



VIRTUAL NETWORK BROKER USAGE GUIDE

Version 1.0

06/01/2017

1 ARCHITECTURE

Secure Monitoring is implemented as distributed architecture. Each node doing specific functions. SecMon EMS node handling configurations according to which SecMon filter's traffic. IPsec EMS node handling IPsec tunnel configurations which are created between SecMon and Analyzers VM. Both SecMon VM and Analyzers VM contain IPsec enforcer which fetch configurations from IPsec EMS server. Analyzer nodes have analyzers running to analyze the filtered traffic coming from SecMon node. Below visual representation of whole architecture is given.

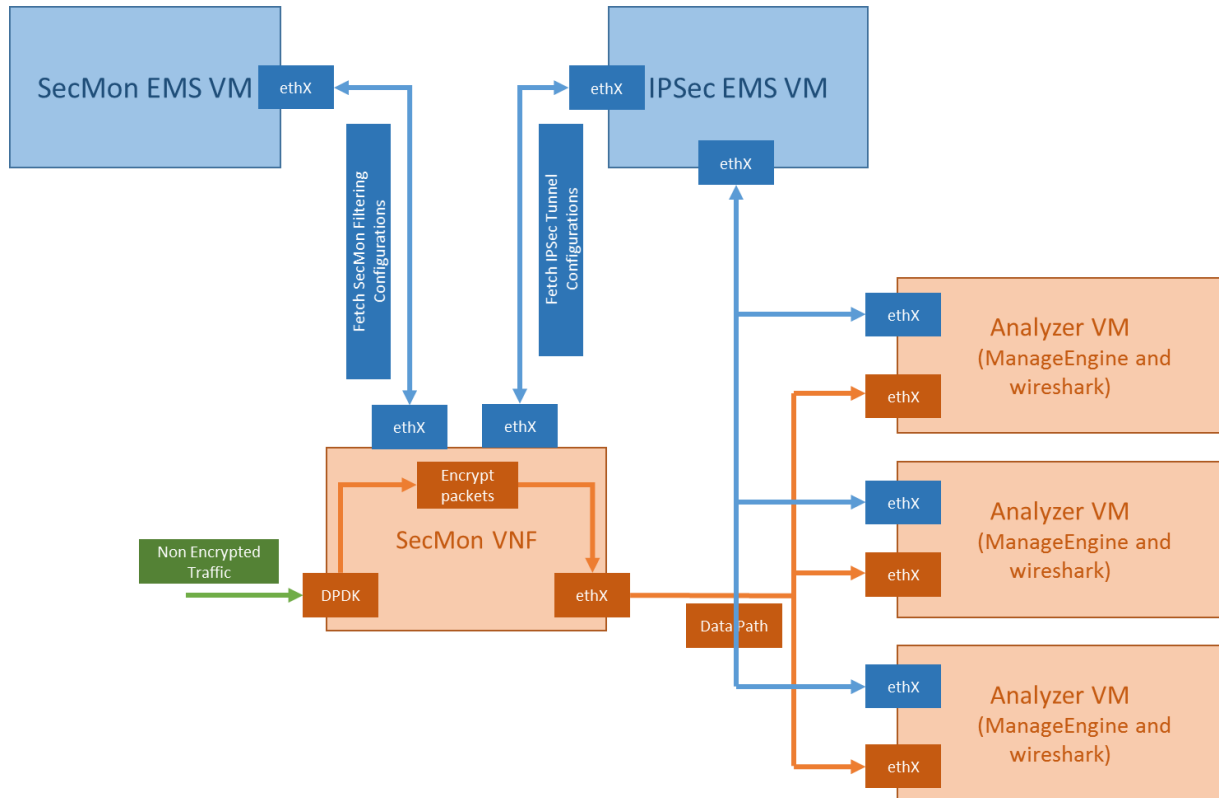


Figure 1 Secure Monitoring Architecture with IPsec Secure Tunneling

1.1 COMPONENTS

1.1.1 SECMON

- The SecMon VNF shall be responsible for all the monitoring/filtering services. The traffic monitoring can be classified based on the scope.
- SecMon VNF supports dynamic plugin architecture, which means, at any point in time, supported plugins can be added and it shall get added/taken into consideration in seamless manner. However it shall be possible to attach any third party plugins to introduce other monitoring protocols.

- SecMon VNF plugin once added, interacts with SecMon EMS system to fetch configurations. Plugins configurations are updated and deleted from Graphical User Interface of SecMon EMS, which interact with SecMon EMS server backend.
- SecMon VNF can also contain IPSec Enforcer to fetch IPSec tunnel configurations from IPSec EMS server. IPSec EMS server configurations are updated and deleted from Graphical User Interface of IPSec EMS, which interact with IPSec EMS backend.

Steps to install SecMon are given below:

1. Run the **dcg_networksecurity_secmon/installer/install.sh** installation script.
2. Choose operations sequentially [Install dependencies, Build SecMon, Configure SecMon and Run SecMon].
3. **Install Dependencies** will install all the required dependencies to run SecMon. This need to be done only once.
4. **Build SecMon** will compile SecMon Agent and build all plugins shared library files.
5. **Configure SecMon** will ask for configuration which are required by SecMon. If your configurations are not changing than this step is also required only once.
 - a. **SecMon Egress Interface** is interface used by SecMon to send packets to Analyzers.
 - b. **SecMon Plugin server IP** is IP address used by server present inside plugins to communicate to SecMon EMS server.
 - c. **SecMon rawforward plugin server port** is port used by rawforward plugin server to communicate to EMS server.
 - d. **SecMon netflow plugin server port** is port used by netflow plugin server to communicate to EMS server.
 - e. **EMS server IP** is IP address of SecMon EMS server.
 - f. **EMS server Port** is port of SecMon EMS server.
 - g. **EMS server Scope** is namespace of which filtering configurations particular instance of SecMon interested in.
 - h. **Interface to be bound to DPDK** is interface that is going to bind to DPDK for packet reception.
6. **Run SecMon** will run SecMon according to configuration provided in previous step.

1.1.2 SECMON EMS

- SecMon EMS is central entity responsible for configuring all the SecMon VNF for specific tenant network and maintains the database for each SecMon VNF
- SecMon EMS can be executed inside the VM.
- It shall expose the REST APIs to receive the SecMon EMS configurations from configuration management interface i.e., CLI / GUI and maintains them in the database.
- It interacts with SecMon VNF over the REST based interface.

- SecMon EMS uses single instance consul DB (bootstrap mode) database for storing Plugins configurations.

Steps to install SecMon EMS are given below:

1. Run **dcg_networksecurity_secmon-ems/install/install.sh** bash script.
2. Choose operations sequentially [Install Dependencies and Run SecMon EMS]
3. **Install Dependencies** will install all the required dependencies to run SecMon EMS. This need to be done only once.
4. **Run SecMon EMS** will run SecMon EMS server in background.
5. Open web browser and go to <http://localhost:9082/maas/> to access GUI.

SecMon EMS GUI has four subsections **Overview Management, Rawforward Management, Netflow Management and SFlow Management.**

1.1.2.1 OVERVIEW MANAGEMENT

Traffic filtering configurations are mentioned in this section. This section is divided into subsections **scope, classification object, rule object and policy.**

1.1.2.1.1 Overview > Scope

We can have many SecMon instances running with different filtering configurations and each instance is differentiated basis on **scope**.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Overview

ScopeClassification ObjectRule ObjectPolicy

Scope

Search ScopesFLUSH+ CREATE SCOPEDELETE SCOPE

	scope name	sflowstatus	netflowstatus	rawforwardstatus	Action
<input type="checkbox"/>	scope1	enable	enable	disable	SELECT ACTION

© Intel corporation 2016

1.1.2.1.2 Overview > Flush

Flush provide the functionality to trigger configurations update on SecMon plugin servers. Changing configurations on EMS Graphical user interface will only effect when that particular scope configurations are **flushed**.

Overview

Netflow

Rawforward

Sflow

Overview

Scope

Classification Object

Rule Object

Policy

Scope

Search Scopes

FLUSH

+ CREATE SCOPE

DELETE SCOPE

	scope name	sflowstatus	netflowstatus	rawforwardstatus	Action
<input type="checkbox"/>	scope1	enable		disable	SELECT ACTION ▾

Flush

x

Scope Name

scope1 ▾

CANCEL

FLUSH

1.1.2.1.3 Overview > Classification Object

Classification Object manage all the traffic filtering configurations. According to these configurations SecMon classify the incoming traffic. We mention configurations to match traffic in which we are interested.

MAAS

UserSignoutHelp

Overview

Netflow

Rawforward

Sflow

Overview

Classification Object

Rule Object

Policy

Classification Object

Search classification objects

+ CREATE CLASSIFICATION OBJECT

DELETE CLASSIFICATION OBJECT

	Classification Object Name	Source IP Address	Source Mac	Source Subnet	Minimum Source Port	Maximum Source Port	Destination IP Address	Destination Mac	Destination Subnet	Minimum Destination Port
<input type="checkbox"/>	Dataplane_ICMP_class	0.0.0.0	*	24	1	65535	0.0.0.0	*	24	1
<input type="checkbox"/>	Dataplane_UDP_class	0.0.0.0	*	24	1	65535	0.0.0.0	*	24	1
<input type="checkbox"/>	Dataplane_TCP_class	0.0.0.0	*	24	1	65535	0.0.0.0	*	24	1

1/1

© Intel corporation 2016

1.1.2.1.4 Overview > Rule Object

Rule Object created in this view is associated with **Classification Object** section and is configured to **drop or forward traffic**. So we can decide which traffic we want to Analyze and which not. This provide fine grain control over filtering criteria.

MAAS

UserSignoutHelp

Overview

Netflow

Rawforward

Sflow

Overview

Classification Object

Rule Object

Policy

Rule Object

Search Rule Objects

+ CREATE RULE OBJECT

DELETE RULE OBJECT

	Rule Object Name	Classification Object	Priority	Truncate to Size	Packet Action	Action
<input type="checkbox"/>	Dataplane_ICMP_Rule	Dataplane_ICMP_class	1	128	Forward	SELECT ACTION ▾
<input type="checkbox"/>	Dataplane_UDP_Rule	Dataplane_UDP_class	1	1300	Forward	SELECT ACTION ▾
<input type="checkbox"/>	Dataplane_TCP_Rule	Dataplane_TCP_class	1	1300	Forward	SELECT ACTION ▾

© Intel corporation 2016

1.1.2.1.5 Overview > Policy

Policy is group of **rules**. Rules created in **rule object** subsection are grouped together to create filtering configurations. Policy is then associated with **Collectors or Collector Sets** of plugins.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Overview

ScopeClassification ObjectRule ObjectPolicy

Policy

Search Policy+ CREATE POLICYDELETE POLICY

	Policy Name	Rule Objects	Action
<input type="checkbox"/>	Policy_TCP	Dataplane_TCP_Rule	SELECT ACTION ▾
<input type="checkbox"/>	Policy_ICMP	Dataplane_ICMP_Rule,Dataplane_UDP_Rule,Dataplane_TCP_Rule	SELECT ACTION ▾
<input type="checkbox"/>	Policy_UDP	Dataplane_UDP_Rule	SELECT ACTION ▾

© Intel corporation 2016

1.1.2.2 NETFLOW MANAGEMENT

Netflow plugin configurations are mentioned in this section. Netflow management is divided into subsections **collector**, **config**, **monitor**, **association** and **collector set**. Each subsection is handling specific configurations related to Netflow plugin.

1.1.2.2.1 NetFlow > NetFlow Collector

Netflow collector field manages configurations related to the collectors (or Analyzers) which going to receive Netflow traffic. **IP address and Port** combination is used to identify individual collector. Netflow collectors generally uses **port 2055**.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Netflow

Netflow CollectorNetflow ConfigNetflow MonitorNetflow AssociationNetflow Collectorset

Netflow Collector

Search netflowcollector+ CREATE NETFLOW COLLECTORDELETE NETFLOW COLLECTOR

	NetFlow Collector Name	Collector IP	Collector UDP Port	Action
<input type="checkbox"/>	NF_Collector_Tool	10.212.92.31	2055	SELECT ACTION

© Intel corporation 2016

1.1.2.2.2 NetFlow > NetFlow Config

Netflow Config manages Netflow configurations. **Active timeout** field manages how long after which active traffic should be exported. **Inactive timeout** field manages how long after which non active traffic should be exported. Similarly, other fields manage Netflow related configurations.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Netflow

Netflow CollectorNetflow ConfigNetflow MonitorNetflow AssociationNetflow Collectorset

Netflow Config

Search netflowconfigDELETE NETFLOW CONFIG

	Scope Name	Active Timeout	Inactive Timeout	Refresh Rate	Timeout Rate	MaxFlows	Action
<input type="checkbox"/>	scope1	10	10	10	10	60	<div>SELECT ACTION</div>

© Intel corporation 2016

1.1.2.2.3 NetFlow > NetFlow Monitor

Netflow Monitor manages Netflow monitoring configurations. **Match fields** are key fields of packet such as source/destination IP address, source/destination ports and protocol which decide if packet resemble some other packet, on the basis of this unique flows are created. **Collect fields** are non-key fields of packets such as TCP flags, subnet masks and packets but are collected anyways and exported in netflow flow.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Netflow

Netflow CollectorNetflow ConfigNetflow MonitorNetflow AssociationNetflow Collectorset

Netflow Monitor

Search netflowmonitor

DELETE NETFLOW MONITOR

	Scope Name	Match Fields	Collect Fields	Action
<input type="checkbox"/>	scope1	SOURCE-PORT,DESTINATION-PORT,INPUT-INTERFACE,MAC-ADDRESS,VLAN,DESTINATION-ADDRESS,SOURCE-ADDRESS,PROTOCOL,TOS	FLOW ACCESS TIMESTAMP,COLLECT COUNTER,MAC-ADDRESS,VLAN	<div>SELECT ACTION</div>

© Intel corporation 2016

1.1.2.2.4 NetFlow > NetFlow Association

Netflow Association creates association between **Netflow collector** and **policy** for specific **scope**. Collector only receives traffic according to filtering configurations mentioned in **policy**.

MAAS

UserSignoutHelp

Overview

Netflow

Rawforward

Sflow

Netflow

Netflow CollectorNetflow ConfigNetflow MonitorNetflow AssociationNetflow Collectorset

Netflow Association

Search netflowassociation+ CREATE NETFLOW ASSOCIATIONDELETE NETFLOW ASSOCIATION

	Source Vm Name	Scope Name	NetFlow Collector/CollectorSet Name	Direction	Policy Name	Action
<input type="checkbox"/>	Control_Plane	scope1	NF_Collector_Tool	BOTH	Policy_ICMP	SELECT ACTION

© Intel corporation 2016

1.1.2.2.5 NetFlow > NetFlow CollectorSet

Netflow collectorset manages collector set's configurations. Plugins support load balancing between different collectors. Load balancing happen by creating set of collectors. Traffic is divided among these collectors on the basis of load balancing algorithms. Currently **round-robin, session based and weighted round robin** algorithms are implemented for load balancing.

MAAS

UserSignoutHelp

Overview

Netflow

Rawforward

Sflow

Netflow

Netflow CollectorNetflow ConfigNetflow MonitorNetflow AssociationNetflow Collectorset

Netflow Collectorset

Search netflowcollectorset

+ CREATE NETFLOW COLLECTORSET

DELETE NETFLOW COLLECTORSET

NetFlow CollectorSet Name	NetFlow Collector Name	NetFlow Algorithm	Action
<input type="checkbox"/> NF_Collectorset	[[{'id': '148c029f-999d-4055-a443-53a8eda72567', 'weight': 20}, {'id': '7db01582-5e19-4448-8356-508feaab32bc', 'weight': 30}, {'id': 'f6d8474c-2af1-4c77-9829-f31a3bd0f551', 'weight': 50}]]	2	<div>SELECT ACTION</div>

© Intel corporation 2016

1.1.2.3 RAWFORWARD MANAGEMENT

Rawforward Management manages Rawforward plugin related configurations.

1.1.2.3.1 RawForward > RawForward Collector

Rawforward Collector section manages configurations related to collector. **IP address and Port** combination is used to identify individual Collector (or Analyzer) where traffic should go for being analyzed.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Rawforward

Rawforward CollectorRawforward AssociationRawforward Collectorset

Rawforward Collector

Search rawforwardcollector+ CREATE RAWFORWARD COLLECTORDELETE RAWFORWARD COLLECTOR

	RawForwarding Collector Name	Collector IP	Collector UDP Port	Encapsulation Protocol	Action
<input type="checkbox"/>	raw forwarding	192.168.1.100	30000	UDP	SELECT ACTION

© Intel corporation 2016

1.1.2.3.2 RawForward > RawForward Association

Rawforward Association creates association between Rawforward **collector** and **policy** for specific **scope**. Collector only receive traffic according to filtering configurations mentioned in **policy**.

MAAS

UserSignoutHelp

Overview

Netflow

Rawforward

Sflow

Rawforward

Rawforward CollectorRawforward AssociationRawforward Collectorset

Rawforward Association

Search Rawforward AssociationCREATE RAWFORWARD ASSOCIATIONDELETE RAWFORWARD ASSOCIATION

	Source Vm Name	Scope Name	RawForwarding Collector/CollectorSet Name	Direction	Policy Name	Action
<input type="checkbox"/>	PDN	scope1	raw forwarding	BOTH	Policy_UDP	SELECT ACTION

© Intel corporation 2016

1.1.2.3.3 RawForward > RawForward Collectorset

Rawforward collectorset manages collector sets configurations. Plugins support load balancing between different collectors. Load balancing happen by creating set of collectors. Traffic is divided among these collectors on the basis of load balancing algorithms. Currently **round-robin, session based and weighted round robin** algorithms are implemented for load balancing.

MAAS

UserSignoutHelp

OverviewNetflowRawforwardSflow

Rawforward

Rawforward

Rawforward CollectorRawforward AssociationRawforward Collectorset

Rawforward Collectorset

Search RawforwardCollectorst+ CREATE RAWFORWARD COLLECTORSETDELETE RAWFORWARD COLLECTORSET

RawForwarding CollectorSet Name	RawForwarding Collector Name	RawForwarding Algorithm	Action
<input type="checkbox"/> RF_Collectorset	[[{"id": "69db6ba1-2c9f-4744-9308-320245b63316", "weight": 20}, {"id": "ef25336c-6030-4140-b531-5985f857d6c3", "weight": 30}, {"id": "fd37c569-48f8-485a-a9b0-fc03a4bb42e5", "weight": 50}]]	2	<div>SELECT ACTION -</div>

© Intel corporation 2016

1.1.3 IPSEC EMS

- IPsec EMS is central entity responsible for configuring IPsec tunnel configurations in each IPsec Enforcers.
- IPsec EMS service can be run on VM also.
- IPsec EMS expose REST API's for IPsec Enforcers. Then these IPsec Enforcers fetch IPsec tunnel configurations from IPsec EMS using these API's.
- IPsec Enforcers listening for changes happen in configurations in IPsec EMS.
- IPsec EMS provide easy to use GUI to configure IPsec configuration.
- IPsec EMS uses single instance consul DB (bootstrap mode) database for storing IPsec configurations of IPsec Enforcers.

Steps to Run IPsec EMS Server are given below:

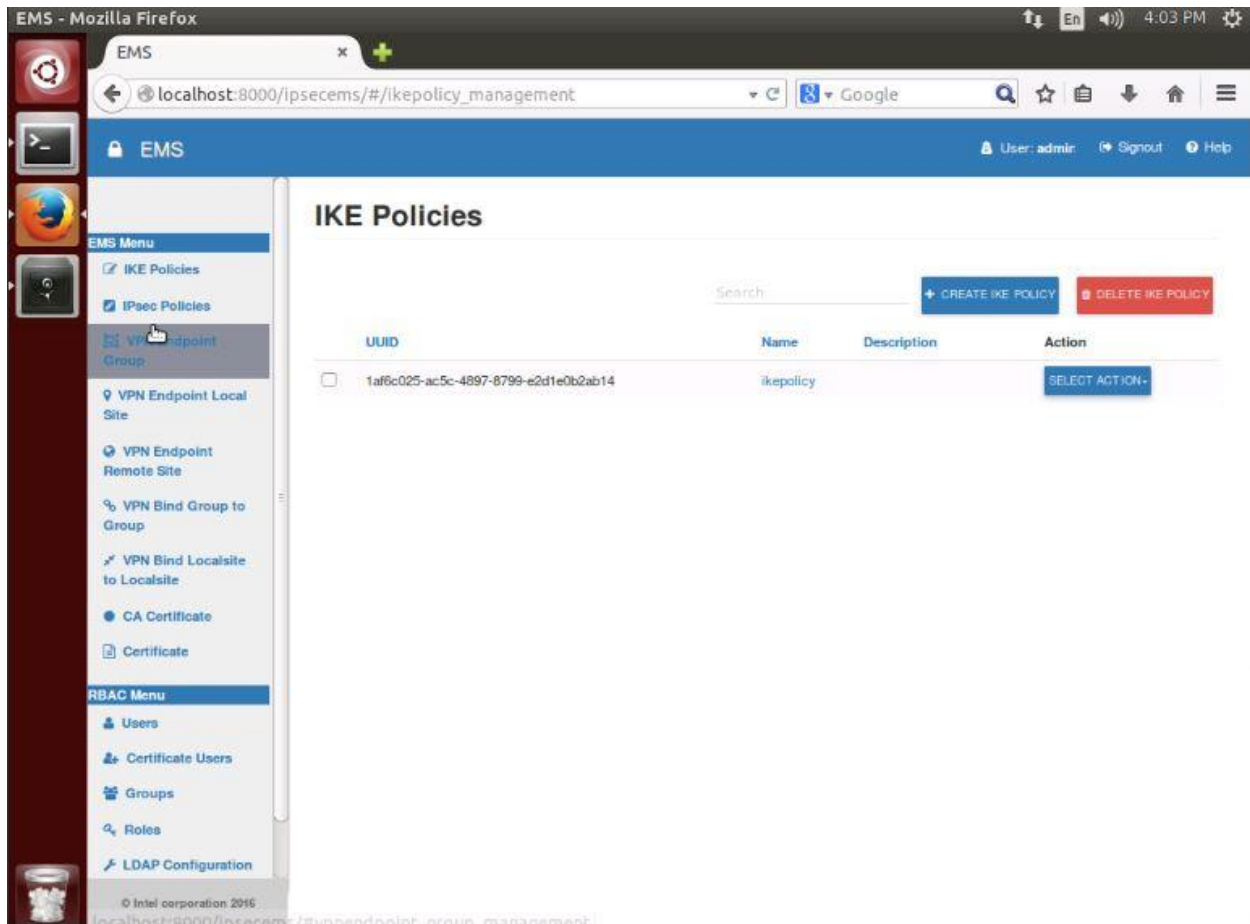
1. Run **dcg_networksecurity_ipsec-ems/common/install.sh** bash script.
2. Choose operations sequentially [Install Dependencies and Run IPsec EMS].
3. **Install Dependencies** will install all the required dependencies to run IPsec EMS. This need to be done only once.
4. **Run IPsec EMS** will run IPsec EMS server in background.
5. Open web browser and go to <http://localhost:8000/ipsecems/> to access GUI.

IPsec EMS GUI broadly divided into two sections **IPsec EMS Menu** and **RBAC Menu**.

1.1.3.1 IPSEC EMS MENU

1.1.3.1.1 IKE POLICY

IKE Policy manages IKE related configurations. Like **IKE version, Encryption Algo, Integrity Algo** and etc. These policies then are used in **VPN Endpoint group to group** and **VPN Endpoint Local site to Local site**.



1.1.3.1.2 IPSEC POLICY

IPsec Policy manages configurations related to IPsec tunnel. Like **Encryption Algo**, **Integrity Algo**, **Encapsulation Protocol** and etc. These policies then are used in **VPN Endpoint group to group** and **VPN Endpoint Localsite to Localsite**.

EMS - Mozilla Firefox

4:04 PM

EMS

localhost:8000/ipsecems/#/ipsecpolicy_management

User: admin | Signout | Help

EMS Menu

IKE Policies

IPsec Policies

VPN Endpoint Group

VPN Endpoint Local Site

VPN Endpoint Remote Site

VPN Bind Group to Group

VPN Bind Localsite to Localsite

CA Certificate

Certificate

RBAC Menu

Users

Certificate Users

Groups

Roles

LDAP Configuration

© Intel corporation 2016

IPSEC Policies

Search

+ CREATE IPSEC POLICY

DELETE IPSEC POLICY

UUID	Name	Description	Action
<input type="checkbox"/> f0ad30c0-3cbb-4701-adfd-8ae930f5479e	ipsecpolicy		<div>SELECT ACTION-</div>

1.1.3.1.3 VPN ENDPOINT GROUP

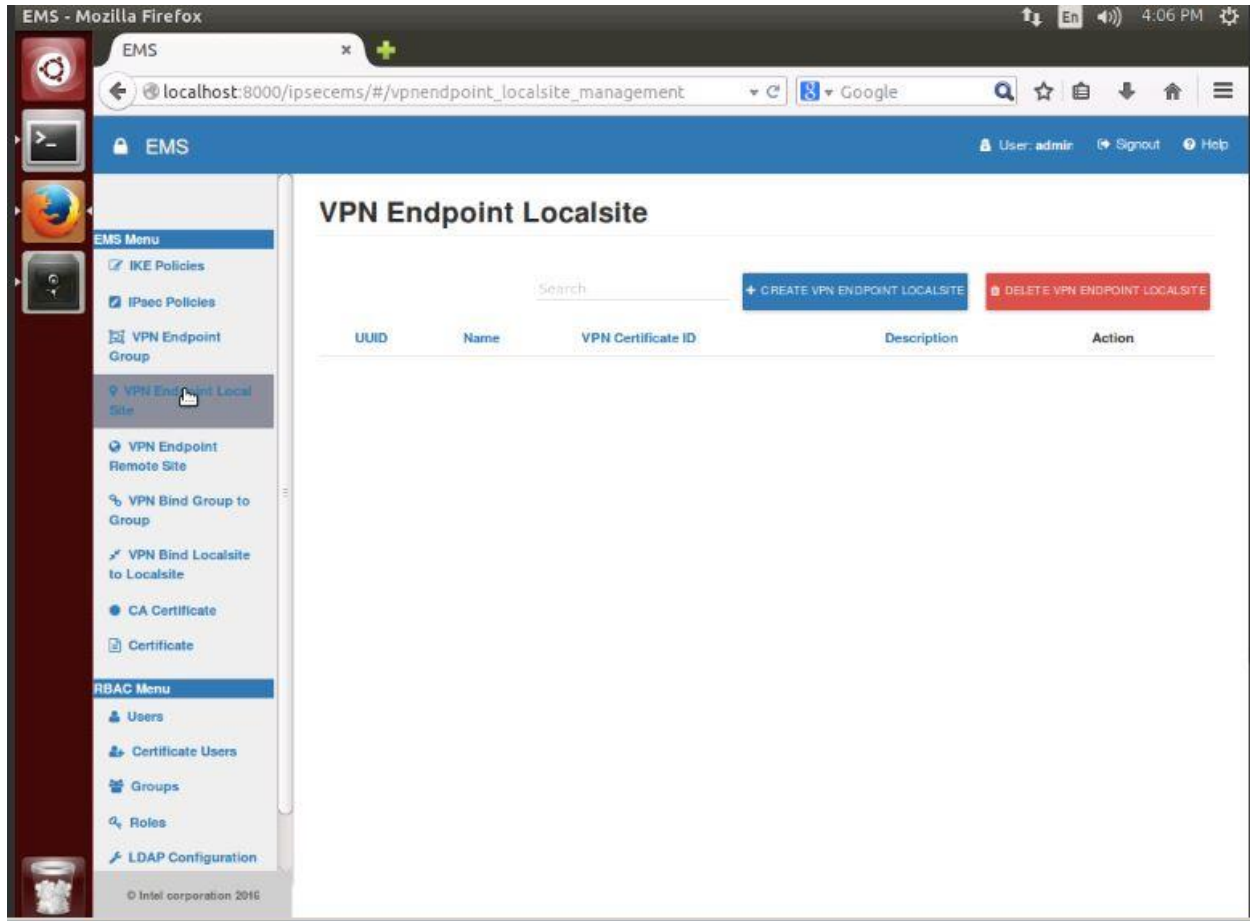
VPN Endpoint Group manages configurations related to grouping of IPsec Enforcers. Enforcers can be grouped together and then we can apply common configurations between them. We can create **m-n tunnel** configurations using Endpoint Groups.

The screenshot shows the EMS (Enterprise Management System) interface in a Mozilla Firefox browser. The address bar displays `localhost:8000/ipsecems/#/vpnendpoint_group_management`. The page title is "VPN Endpoint Group". On the left, there is a sidebar menu with the following items: EMS Menu (IKE Policies, IPsec Policies, VPN Endpoint Group, VPN Endpoint Local Site, VPN Endpoint Remote Site, VPN Bind Group to Group, VPN Bind Local site to Local site, CA Certificate, Certificate), RBAC Menu (Users, Certificate Users, Groups, Roles, LDAP Configuration), and a footer with "© Intel corporation 2016". The main content area features a search bar, two buttons: "+ CREATE VPN ENDPOINT GROUP" and "DELETE VPN ENDPOINT GROUP", and a table with the following columns: UUID, Name, VPN Certificate ID, Description, and Action. The table contains two rows: "group1" with UUID "579e9dd3-6080-4716-92b4-26744c9454ce" and "group2" with UUID "a707c156-e3dd-4db6-abc2-26329b161d29". Each row has a "SELECT ACTION -" button.

UUID	Name	VPN Certificate ID	Description	Action
<input type="checkbox"/> 579e9dd3-6080-4716-92b4-26744c9454ce	group1			SELECT ACTION -
<input type="checkbox"/> a707c156-e3dd-4db6-abc2-26329b161d29	group2			SELECT ACTION -

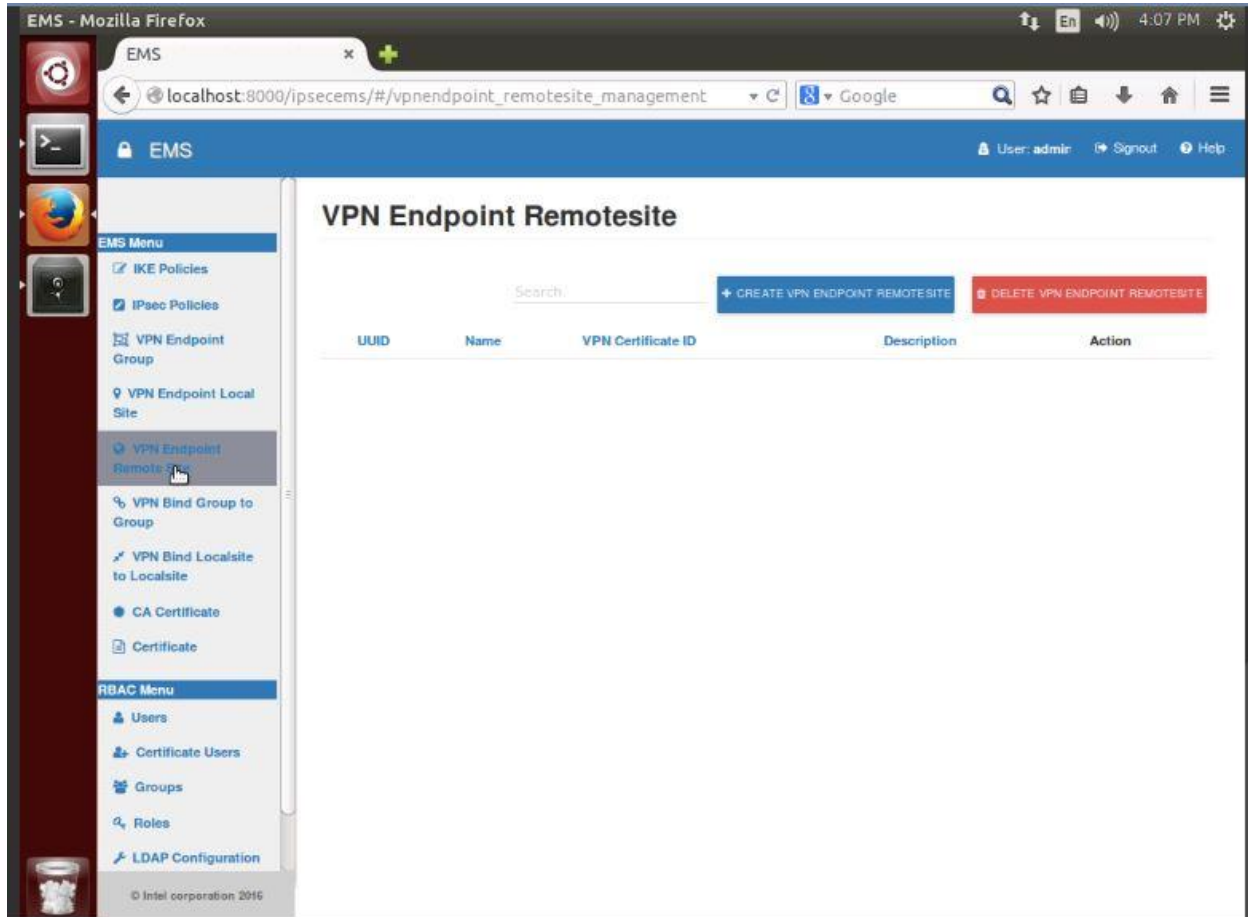
1.1.3.1.4 VPN ENDPOINT LOCALSITE

VPN Endpoint Localsite manages configurations related to localsite endpoints. In this section we mention configurations for IPsec Enforcers which are present inside same network. Like **CIDRS (classless inter domain routing)** and **VPN Certificate ID** (if we are using Certificate based IPsec authentication). **CIDRS** are used to specify the network in which **Endpoint** belongs.



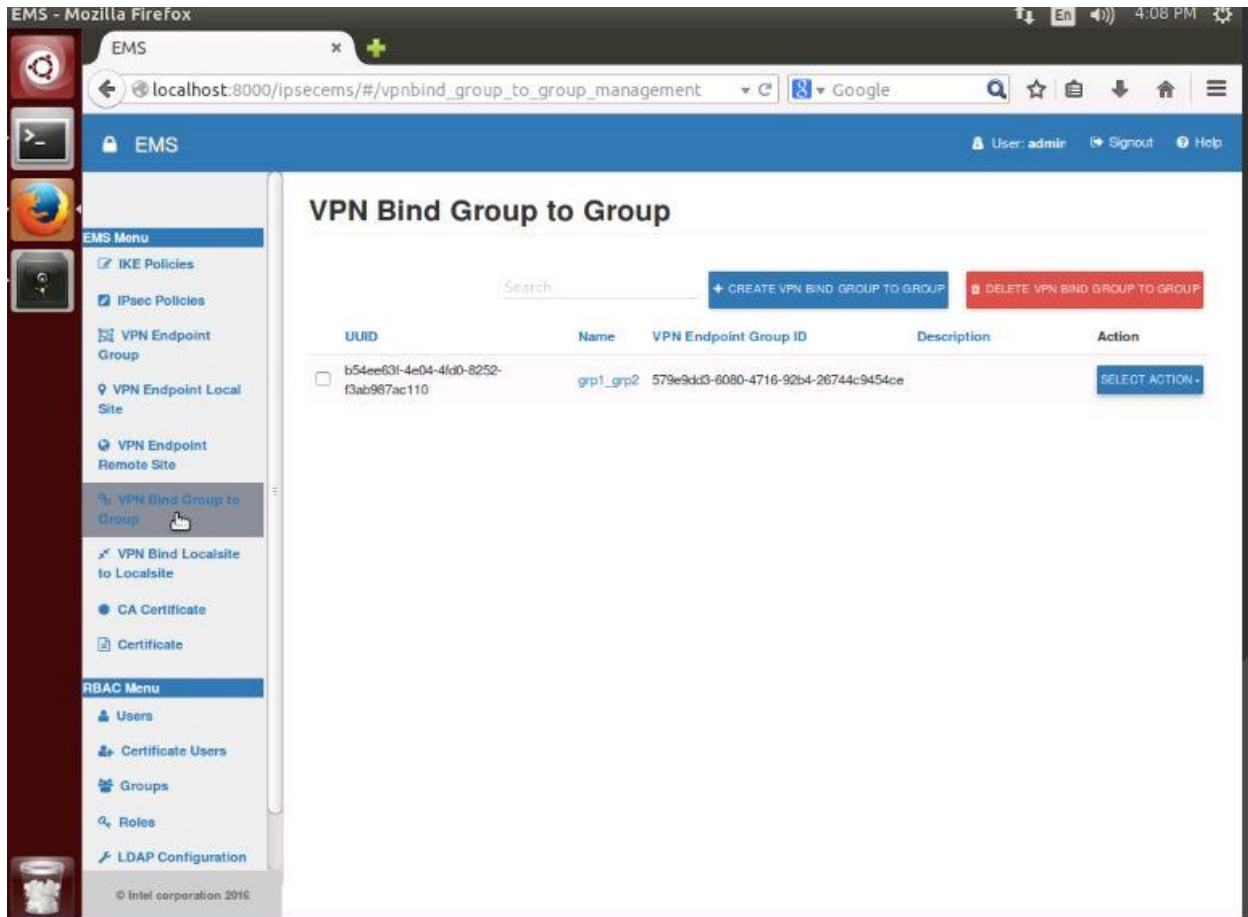
1.1.3.1.5 VPN ENDPOINT REMOTESITE

VPN Endpoint Remotesite manages configurations related to remotesite endpoints. In this section we mention configurations for IPsec Enforcers which are present in outside network. Like **Peer address**, **Peer CIDRS (classless inter domain routing)** and **VPN Certificate ID** (if we are using Certificate based IPsec authentication).



1.1.3.1.6 VPN BIND GROUP TO GROUP

In **VPN Bind group to group** we mention configurations to bind groups together to create **m-n** relationships between IPsec Enforcers. Groups created in **VPN Endpoint Group** are associated with one another as per need.

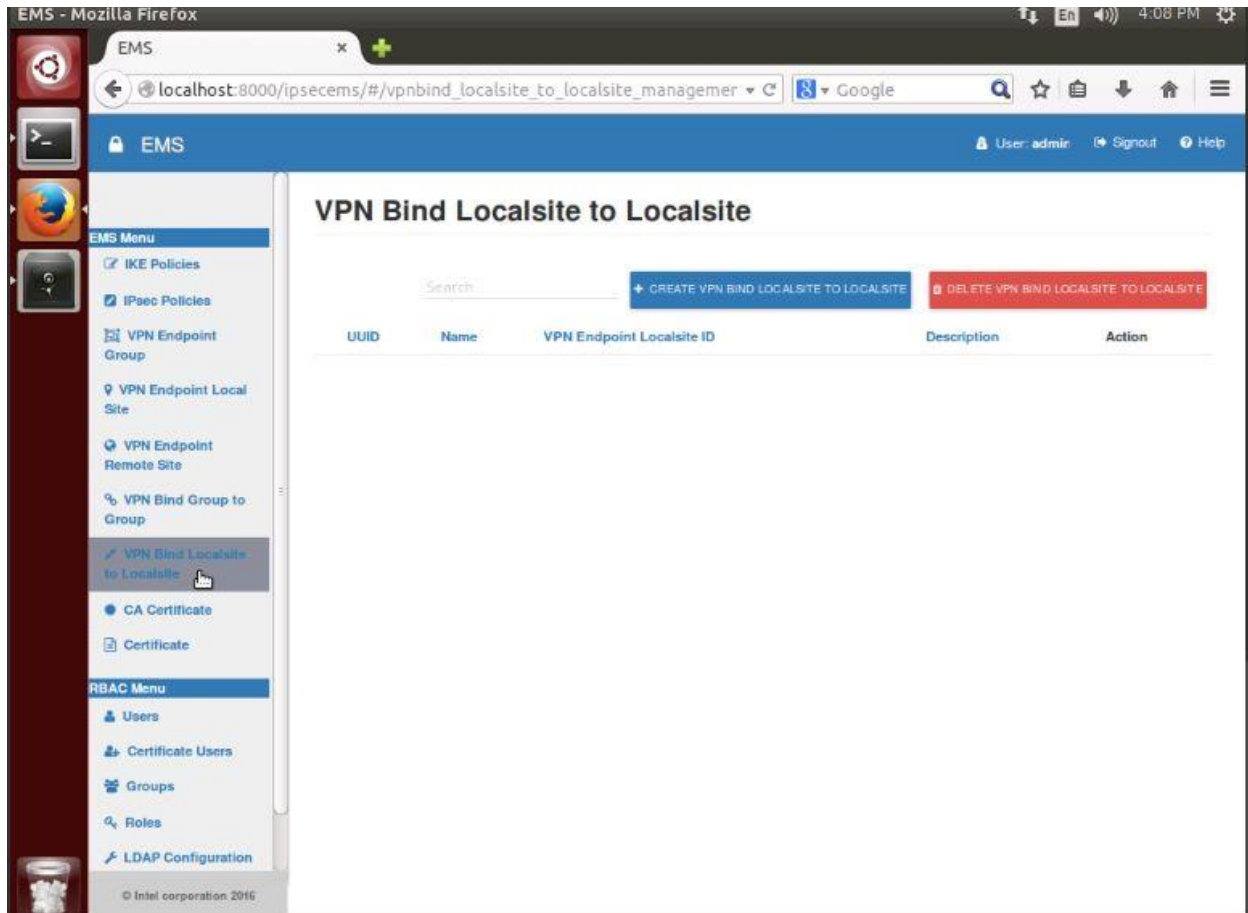


The screenshot shows the EMS web interface in Mozilla Firefox. The browser address bar displays `localhost:8000/ipsecems/#/vpnbind_group_to_group_management`. The page title is "VPN Bind Group to Group". The left sidebar contains the "EMS Menu" with options like IKE Policies, IPsec Policies, VPN Endpoint Group, and VPN Bind Group to Group (which is selected). Below the menu is the "RBAC Menu" with options like Users, Certificate Users, Groups, Roles, and LDAP Configuration. The main content area shows a table with one entry for a group binding. The table has columns for UUID, Name, VPN Endpoint Group ID, Description, and Action. The entry has a checkbox, a UUID, a name "grp1_grp2", a VPN Endpoint Group ID, and a description. The Action column has a "SELECT ACTION" button.

UUID	Name	VPN Endpoint Group ID	Description	Action
<input type="checkbox"/> b54ee631-4e04-4fd0-8252-f3ab987ac110	grp1_grp2	579e9dd3-6080-4716-92b4-26744c9454ce		SELECT ACTION

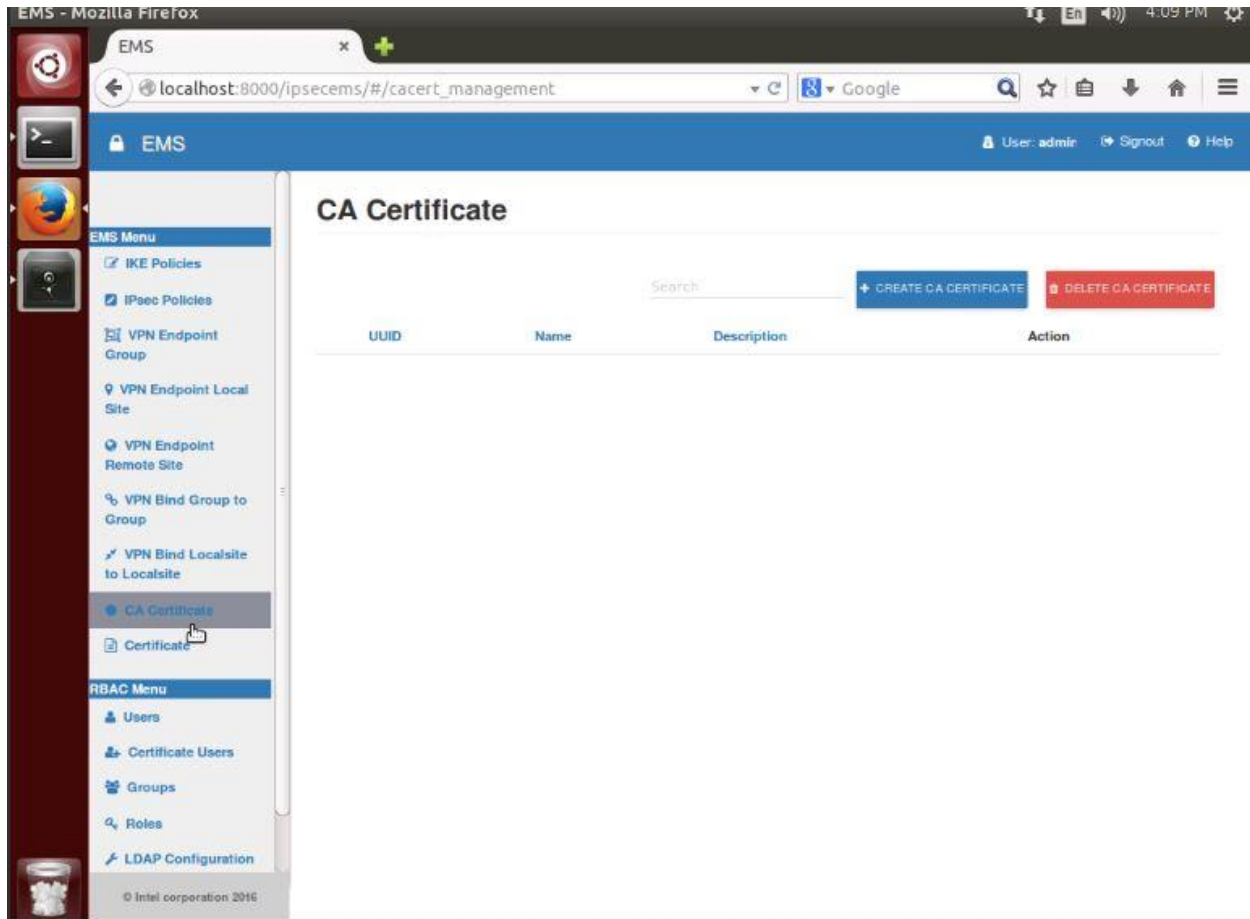
1.1.3.1.7 VPN BIND LOCALSITE TO LOCALSITE

In **VPN Bind localsite to localsite** we mention configurations to create relationships between IPsec Enforcers which are present in same network. Localsites created in **VPN Endpoint Localsite** are associated with one another as per need.



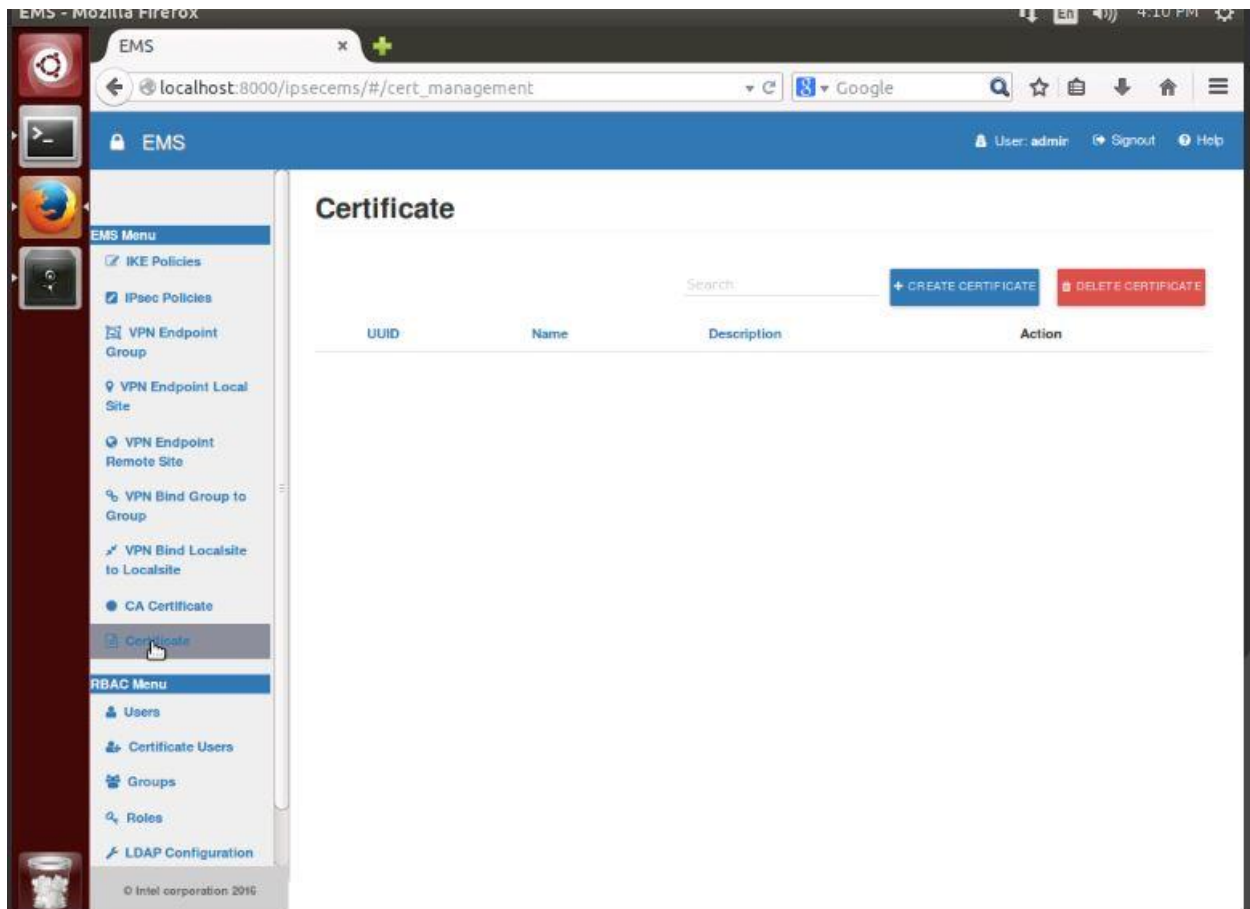
1.1.3.1.8 CA CERTIFICATE

CA Certificate manages Central Authority certificates. Create CA certificate entry by uploading CA certificate already created using tools like OpenSSL. CA certificates are used when certificate based authentication mode of secure tunneling is used.



1.1.3.1.9 CERTIFICATE

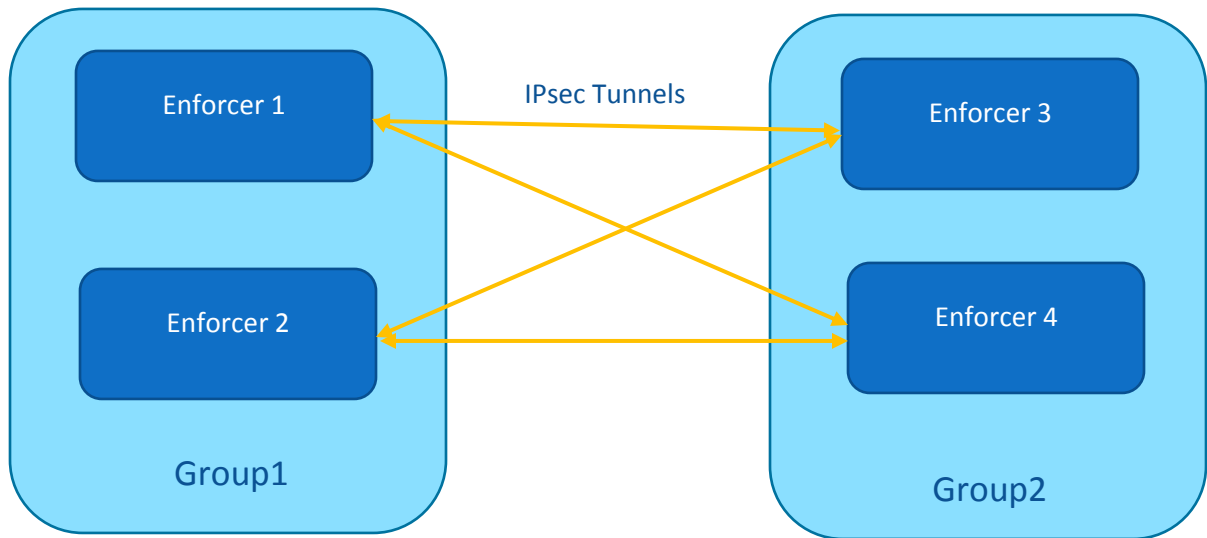
Certificate manages the public certificates and private key entities when establishing secure tunneling between them. Create Certificate for VPN Endpoints by uploading Certificate and Key.



1.1.4 IPSEC Enforcer

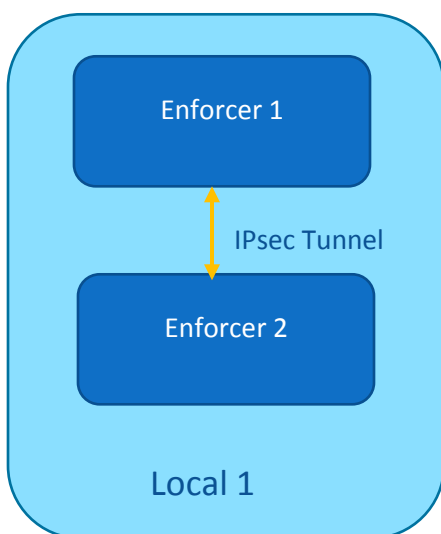
IPsec Enforcer service manages IPsec configurations in entities who need IPsec tunnel for secure communications. This service fetches IPsec configurations from IPsec EMS using RESTful API's exposed by IPsec EMS.

IPsec enforcer supports topology creation using concept of groups. Enforcer in one group can create IPsec tunnels with enforcers in other groups. Like in below image **Enforcer 1 and Enforcer 2** are in **group 1** and **Enforcer 3 and Enforcer 4** are in **group 2**. So if create IPsec tunnels between groups, it will automatically create IPsec tunnels between individual enforcers.



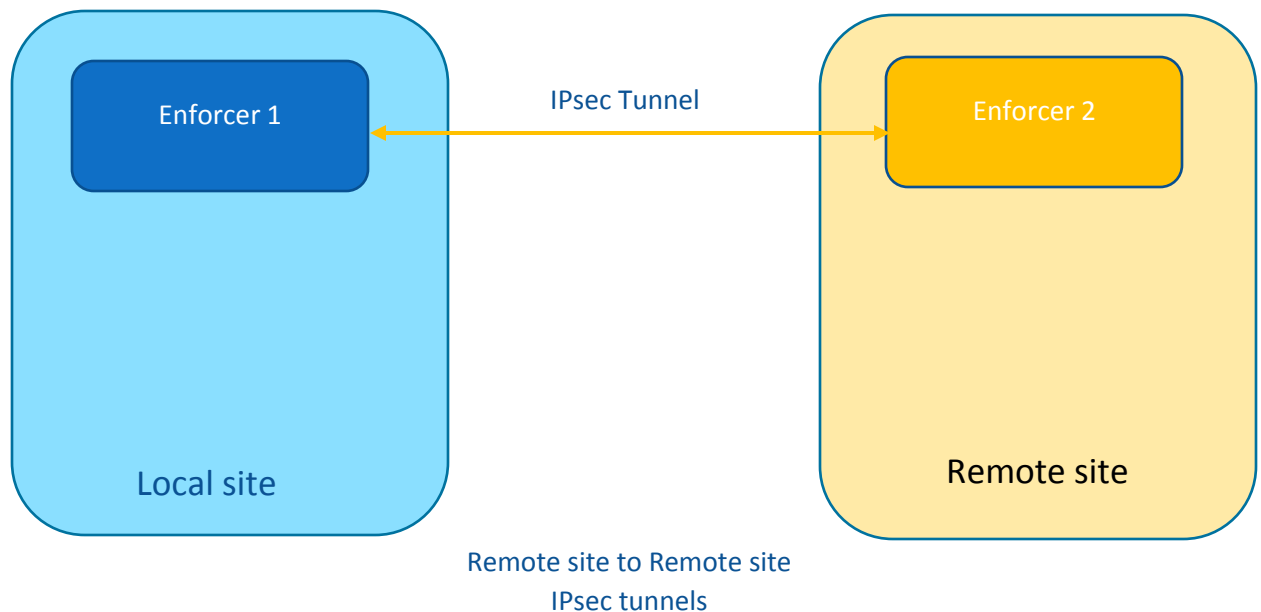
Group to Group IPsec tunnels

IPsec enforcer supports IPsec tunnel creation between enforcers present inside same network. For example in below image **Enforcer 1 and Enforcer 2** both are part of **local 1** local site.



Local site to Local site IPsec tunnels

IPsec enforcer supports IPsec tunnel creation between enforcers present in different networks. For example in below image **Enforcer 1** and **Enforcer 2** are part of **local 1** local site and **remote 1** remote site respectively.



Steps to Run IPsec Enforcer are given below:

1. Run **dcg_networksecurity_ipsec-ems/enforcer/install.sh** bash script.
2. Choose operations sequentially [Install Dependencies, Configure IPsec Enforcer and Run IPsec Enforcer].
3. **Install Dependencies** will install all the required dependencies to run IPsec Enforcer. This need to be done only once.
4. **Configure IPsec Enforcer** will ask for configurations according to which IPsec configurations are fetched from IPsec EMS server and IPsec tunnel created between enforcers. If your configurations are not changing then this step also required only once.
 - a. **Enter group name** is name of group according to which n-m IPsec tunnel will be created between enforcers as shown above in images.
 - b. **Enter localsite name** is name of localsite according to which IPsec tunnel will be created between enforcers.
 - c. **Enter Tunnel Interface** is interface used to create IPsec tunnel between IPsec enforcers.
 - d. **Enter IPsec EMS Controller address with port** is IP address with port of IPsec EMS server from which IPsec Enforcers fetch configurations.
5. **Run IPsec Enforcer** will run IPsec EMS server in background.
6. Now run **strongswan** service to create IPsec tunnel between enforcers. IPsec Enforcer only populate ipsec.conf and ipsec.secrets file according to configurations mentioned in IPsec EMS server. After changing configurations in IPsec EMS server, restart strongswan service so that strongswan can use updated configurations.