548 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
```

```
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853673689]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853675695]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853677696]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853679697]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853681699]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853683700]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853685701]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
```

```
content:
  contentTags=[NOTIF-TAG-201]
  contentType=PLAIN_TEXT
  value=NOTIFICATION[1437853687703]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853689704]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853691705]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853693706]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853695710]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853697711]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853699712]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853701713]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
```

```
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853703714]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853705715]pub-notif-002
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853707717]pub-notif-003
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853709717]pub-notif-001
Publishing notification: GenericMessage:
messageType=NOTIFICATION
capabilityName=Auction
operationName=sampleNotification
body:
  content:
    contentTags=[NOTIF-TAG-201]
    contentType=PLAIN_TEXT
    value=NOTIFICATION[1437853711718]pub-notif-002
Press <enter> to disconnect...Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853868986]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853869075]resp-001 - for request[QUERY[1437853868986]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
```

```
value=QUERY[1437853869589]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN_TEXT
value=RESPONSE[1437853869616]resp-002 - for request[QUERY[1437853869589]req-002]
15:51:10.148 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN_TEXT
value=QUERY[1437853870656]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN_TEXT
value=RESPONSE[1437853870693]resp-003 - for request[QUERY[1437853870656]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN_TEXT
value=QUERY[1437853871201]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN_TEXT
value=RESPONSE[1437853871231]resp-004 - for request[QUERY[1437853871201]req-002]
15:51:11.776 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN_TEXT
value=QUERY[1437853872281]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN_TEXT
value=RESPONSE[1437853872418]resp-001 - for request[QUERY[1437853872281]req-001]
```

```
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853872924]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853872950]resp-002 - for request[QUERY[1437853872924]req-002]
15:51:13.485 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853873991]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853874019]resp-003 - for request[QUERY[1437853873991]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853874538]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853874566]resp-004 - for request[QUERY[1437853874538]req-002]
15:51:15.106 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853875612]req-001
Returning response: GenericMessage:
messageType=RESPONSE
```



```
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853875639]resp-001 - for request[QUERY[1437853875612]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
  body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853876175]resp-002 - for request[QUERY[1437853876145]req-002]
15:51:16.719 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853877240]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853877270]resp-003 - for request[QUERY[1437853877240]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[143785387776]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853877800]resp-004 - for request[QUERY[143785387776]req-002]
15:51:18.383 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853878895]req-001
```

```
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853878925]resp-001 - for request[QUERY[1437853878895]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853879433]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853879459]resp-002 - for request[QUERY[1437853879433]req-002]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853896264]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853896285]resp-003 - for request[QUERY[1437853896264]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853896885]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853896945]resp-004 - for request[QUERY[1437853896885]req-002]
15:51:37.506 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
```

```
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853898077]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853898428]resp-001 - for request[QUERY[1437853898077]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853898938]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853898977]resp-002 - for request[QUERY[1437853898938]req-002]
15:51:39.509 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853900015]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853900041]resp-003 - for request[QUERY[1437853900015]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY[1437853900547]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
```

```
value=RESPONSE[1437853900571]resp-004 - for request[QUERY[1437853900547]req-002]
15:51:41.109 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN TEXT
value=QUERY[1437853901614]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN TEXT
value=RESPONSE[1437853901641]resp-001 - for request[QUERY[1437853901614]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN TEXT
value=QUERY[1437853902147]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN TEXT
value=RESPONSE[1437853902172]resp-002 - for request[QUERY[1437853902147]req-002]
15:51:42.706 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN TEXT
value=QUERY[1437853903210]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
content:
contentTags=[RESP-TAG-101]
contentType=PLAIN TEXT
value=RESPONSE[1437853903237]resp-003 - for request[QUERY[1437853903210]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
content:
contentTags=[QUERY-TAG-301]
contentType=PLAIN TEXT
value=QUERY[1437853903743]req-002
Returning response: GenericMessage:
messageType=RESPONSE
```

```
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
value=RESPONSE [1437853903771]resp-004 - for request[QUERY[1437853903743]req-002]
15:51:44.412 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
  body:
    content:
      contentTags=[RESP-TAG-101]
      contentType=PLAIN_TEXT
      value=RESPONSE [1437853904944]resp-001 - for request[QUERY[1437853904916]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY [1437853905450]req-002
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE [1437853905479]resp-002 - for request[QUERY[1437853905450]req-002]
15:51:46.024 [Smack-Cached Executor 2 (0)] INFO c.c.p.i.GenericMessageDispatcher - Returning error -
Authorization failed for sender: sdk01-sub@xgrid.cisco.com, capability: Auction, operation: BidOnItems
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY [1437853906529]req-001
Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetInventoryItems
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE [1437853906557]resp-003 - for request[QUERY [1437853906529]req-001]
Received request: GenericMessage:
messageType=REQUEST
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[QUERY-TAG-301]
    contentType=PLAIN_TEXT
    value=QUERY [1437853907066]req-002
```

```

Returning response: GenericMessage:
messageType=RESPONSE
capabilityName=Auction
operationName=GetCurrentBids
body:
  content:
    contentTags=[RESP-TAG-101]
    contentType=PLAIN_TEXT
    value=RESPONSE[1437853907099]resp-004 - for request[QUERY[1437853907066]req-002]
    
```

ステップ 2 [管理 (Administration)] > [pxGrid サービス (pxGrid Services)] の順に選択します。
 sdk01 pxGrid クライアントは、パブリッシャとして登録されています

The screenshot shows the Cisco Identity Services Engine Administration console. The navigation menu includes Home, Operations, Policy, Guest Access, Administration, and Work Centers. Under Administration, the 'pxGrid Services' tab is selected. The 'Clients' section is active, displaying a table of clients. The 'sdk01' client is expanded to show its capabilities.

Client Name	Client Description	Capabilities	Status	Client Group(s)	Log
ise-admin-ise238		Capabilities(4 Pub, 2 Sub)	Online	Administrator	View
ise-mnt-ise238		Capabilities(2 Pub, 1 Sub)	Online	Administrator	View
sdk01-sub		Capabilities(0 Pub, 2 Sub)	Online	Auction_Subscribe	View
ise-sxp-ise238		Capabilities(1 Pub, 1 Sub)	Online	Administrator	View
▼ sdk01		Capabilities(1 Pub, 1 Sub)	Online	Basic,Session,Auction_Publish	View

Capability Detail			
Capability Name	Capability Version	Messaging Role	Message Filter
<input type="radio"/> Auction	1.0	Pub	
<input type="radio"/> Core	1.0	Sub	

1 - 2 of 2 Show 25 per page

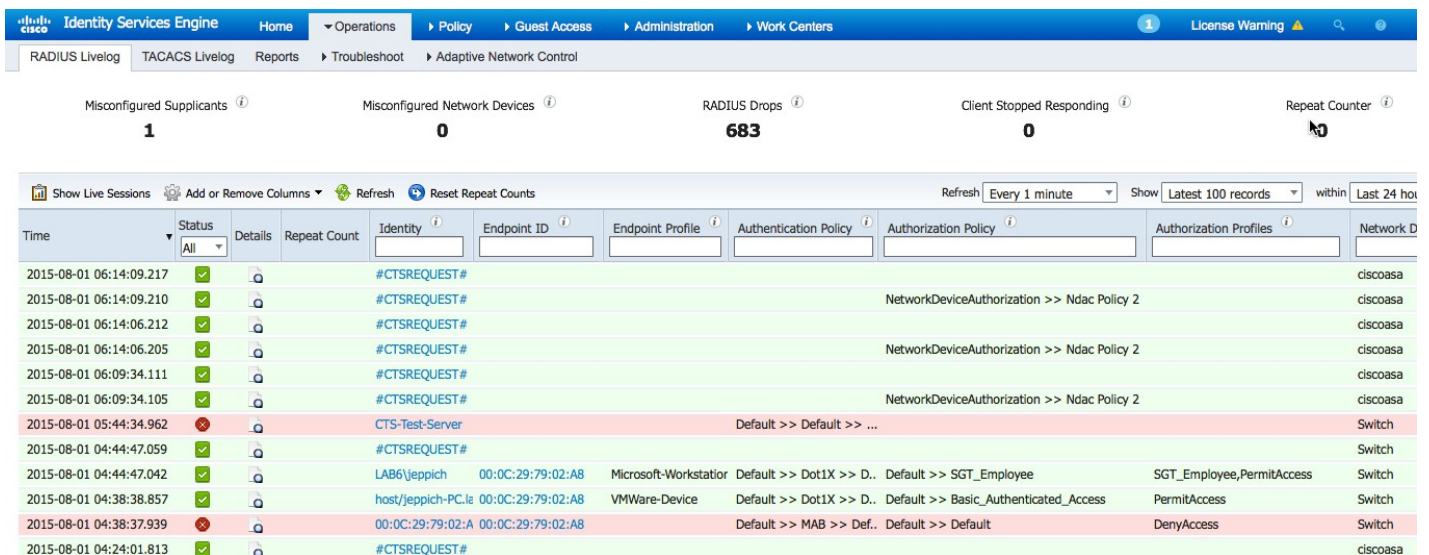
SXP のパブリッシュ

ISE 2.0 は SXP 接続リスナーを提供します。pxGrid は、ISE が、IP アドレス、SGT タグ、送信元、およびピア シーケンスなどの SXP 接続情報をパブリッシュするための機能を提供します。

この情報を取得するには、ISE のサンプル スクリプト `sxp_download` と `sxp_subscribe` スクリプトを使用できます。

この例では、Cisco Catalyst 3750x および ASA 5505 が初期テストに使用されました。これらのデバイスの TrustSec の構成は参考資料のセクションにあります。読者は、シスコの TrustSec ソリューションに精通している必要があります。

SXP バインド設定を構成する前に、SXP が有効なデバイスで CTS が適切に構成されていることを確認します。許可ポリシーで、`#CTSREQUEST#` が適切に表示されていることを確認してください。



The screenshot shows the ISE Operations page with the following summary statistics:

- Misconfigured Suplicants: 1
- Misconfigured Network Devices: 0
- RADIUS Drops: 683
- Client Stopped Responding: 0
- Repeat Counter: 10

The table below displays the SXP records:

Time	Status	Details	Repeat Count	Identity	Endpoint ID	Endpoint Profile	Authentication Policy	Authorization Policy	Authorization Profiles	Network D
2015-08-01 06:14:09.217	✓			#CTSREQUEST#						ciscoasa
2015-08-01 06:14:09.210	✓			#CTSREQUEST#				NetworkDeviceAuthorization >> Ndac Policy 2		ciscoasa
2015-08-01 06:14:06.212	✓			#CTSREQUEST#						ciscoasa
2015-08-01 06:14:06.205	✓			#CTSREQUEST#				NetworkDeviceAuthorization >> Ndac Policy 2		ciscoasa
2015-08-01 06:09:34.111	✓			#CTSREQUEST#						ciscoasa
2015-08-01 06:09:34.105	✓			#CTSREQUEST#				NetworkDeviceAuthorization >> Ndac Policy 2		ciscoasa
2015-08-01 05:44:34.962	✗			CTS-Test-Server			Default >> Default >> ...			Switch
2015-08-01 04:44:47.059	✓			#CTSREQUEST#						Switch
2015-08-01 04:44:47.042	✓			LAB6\jeppich	00:0C:29:79:02:A8	Microsoft-Workstation	Default >> Dot1X >> D..	Default >> SGT_Employee	SGT_Employee,PermitAccess	Switch
2015-08-01 04:38:38.857	✓			host\jeppich-PC.le	00:0C:29:79:02:A8	VMWare-Device	Default >> Dot1X >> D..	Default >> Basic_Authenticated_Access	PermitAccess	Switch
2015-08-01 04:38:37.939	✗			00:0C:29:79:02:A	00:0C:29:79:02:A8		Default >> MAB >> Def..	Default >> Default	DenyAccess	Switch
2015-08-01 04:24:01.813	✓			#CTSREQUEST#						ciscoasa

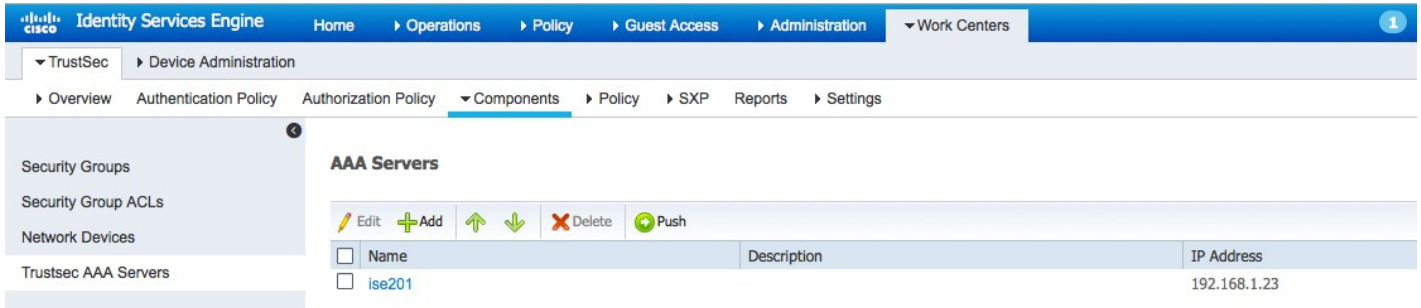
TrustSec の概要に従って手順を進めてください。

[管理 (Administration)] > [展開 (Deployment)] の順に選択してノードを選択し、SXP サービス ポートを有効にすることもできます。

TrustSec AAA デバイス

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [AAA サーバ (AAA Servers)] の順に選択します。

TrustSec AAA サーバは ISE 向けにすでに構成されています



The screenshot shows the Cisco Identity Services Engine (ISE) web interface. The navigation menu includes Home, Operations, Policy, Guest Access, Administration, and Work Centers. The current page is 'TrustSec > Device Administration > Components > AAA Servers'. The main content area displays a table of AAA Servers with the following data:

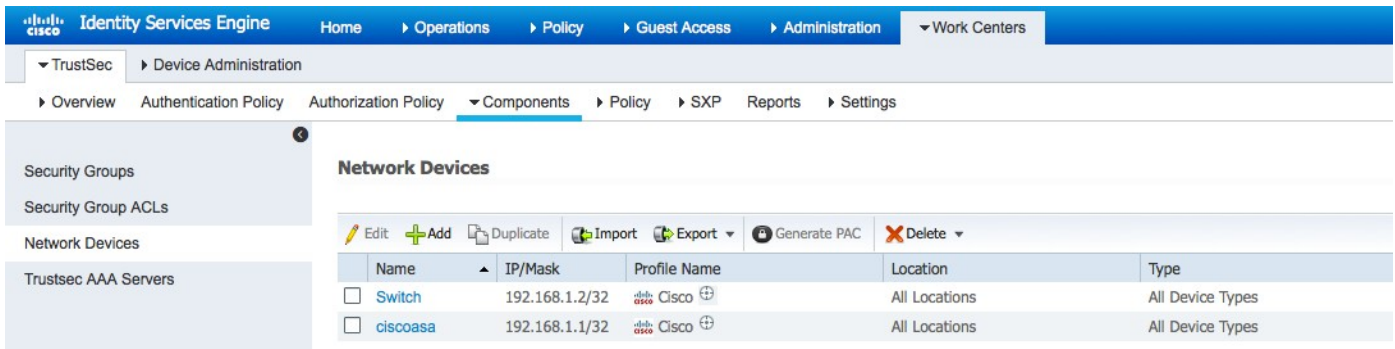
Name	Description	IP Address
ise201		192.168.1.23

TrustSec 向けネットワーク デバイスの構成

TrustSec の動作用にネットワーク デバイスを定義します。Cisco Catalyst 3750x スイッチおよび ASA 5505 が定義されています。

Cisco Catalyst 3750-x

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [ネットワーク デバイス (Network Devices)] の順に選択します。

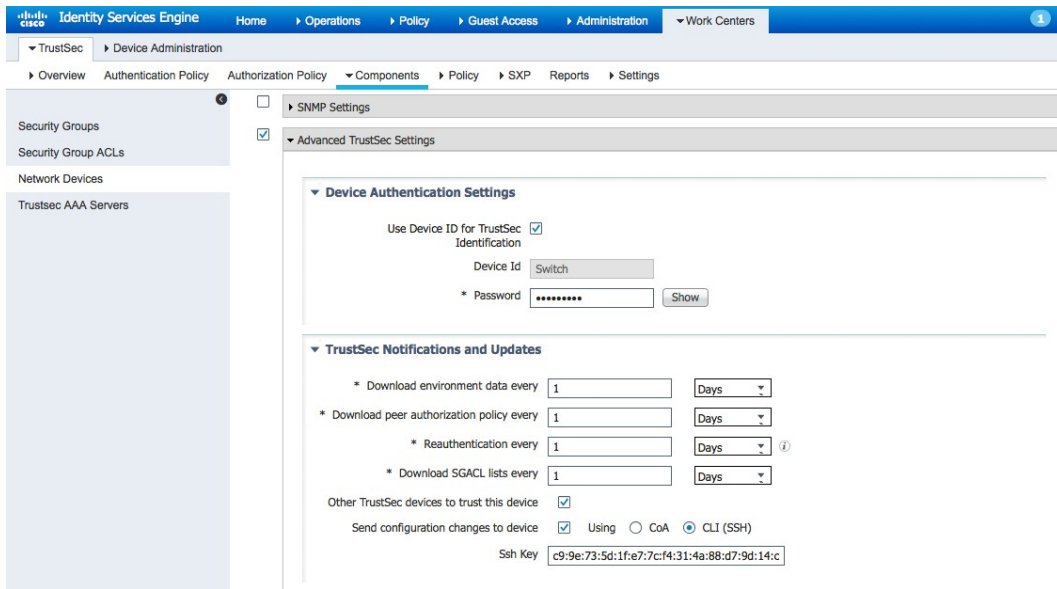


The screenshot shows the Cisco Identity Services Engine (ISE) web interface. The navigation menu includes Home, Operations, Policy, Guest Access, Administration, and Work Centers. The current page is 'TrustSec > Device Administration > Components > Network Devices'. The main content area displays a table of Network Devices with the following data:

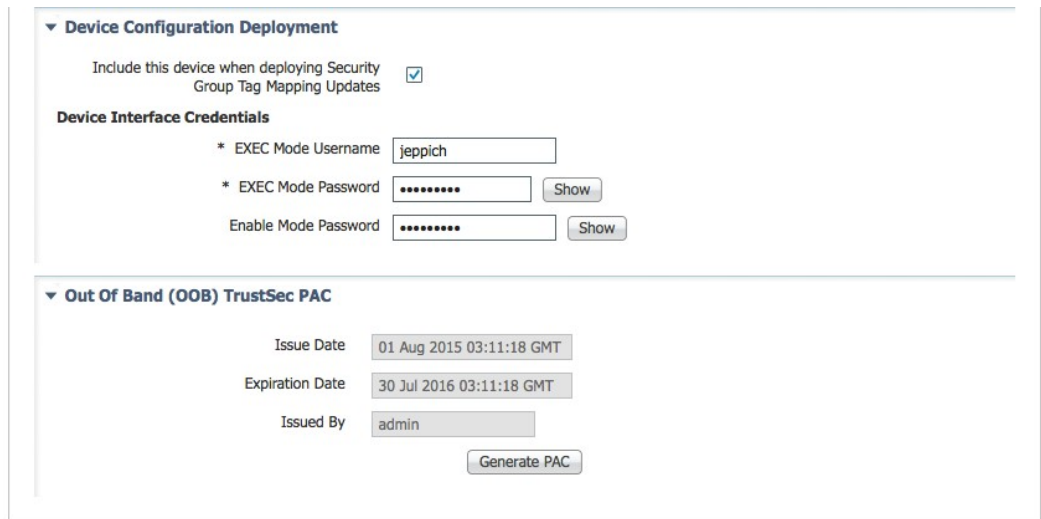
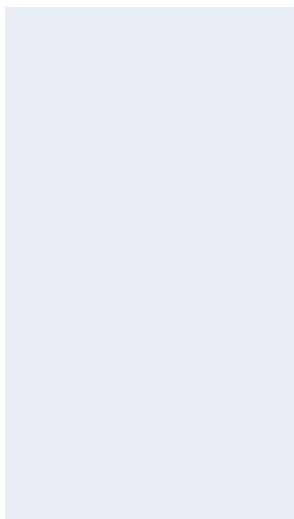
Name	IP/Mask	Profile Name	Location	Type
Switch	192.168.1.2/32	Cisco	All Locations	All Device Types
ciscoasa	192.168.1.1/32	Cisco	All Locations	All Device Types

- ステップ 2** [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [ネットワーク デバイス (Network Devices)] の順に選択します。
- ステップ 3** [TrustSec ID にデバイス ID を使用 (Use Device ID for TrustSec Identification)] を選択します。
- ステップ 4** [CLI (SSH)] を使用してデバイスに構成変更を送信 (Send configuration changes to device using CLI (SSH)) を選択します。

(注) SSH キーを知っている必要があります。SSH キーを知らない場合は、known-hosts ファイルの下のデバイスの IP アドレスを削除できます。IP アドレスに対して SSH 接続を行うと、SSH キーが表示されます。



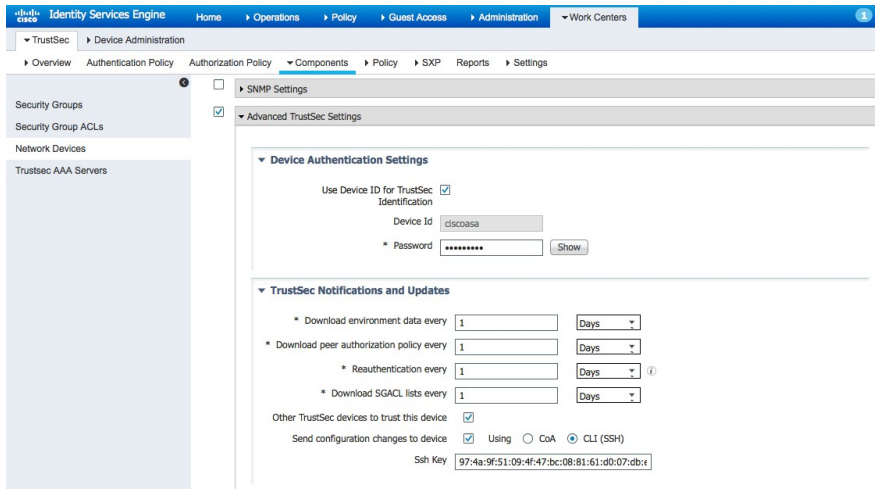
- ステップ 5** [デバイス構成の展開 (Device Configuration Deployment)] の下の [セキュリティグループ タグの更新の展開時にこのデバイスを含める (Include this devices when deploying Security Group Tag Updates)] を有効にします。
- ステップ 6** [デバイス インターフェイス クレデンシヤル (Device Interface Credential)] に情報を入力します。



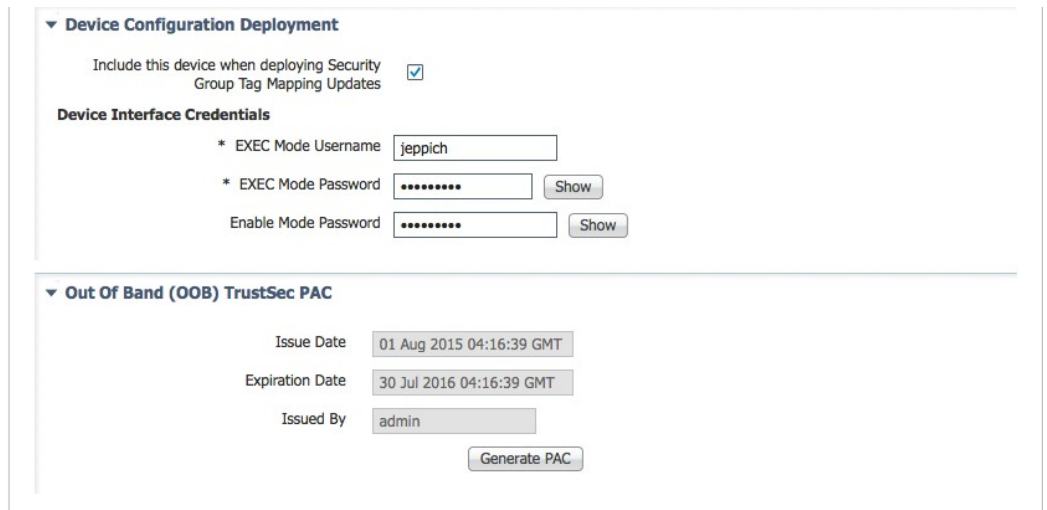
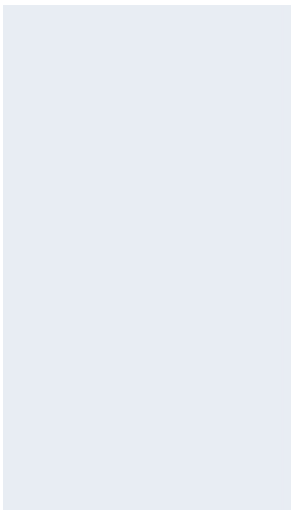
- ステップ 7** 必要に応じて PAC を生成します。

ASA 5505

- ステップ 1** [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [ネットワーク デバイス (Network Devices)] の順に選択します。
- ステップ 2** [TrustSec ID にデバイス ID を使用 (Use Device ID for TrustSec Identification)] を選択します。
- ステップ 3** [CLI (SSH)] を使用してデバイスに構成変更を送信 (Send configuration changes to device using CLI (SSH)) を選択します。



- ステップ 4** [デバイス構成の展開 (Device Configuration Deployment)] の下の [セキュリティグループ タグの更新の展開時にこのデバイスを含める (Include this devices when deploying Security Group Tag Updates)] を有効にします。
- ステップ 5** [デバイス インターフェイス クレデンシヤル (Device Interface Credential)] に情報を入力します。



TrustSec 設定の構成

このドキュメントでは、デフォルトが使用されています。

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [設定 (Settings)] の順に選択します。

General TrustSec Settings

Protected Access Credential (PAC)

*Tunnel PAC Time To Live: 90 Days

*Proactive PAC update when: 10 % PAC TTL is Left

Security Group Tag Numbering

System Will Assign SGT Numbers

Except Numbers In Range - From 1,000 To 1,100

User Must Enter SGT Numbers Manually

セキュリティグループの構成

3750x および ASA5505 SGT タグが作成されました。

ステップ 1 [ワーク センター (Work Centers)] > [コンポーネント (Components)] > [セキュリティグループ (Security Groups)] > [セキュリティグループを追加 (Add security groups)] の順に選択します。

Security Groups

For Policy Export go to Administration > System > Backup & Restore > Policy Export Page

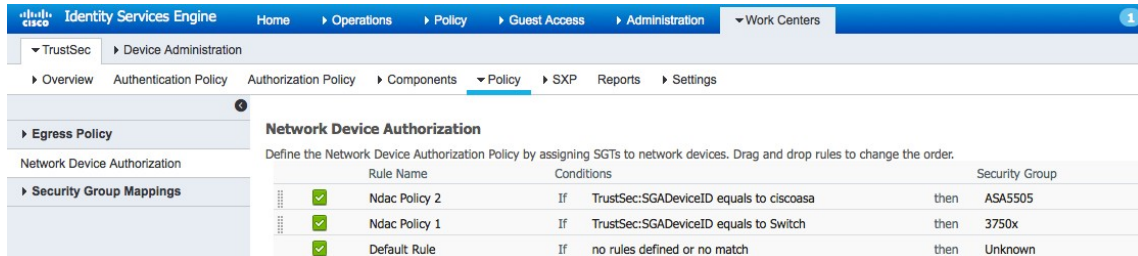
Edit
 Add
 Import
 Export
 Delete
 Push

	Name	SGT (Dec / Hex)	Description
<input type="checkbox"/>	3750x	16/0010	
<input type="checkbox"/>	ASA5505	17/0011	

ネットワーク デバイスの許可ポリシーの設定

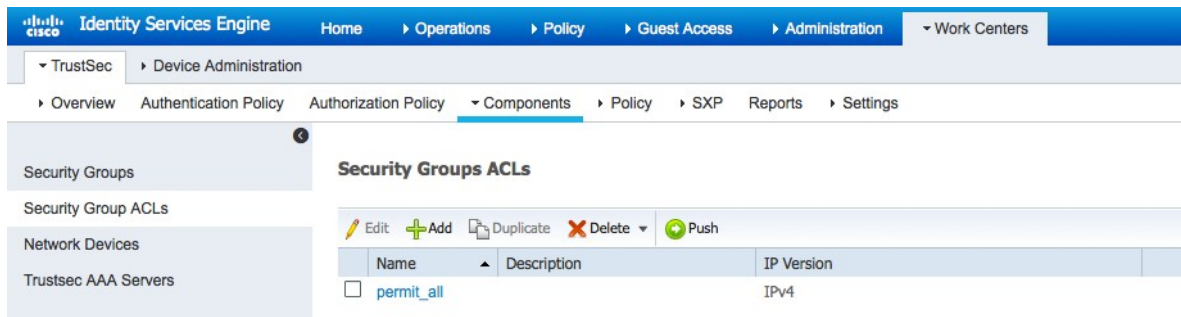
ASA5505 および 3750x セキュリティグループのために 2 つの規則が作成されました

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [ポリシー (Policy)] > [ネットワーク デバイスの規則を追加 (Add network device rules)] の順に選択します。



SGACL の定義

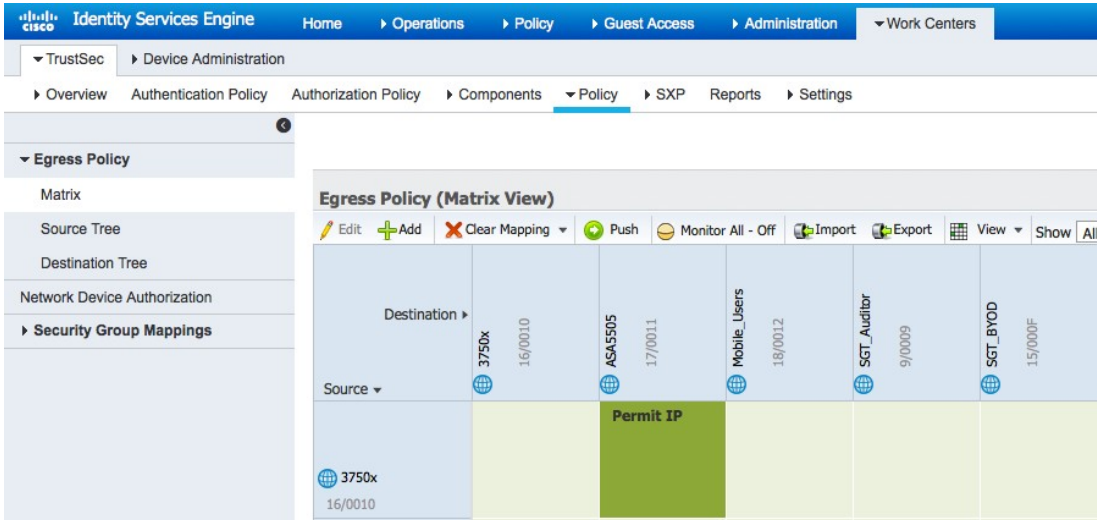
ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [コンポーネント (Components)] > [セキュリティグループ ACL (Security Group ACLs)] の順に選択して、`permit_all` を追加します。



SAGL のマトリクスへの割り当て

SAGL をイーグレス ポリシー マトリクスに割り当てて、他のタグ付けされたネットワーク デバイスへのアクセスを許可します。Cisco 3750x と ASA 5505 の間で、一括でのすべての許可が作成されました。

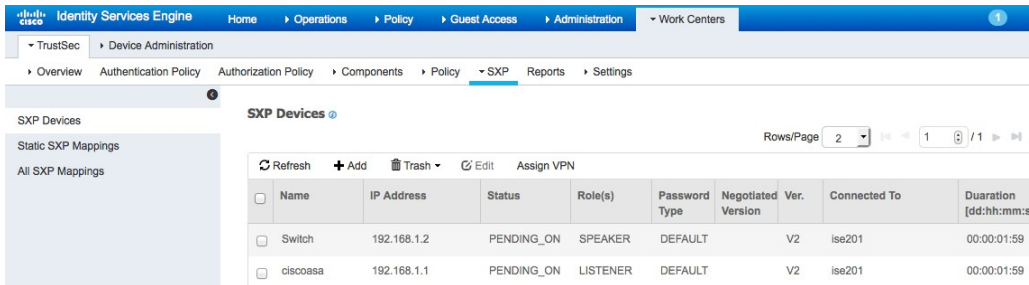
ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [ポリシー (Policy)] > [イーグレス ポリシー マトリクス (Egress Policy Matrix)] > [追加 (Add)] の順に選択します。



IP の分散を SGT マッピングから TrustSec 以外のデバイスに許可するように SXP を構成する

3750x および ASA5505 デバイスは、それらの IP アドレス(ロール)に基づいて定義されます。

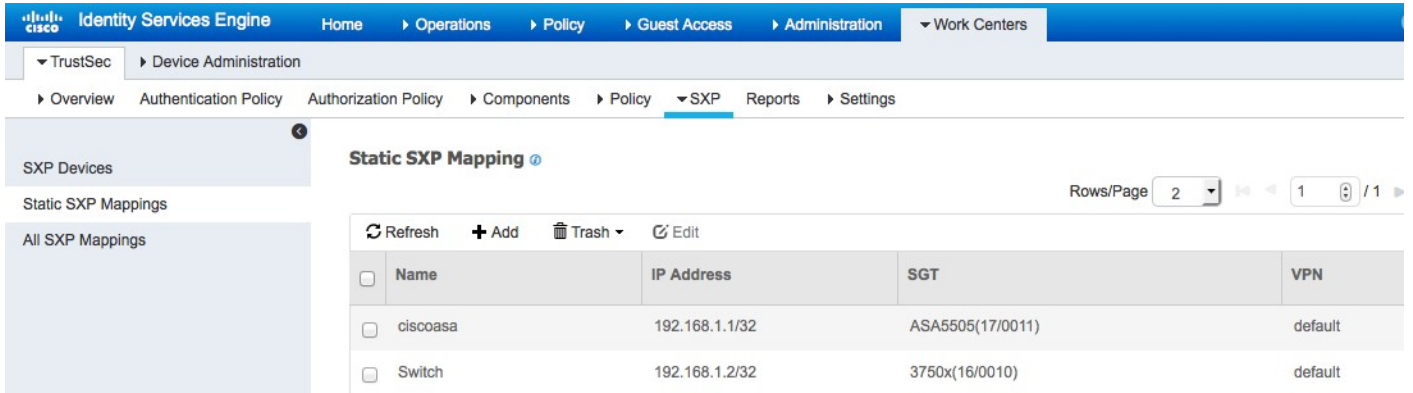
ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [ポリシー (Policy)] > [SXP デバイス (SXP Devices)] > [追加 (Add)] の順に選択します。



静的マッピングの割り当て

3750x および ASA5505 のマッピングが作成され、ネットワークにパブリッシュされました。

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [SXP] の順に選択し、ネットワーク デバイスの静的マッピングを定義します。



Static SXP Mapping

Name	IP Address	SGT	VPN
ciscoasa	192.168.1.1/32	ASA5505(17/0011)	default
Switch	192.168.1.2/32	3750x(16/0010)	default

pxGrid での SXP バインドのパブリッシュ

SXP スクリプトを使用して TrustSec セッション情報を取得できるように、SXP マッピングを pxGrid 上でパブリッシュします。

- ステップ 1** [ワークセンター (Work Centers)] > [TrustSec] > [設定 (Settings)] の順に選択し、[pxGrid 上で SXP バインドをパブリッシュ (Publish SXP bindings on pxGrid)] を有効にします。
- ステップ 2** [RADIUS マッピングを SXP IP SGT マッピング テーブルに追加する (Add radius mappings into SXP IP SGT mapping table)] を有効にします。
- ステップ 3** [グローバル パスワード (Global Password)] を入力します。



Global Password

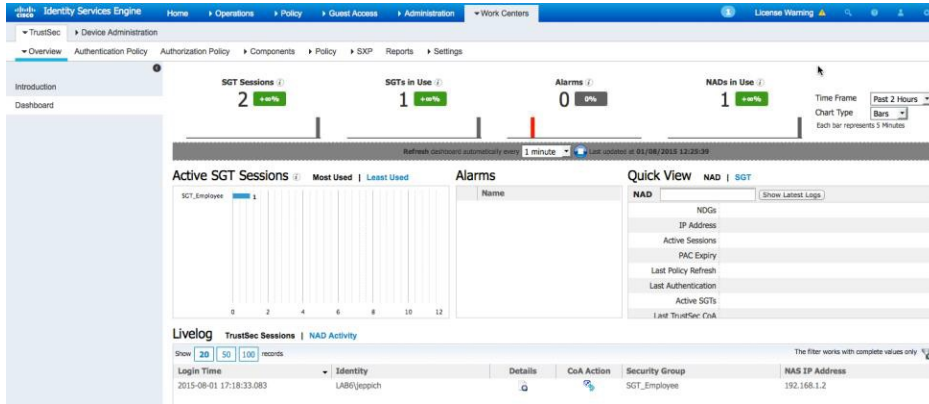
Global Password:

This global password will be overridden by the device specific password

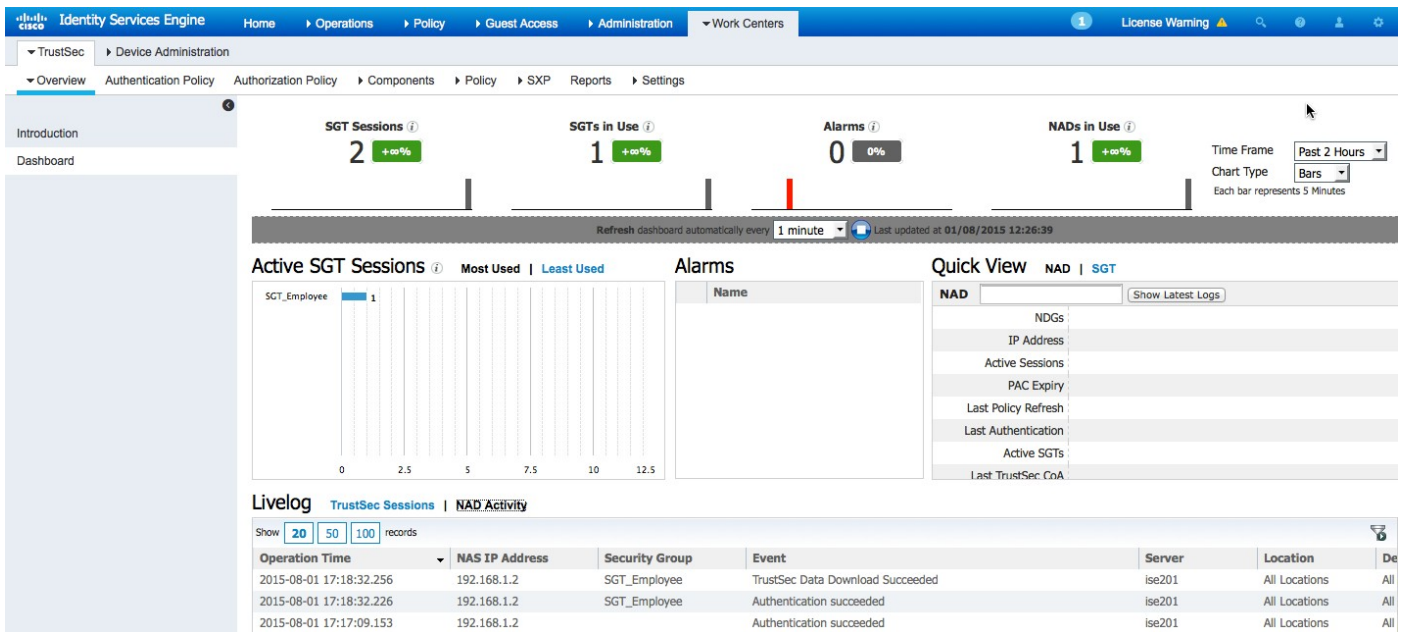
TrustSec ダッシュボード

アクティブな SGT セッションや NAD アクティビティなど、TrustSec アクティビティを表示します。

- ステップ 1** [ワークセンター (Work Centers)] > [TrustSec] > [ダッシュボード (Dashboard)] の順に選択します。



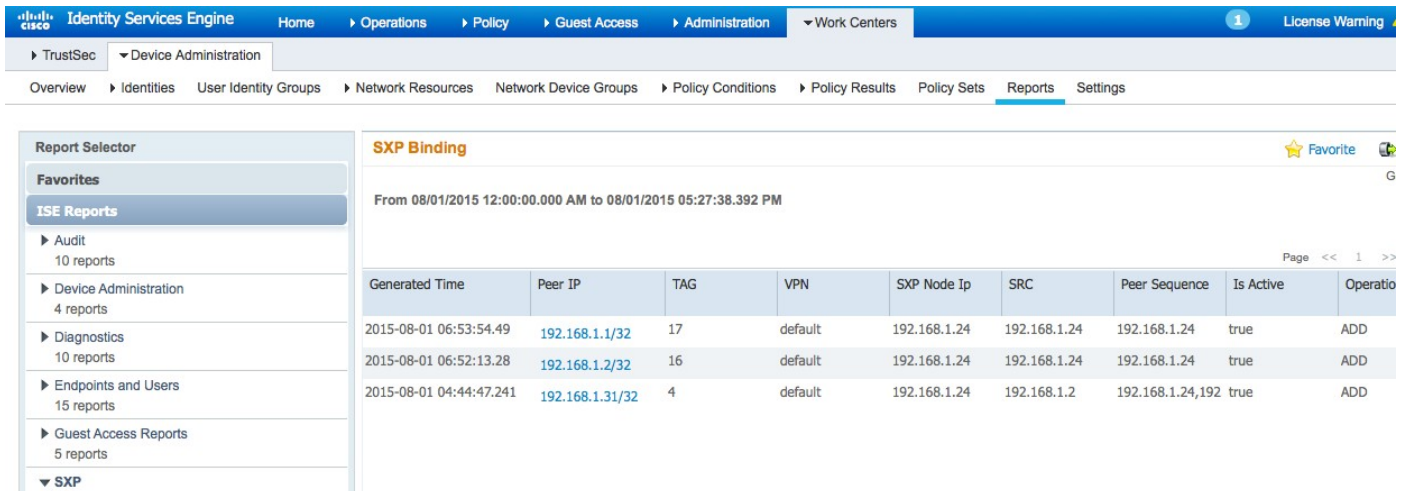
ステップ 2 [NAD アクティビティ(NAD Activity)] を選択します。



SXP バインドのレポート

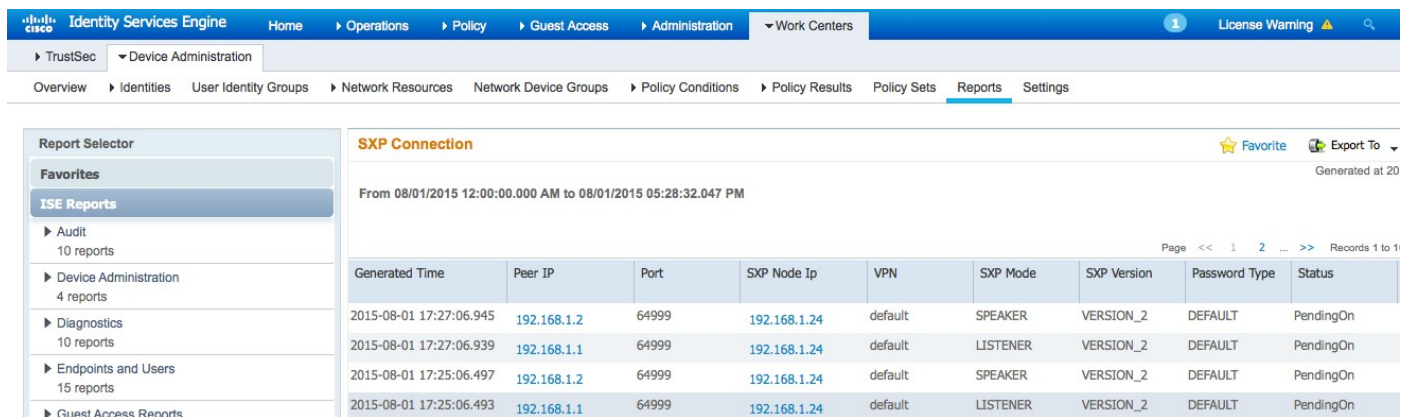
SXP レポートには、バインドと接続の 2 種類があります。

ステップ 1 [ワーク センター (Work Centers)] > [デバイス管理 (Device Administration)] > [レポート (Reports)] > [SXP] > [SXP バインド (SXP Binding)] の順に選択します。



Generated Time	Peer IP	TAG	VPN	SXP Node Ip	SRC	Peer Sequence	Is Active	Operatio
2015-08-01 06:53:54.49	192.168.1.1/32	17	default	192.168.1.24	192.168.1.24	192.168.1.24	true	ADD
2015-08-01 06:52:13.28	192.168.1.2/32	16	default	192.168.1.24	192.168.1.24	192.168.1.24	true	ADD
2015-08-01 04:44:47.241	192.168.1.31/32	4	default	192.168.1.24	192.168.1.2	192.168.1.24,192	true	ADD

ステップ 2 [ワーク センター (Work Centers)] > [デバイス管理 (Device Administration)] > [レポート (Reports)] > [SXP] > [SXP 接続 (SXP Connection)] の順に選択します。



Generated Time	Peer IP	Port	SXP Node Ip	VPN	SXP Mode	SXP Version	Password Type	Status
2015-08-01 17:27:06.945	192.168.1.2	64999	192.168.1.24	default	SPEAKER	VERSION_2	DEFAULT	PendingOn
2015-08-01 17:27:06.939	192.168.1.1	64999	192.168.1.24	default	LISTENER	VERSION_2	DEFAULT	PendingOn
2015-08-01 17:25:06.497	192.168.1.2	64999	192.168.1.24	default	SPEAKER	VERSION_2	DEFAULT	PendingOn
2015-08-01 17:25:06.493	192.168.1.1	64999	192.168.1.24	default	LISTENER	VERSION_2	DEFAULT	PendingOn

sxp_download および sxp_subscribe スクリプト

SXP バインド情報をダウンロードします。

ステップ 1 [ワーク センター (Work Centers)] > [TrustSec] > [SXP] > [静的 SXP マッピング (Static SXP mappings)] の順に選択し、SXP スクリプトをトリガーするためのネットワーク デバイスを追加します。

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers 1

TrustSec Device Administration

Overview Authentication Policy Authorization Policy Components Policy SXP Reports Settings

SXP Devices Static SXP Mappings All SXP Mappings

Static SXP Mapping

Rows/Page 2 1 / 1

<input type="checkbox"/>	Name	IP Address	SGT	VPN
<input type="checkbox"/>	ciscoasa	192.168.1.1/32	ASA5505(17/0011)	default
<input type="checkbox"/>	Switch	192.168.1.2/32	3750x(16/0010)	default

ステップ 2 `sxp_download` スクリプトおよび `sxp_subscribe` スクリプトを実行します。

```
Johns-MacBook-Pro:bin jeppich$ ./sxp_download.sh -a 192.168.1.23 -u mac -k alpha.jks -p cisco123 -t
alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
```

```
12:42:02.433 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:42:03.677 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
SXPBinding={ipPrefix=192.168.1.1/32 tag=17 source=192.168.1.24 peerSequence=192.168.1.24}
SXPBinding={ipPrefix=192.168.1.2/32 tag=16 source=192.168.1.24 peerSequence=192.168.1.24}
Binding count=2
Connection closed
12:42:05.062 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Stopped
Johns-MacBook-Pro:bin jeppich$
```

```
Johns-MacBook-Pro:bin jeppich$ ./sxp_subscribe.sh -a 192.168.1.23 -u mac -k alpha.jks -p cisco123 -t
alpha_root.jks -q cisco123
----- properties -----
version=1.0.2-30-SNAPSHOT
hostnames=192.168.1.23
username=mac
group=Session
description=null
keystoreFilename=alpha.jks
keystorePassword=cisco123
truststoreFilename=alpha_root.jks
truststorePassword=cisco123
-----
12:43:00.420 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Started
Connecting...
Connected
12:43:01.646 [Thread-1] INFO com.cisco.pxgrid.ReconnectionManager - Connected
press <enter> to disconnect...Binding deleted: SXPBinding={ipPrefix=192.168.1.1/32 tag=17
source=192.168.1.24 peerSequence=192.168.1.24}
Binding added: SXPBinding={ipPrefix=192.168.1.1/32 tag=17 source=192.168.1.24 peerSequence=192.168.1.24}
Binding deleted: SXPBinding={ipPrefix=192.168.1.2/32 tag=16 source=192.168.1.24 peerSequence=192.168.1.24}
Binding added: SXPBinding={ipPrefix=192.168.1.2/32 tag=16 source=192.168.1.24 peerSequence=192.168.1.24}
```

トラブルシューティング

一部の基本的なトラブルシューティングの手順について説明します。

```
19:37:39.475 [main] WARN o.a.cxf.phase.PhaseInterceptorChain - Interceptor for  
{https://ise238.lab6.com:8910/pxgrid/mnt/sd}WebClient has thrown exception,  
unwinding now
```

pxGrid クライアントと Windows 7 クライアントの DNS が解決可能であることを確認します。

```
19:37:39.475 [main] WARN o.a.cxf.phase.PhaseInterceptorChain - Interceptor for  
{https://ise238.lab6.com:8910/pxgrid/mnt/sd}WebClient has thrown exception, unwinding now
```

```
org.apache.cxf.interceptor.Fault: Could not send Message.
```

```
at
```

```
org.apache.cxf.interceptor.MessageSenderInterceptor$MessageSenderEndingInterceptor.handleMessage(MessageSend  
erInterceptor.java:64) ~[cxf-api-2.7.3.jar:2.7.3]
```

参考資料

TrustSec デバイス構成

TrustSec デバイス構成

ASA-5505 向けデバイス構成

ステップ 1 ASA で RADIUS を構成します。

```
conf t
aaa-server isel protocol radius
aaa-server isel host 192.168.1.23 {shared secret}
```

ステップ 1 server-group を作成します。

```
conf t
aaa-server ciscoasa protocol radius
aaa-server ciscoasa (inside) host 192.168.1.23
key Richard08
exit
cts server-group ciscoasa
```

ステップ 2 ネットワーク構成から OOB PAC ファイルをインポートします。

```
conf t
cts import-pac ftp://jeppich:Richard08192.168.1.13/ciscoasa.pac password Richard08 {shared secret}
```

ステップ 3 ASA を SPX リスナーとして構成します。

```
conf t
cts sxp enable
cts sxp default password Richard08 {password should match other SXP devices}
cts sxp default source-ip 192.168.1.1 {ASA internal IP address}
cts sxp connection peer 192.168.1.2 {switch IP address} password default mode local listener
cts sxp default sxp connection peer 192.168.1.37 {bayshore} password default mode local listener
```

ステップ 4 ASA が SGT マッピングを受信しているかどうかチェックするため、次を入力します。

```
conf t
sh cts sxp sgt-map ipv4 detail
```

3750x 向けのデバイス構成

ステップ 1 RADIUS 向けにスイッチを構成します。

```
conf t
aaa authorization network isel group radius
cts authorization list isel
ip device tracking
radius-server host 192.168.23 pac key Richard08
```

ステップ 2 CTS 向けにスイッチを構成します。

```
cts sxp enable
cts sxp default source-ip 192.168.1.2 {ip address of switch}
cts sxp default password Richard08 {shared secret}
cts sxp connection peer 192.168.1.1 (ip address of ASA) password default mode local
```