Cisco APIC-EM Security

Information about Cisco APIC-EM Security

The Cisco APIC-EM requires a multi-layered architecture to support its basic functionality. This multi-layered architecture consists of the following components:

- External network or networks—The external network exists between administrators and applications on one side of the network, and the Grapevine root and clients within an internal network or cloud on the other side. Both administrators and applications access the Grapevine root and clients using this external network.
- Internal network—The internal network consists of both the Grapevine root and clients.
- Device management network—This network consists of the devices that are managed and monitored by the controller. Note that the device management network is essentially the same as the external network described above. This may be physically or logically segmented from the admins or northbound applications.

Important Any inter-communications between the layers and intra-communications within the layers are protected through encryption, authentication, and segmentation.

Note For information about the different services running on the clients within the internal network, see Chapter 4, *Cisco APIC-EM Services*.

External Network Security

The Cisco APIC-EM provides its service over HTTPS and presents its X.509 server public certificate to client communications arriving at any of the external interfaces (eth0, eth1, eth2, etc.). The external clients (for example, northbound REST API consumer applications, devices performing file downloads from the controller, DMVPN certificate renewal, or certificate revocation list (CRL), etc.) may reach the controller via a NAT, proxy gateway, or directly.

The external X.509 certificate that is presented by the controller is one that has been either dynamically generated and self-signed by the controller itself, or one that has been imported (user's X.509 certificate) with a private key into the controller using the GUI. You have the option to either use the a self-signed X.509 certificate from the controller or to import and use your own X.509 certificate and private key. By default, the self-signed X.509 certificate presented to an API request is signed by Grapevine's internal Certificate Authority (CA). This self-signed X.509 certificate may not be recognized and accepted by your host. To proceed with your API request, you must ignore any warning and trust the certificate to proceed.

Note We recommend against using and importing a self-signed certificate into the controller. Importing a valid X.509 certificate from a well-known, certificate authority (CA) is recommended.

Northbound REST API requests from the external network to the Cisco APIC-EM are made secure using the Transport Layer Security (TLS) protocol. Although the controller supports several TLS versions, the default setting for the controller is TLS, version 1.0. You can restrict TLS support to a later and more secure version using the CLI. For additional information, see Configuring the TLS Version Using the CLI, on page 15 .

Related Topics

Configuring the TLS Version Using the CLI, on page 15

Internal Network Security

Several key intra-Grapevine communications using HTTP are sent over SSL using the internal public key infrastructure (PKI). All the internal Grapevine services, database servers, and the Cisco APIC-EM services themselves listen only on the internal network in order to keep these services segmented and secured.

Note This PKI plane exists within the Cisco APIC-EM. This PKI plane is inaccessible to northbound REST API callers, such as third-party applications. For information about the other PKI planes, see Cisco APIC-EM PKI Planes, on page 4.

Related Topics

Configuring IPSec Tunneling for Multi-Host Communications, on page 17

Device Management Network Security

Device management network security involves both controller-initiated communications and device-initiated communications.

For controller-initiated communications (discovery or pushing policy to the devices), the Cisco APIC-EM uses the following protocols to access and program network devices:

- SSH version 2
- SNMP versions 2c and 3
- Telnet (disabled by default)

Note If supported by the network devices, we strongly recommend using SNMP version v3c with authentication and privacy enabled. The controller does not connect to devices that are SSH version 1. HTTP and HTTPS are not supported for device discovery by the controller.

For device-initiated communications, network devices can use the following protocols to communicate and interact with the controller:

- HTTP
- HTTPS
- SNMP versions 2c

The use of HTTP or HTTPS is not up to the device itself; it is determined by the NB REST API that the device is calling. HTTP is supported for less sensitive communications.

Related Topics

Configuring the TLS Version Using the CLI, on page 15

Information about PKI

The Cisco APIC-EM relies on Public Key Infrastructure (PKI) to provide secure communications. PKI consists of certificate authorities, digital certificates, and public and private keys.

Certificate authorities (CAs) manage certificate requests and issue digital certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate the hosts, devices and/or individual users. In public key cryptography, such as the RSA encryption system, each entity has a key pair that contains both a private key and a public key. The private key is kept secret and is known only to the owning host, device or user. However, the public key is known to everyone. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates link the digital signature to the sender. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The CA that signs the certificate is a third party that the receiver explicitly trusts to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Typically this process is handled out of band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default.

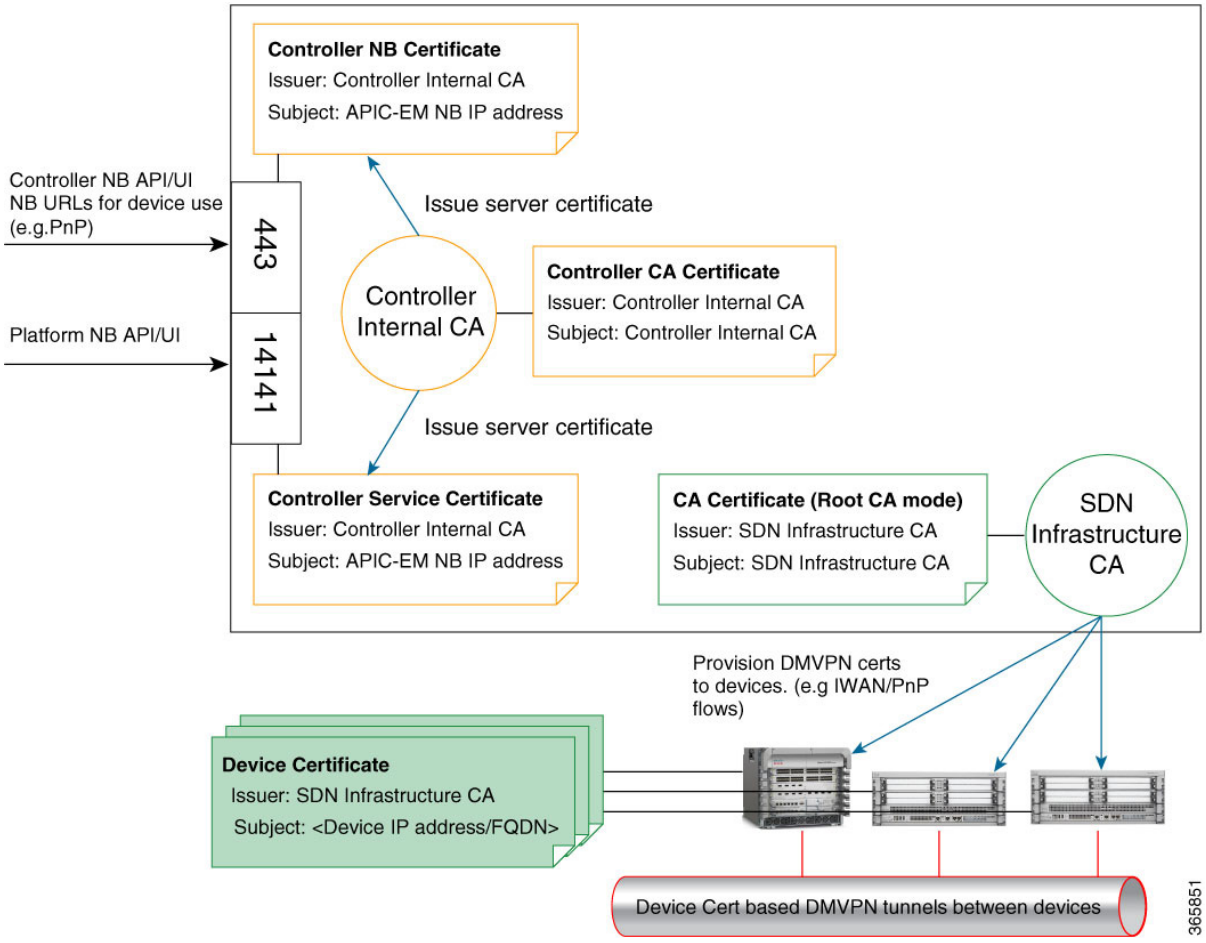
Cisco APIC-EM PKI Planes

The Cisco APIC-EM provides PKI-based connections in the following distinct PKI planes:

- Controller PKI Plane—HTTPS connections in which the controller is the server in the client-server model, and the controller's server certificate secures the connection. The controller's server certificate can be self-signed (default) or issued by an external CA (recommended.)
- Device PKI Plane—DMVPN connections between devices in the control plane of the network, bilaterally authenticated and secured by the device ID certificates of both devices that participate in the connection. A private CA provided by the Cisco APIC-EM controller (the Device PKI CA) manages these certificates and keys.
- Grapevine Service PKI Plane—The Grapevine root manages this internal PKI plane that secures communications between Grapevine services in a multi-host cluster; the Grapevine Service PKI Plane is not externally accessible, so it is not discussed further here.

The following is a schematic of the Cisco APIC-EM PKI planes, certificate authorities, and certificates. The Controller PKI Plane employs a Controller Internal CA that in response to external requests provides a Controller NB certificate and Controller CA certificate. The Grapevine PKI Plane employs the same Controller Internal CA that in response to internal requests (from controller services) provides a Controller Service Certificate. The Device PKI Plane employs a SDN Infrastructure CA that provides a CA Certificate (Root CA mode in this schematic) for IWAN and PnP devices.

Figure 1: Cisco APIC-EM PKI Planes



The Cisco APIC-EM PKI planes support different trust relationships or domains as displayed with the use cases in the following table:

Table 1: PKI Planes in Cisco APIC-EM

	Authentication	Encryption	Use Case
Controller PKI Plane: external caller initiates connection to controller			
HTTPS	Caller presents username and password or service ticket; Controller presents server certificate.	Yes	REST client, including Cisco Network Plug N Play (PnP) mobile app or Cisco Prime Infrastructure
HTTPS	One-way: controller presents its server certificate.	Yes	Cisco Network Plug N Play (PnP) provisioning workflow
Device PKI Plane: device-to-device connections			

	Authentication	Encryption	Use Case
DMVPN	Bilateral authentication via Internet Key Exchange Version 2 (IKEv2) using certificates/keys issued by a private CA within the Cisco APIC-EM controller.	Yes	DMVPN connections between devices

Note The security content and discussion in this deployment guide concerns itself primarily with the Controller PKI Plane. For information about the Device PKI Plane, see the *PKI Planes in Cisco APIC-EM Technote*.

Controller PKI Plane

When an external caller initiates an HTTPS connection to the controller, the controller presents its server certificate. Such connections include the following:

- Logins to the Cisco APIC-EM GUI via HTTPS
- Logins to the Grapevine APIs (port 14141) via HTTPS
- Invocations of the NB REST API via HTTPS

When a NB REST API caller initiates an HTTPS connection to the controller to invoke a NB REST API or to download a file (such as a device image, a configuration, and so on) the controller (server) presents its server certificate to the caller (client) that requested the connection.

Only two NB REST APIs use HTTP instead of HTTPS: the API that downloads the trustpool bundle (GET /ca/trustpool), and the API that downloads the controller's certificate (GET /ca/pem). All other NB REST APIs utilize HTTPS.

Note that controller-initiated connections to devices do NOT take place within the Controller PKI Plane. Even if the connections use SSH or SNMPv3, no CA manages the keys involved, so the connection is not considered to be PKI-based. The controller may initiate connections to devices for purposes that include discovery, managing tags, pushing policy to devices, or interacting with devices on behalf of a REST caller. For compatibility with older devices, discovery can optionally use the TELNET protocol, which is insecure and therefore outside the scope of this PKI discussion.

Device PKI Plane

IWAN-managed control-plane devices form Dynamic Multipoint VPN (DMVPN) connections among themselves. A private Certificate Authority (CA) provided by the Cisco APIC-EM (the Device PKI CA) provisions the certificates and keys that secure these DMVPN connections. The PKI broker service manages these certificates and keys as directed by an admin in the IWAN GUI or as directed by a REST caller that uses the /certificate-authority and /trust-point NB REST APIs.

Note In the default mode, the Device PKI CA in the Cisco APIC-EM cannot be a subordinate/intermediate CA to any external CA. These two PKI planes (one for the controller connections and the other for the device-to-device DMVPN connections) remain completely independent of each another. In the current release, the IWAN devices' mutual interaction certificates are managed only by the Device PKI CA. External CAs cannot manage the IWAN-specific certificates that devices present to each other for DMVPN tunnel-creation and related operations.

Device PKI Plane Modes

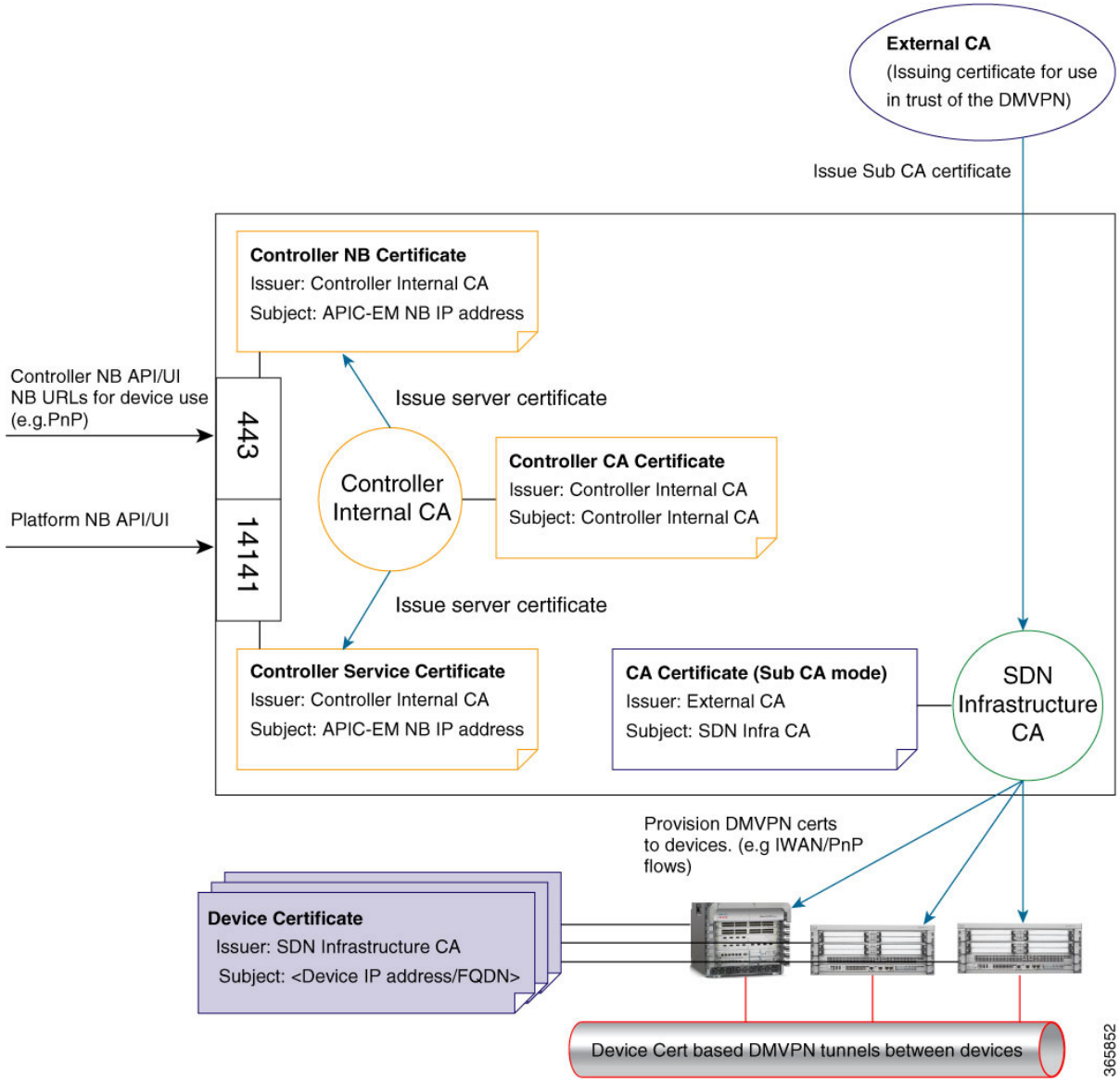
The Device PKI Plane supports two modes:

- Root mode-—The private CA provided by the Cisco APIC-EM controller does not interact with any other CA. This is the default mode for the controller.
- Sub CA mode —In Sub CA mode, the private CA provided by the Cisco APIC-EM controller can be an intermediary CA to an external CA. This means that the private controller CA still manages the certificates and keys that secure device-to-device communications, but it is in a subordinate position to that external CA. This mode must be enabled by an administrator (ROLE_ADMIN).

Changing the PKI mode from root to Sub CA (subordinate CA), changes the hierarchy and subordinates the private controller CA to an external CA. The following is a schematic of the distinct PKI planes, with the Device PKI plane being in Sub CA mode.

The following schematic displays the Sub CA mode for the Device PKI plane. In this schematic the Root CA is external to the controller. See Cisco APIC-EM PKI Planes, on page 4 for a schematic of Root CA mode for the Device PKI plane.

Figure 2: Device PKI Plane—Sub CA Mode



Related Topics

Changing the Role of the PKI Certificate from Root to Subordinate
Viewing the Device Certificate Lifetime

Device PKI Notifications

The Cisco APIC-EM provides device PKI notifications to assist the user with both troubleshooting and serviceability.

Important The device PKI notifications described in this section are only activated from device-to-device DMVPN connections and not the controller connections.

The following device PKI notifications are available:

- System Notifications—Notifications indicating that user action is required. These notifications are visible from the **Systems Notifications** view that is accessible from the **Global** toolbar in the GUI.
- Audit Log Notifications—Notifications in system logs that are visible using the controller's **Audit Log** GUI. For information about viewing the audit logs in the controller's GUI, see Viewing Audit Logs.

The following PKI *System* notification types are supported:

- Information
  - New trust point creation
  - New PKCS12 file creation
  - Successful enrollment of a device certificate
  - Successful renewal of a device certificate
  - Revocation of a device certificate
- Warning
  - Partial revocation—Device unreachable or trust point is in use
  - Enrollment delay after 80 percent of a certificate's lifetime
  - Service launch delay
- Critical
  - Certificate Authority handshake failed
  - Enrollment failed
  - Revocation failed
  - Renew failed

The following *audit log* notifications are available in the system logs:

- Device enrollment
- Certificate push to the device
- Renewal of a device certificate
- Revocation of a device certificate

Cisco APIC-EM Controller Certificate and Private Key Support

The Cisco APIC-EM supports a PKI certificate management feature (Controller PKI Plane) that is used to authenticate sessions (HTTPS). These sessions use commonly recognized trusted agents called certificate authorities (CAs). The Cisco APIC-EM uses the PKI certificate management feature to import, store, and manage an X.509 certificate from well-known CAs. The imported certificate becomes an identity certificate

for the controller itself, and the controller presents this certificate to its clients for authentication. The clients are the NB API applications and network devices.

The Cisco APIC-EM can import the following files (in either PEM or PKCS file format) using the controller's GUI:

- X.509 certificate
- Private key

Note For the private key, Cisco APIC-EM supports the importation of RSA keys. You should not import DSA, DH, ECDH, and ECDSA key types; they are not supported. You should also keep the private key secure in your own key management system.

Prior to import, you must obtain a valid X.509 certificate and private key from a well-known, certificate authority (CA) or create your own self-signed certificate. After import, the security functionality based upon the X.509 certificate and private key is automatically activated. The Cisco APIC-EM presents the certificate to any device or application that requests them. Both the northbound API applications and network devices can use these credentials to establish a trust relationship with the controller.

In an IWAN configuration and for the Network PnP functionality, an additional procedure involving a PKI trustpool is used to ensure trust between devices within the network. See the following *Cisco APIC-EM Trustpool Support* section for information about this procedure.

Note We recommend against using and importing a self-signed certificate into the controller. Importing a valid X.509 certificate from a well-known, certificate authority (CA) is recommended. Additionally, you must replace the self-signed certificate (installed in the Cisco APIC-EM by default) with a certificate that is signed by a well-known certificate authority for the Network PnP functionality to work properly.

The Cisco APIC-EM supports only one imported X.509 certificate and private key at a time. When you import a second certificate and private key, it overwrites the first (existing) imported certificate and private key values.

Note If the external IP address changes for your controller for any reason, then you need to re-import a new certificate with the changed or new IP address.

Related Topics

Importing the Controller's Server Certificate

Cisco APIC-EM Controller Certificate Chain Support

The Cisco APIC-EM is able to import certificates and private keys into the controller through its GUI.

If there are subordinate certificates involved in the certificate chain leading to the certificate that is imported into the controller (controller certificate), then both the subordinate certificates as well as the root certificate of these subordinate CAs must be appended together into a single file to be imported. When appending these certificates, you must append them in the same order as the actual chain of certification.

For example, assume that a well-known and trusted CA with a root certificate (CA root) signed an intermediate CA certificate (CA1). Next, assume that this certificate, CA1 signs another intermediate CA certificate (CA2). Finally, assume that the CA certificate (CA2) was the CA that signed the controller certificate (Controller_Certificate). In this example, the PEM file that needs to be created and imported into the controller should have the following order from the top (beginning) of the file to the bottom of the file (end):

1. Controller_Certificate (top of file)
2. CA2 certificate
3. CA1 certificate

The requirement to append the root and subordinate certificates to the controller certificate to create a single file only applies to a PEM file. The requirement for appending a root and intermediate certificates to a root certificate for import is not required for a PKCS file.

Related Topics

Importing the Controller's Server Certificate

Obtaining a CA-Signed Certificate for the Cisco APIC-EM Controller

You can perform the following steps to obtain a CA signed certificate to import into and use for the Cisco APIC-EM.

1. Determine the IP address or DNS-resolvable FQDN of your Cisco APIC-EM cluster.
2. Use that IP address as the common name in your certificate signing request (CSR).
3. Follow the procedure described below to create the CSR.
4. Send the completed CSR to the certificate authority (CA) that you have selected.
5. Receive the signed certificate back from the CA.
6. Install the certificate into the controller using the controller's GUI.

Note This example procedure is performed on the host where the Cisco APIC-EM is installed. You can also perform this procedure to generate a CSR and private key on a Linux OS or Apple Macintosh computer. You do not have to perform this procedure on the host where the Cisco APIC-EM is installed.

Before you begin

Before you attempt this procedure, you should have knowledge of these topics:

- How to use the OpenSSL application
- Public key infrastructure and digital certificates

Step 1 Using a Secure Shell (SSH) client, log into the host (physical or virtual) with the IP address that you specified using the configuration wizard.

The IP address to enter for the SSH client is the IP address that you configured for the network adapter. This IP address connects the host to the external network.

Step 2 When prompted, enter your Linux username ('grapevine') and password for SSH access.

Step 3 Enter the following command to create a private key and a CSR.

```
$ openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key

Generating a 2048 bit RSA private key
.....................................................+++
........+++
writing new private key to 'privateKey.key'
-----
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.
-----
```

Step 4 Respond to the certificate prompts with customer specific information as needed.

For the common name IP address, if this request is for multi-host Cisco APIC-EM deployment, then enter the Virtual IP address planned for the multi-host. If this request is for a single Cisco APIC-EM appliance or VM, then enter the eth0 IP address.

For example:

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:San Jose
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cisco
Organizational Unit Name (eg, section) []:Cloud Unit
Common Name (e.g. server FQDN or YOUR name) []:209.165.201.22
Email Address []:myemail@email.com
```

Step 5 Do not enter values for the extra attributes fields, just press **Enter**.

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:

An optional company name []:
```

After pressing **Enter**, two files (CSR and private key) will be generated.

Step 6 Locate the two files (CSR and private key) that were generated on the host.

The two files are: `privateKey.key` and `CSR.csr`.

For example, information about the files is displayed using the following command:

```
$ ls -ltr
total 8

-rw-rw-r-- 1 grapevine grapevine 1708 Apr 18 15:39 privateKey.key
-rw-rw-r-- 1 grapevine grapevine 1054 Apr 18 15:39 CSR.csr
```

Step 7 Secure the privateKey.key file.

Note Never send out the private key. Keep it in a secure location in your network.

Step 8 Copy and paste the CSR content from the CSR.csr file and send it to the CA for signing.

Note The CA will usually be a trustpool CA, unless your company runs its own CA.

In this example, the content in bold below will be the CSR that is copied and sent to the CA for signing and to get the certificate sent back.

```
$ cat CSR.csr

-----BEGIN CERTIFICATE REQUEST-----
MIIC0jCCAboCAQAwgYwxCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJDQTERMA8GA1UE
MRYwFAYDVQQDDA0xNzIuMjQuMTAwLjU1MSAwHgYJKoZIhvcNAQkBFhFteWVtYWls
QGVtYWlsLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAONJ7M96
rXjg/kwWcfJUlJJG2agLv7EAIxaB7He84fSdNMVXsJmuYBwZBWuZ9t/h3AKs/n/t
MRYwFAYDVQQDDA0xNzIuMjQuMTAwLjU1MSAwHgYJKoZIhvcNAQkBFhFteWVtYWls
QGVtYWlsLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAONJ7M96
rXjg/kwWcfJUlJJG2agLv7EAIxaB7He84fSdNMVXsJmuYBwZBWuZ9t/h3AKs/n/t
87nugrgW7SmI4F1wLsVg8KU2X0bmHoke6yCkhCPykQXJR2blMWp/OBc0ASMTIdhH
XRju1y/5
-----END CERTIFICATE REQUEST-----
(grapevine)
```

Important It is likely that instead of a single root CA certificate being sent back to you, that a chain of CA certificates, (including the CA's own public root certificate) will be sent back to you. In this case, follow the rules of appending the CA certificates as described in Cisco APIC-EM Controller Certificate Chain Support, on page 10, before importing them into the controller using its GUI.

Step 9 Once the CA administrator in your organization provides you with the signed certificate (for example, MyCert.pem), drag and drop the **MyCert.pem** and **privateKey.key** into the Cisco APIC-EM GUI certificate page. For information about this procedure, see Importing the Controller's Server Certificate

Note The content of MyCert.pem file obtained from the CA administrator should look like the CSR content which is base64 encoded and be in PEM format. Run the **cat** command on the obtained file to view its contents. If the file's contents looks like a binary file in the **cat** command output, then use the converter at this link to convert the file's content into PEM format:

https://www.sslshopper.com/ssl-converter.html.

Related Topics

Importing the Controller's Server Certificate

Cisco APIC-EM Trustpool Support

The Cisco APIC-EM and Cisco IOS devices support a special PKI certificate store known as the trustpool. The trustpool holds X.509 certificates that identify trusted certificate authorities (CAs). The Cisco APIC-EM and the devices in the network use the trustpool bundle to manage trust relationships with each other and with these CAs. The controller manages this PKI certificate store and an administrator (ROLE_ADMIN) has the ability to update it through the controller's GUI when certificates in the pool are due to expire, are reissued, or must be changed for other reasons.

Note The Cisco APIC-EM also uses the trustpool functionality to determine whether any certificate file that is uploaded via its GUI is a valid trustpool CA-signed certificate or not.

The Cisco APIC-EM contains a pre-installed, default, Cisco-signed trustpool bundle named ios.p7b. This trustpool bundle is trusted by supported Cisco network devices natively, since it is signed with a Cisco digital signing certificate. This trustpool bundle is critical for the Cisco network devices to establish trust with services and applications that are genuine. This Cisco PKI trustpool bundle file is available on the Cisco website (Cisco InfoSec).

The link is located at:http://www.cisco.com/security/pki/

For the controller's Network PnP functionality, the supported Cisco devices that are being managed and monitored by the controller need to import this file. When the supported Cisco devices first boot-up, they contact the controller to import this file.

The Cisco APIC-EM trustpool management feature operates in the following way:

1. You boot-up the Cisco devices within your network that supports the Network PnP functionality.

   Note that **not** all Cisco devices support the Network PnP functionality. See the *Release Notes for Cisco Network Plug and Play* for a list of the supported Cisco devices.

2. As part of initial PnP flow, these supported Cisco devices download a trustpool bundle directly from the Cisco APIC-EM using HTTP.

3. The Cisco devices are now ready to interact with the Cisco APIC-EM to obtain further device configuration and provisioning per the Network PnP traffic flows.

Important If an HTTP proxy gateway exists between the controller and these Cisco devices, then perform an additional procedure to import the proxy gateway certificate into the controller. See Importing a Proxy Gateway Certificate.

Note At times, you may need to update this trustpool bundle to a newer version due to certificates in the trustpool expiring, being reissued, or for other reasons. Whenever the trustpool bundle that exists on the controller needs to be updated, you can update it by using the controller's GUI. The controller can access the Cisco cloud (where the Cisco approved trustpool bundles are located) and download the latest trustpool bundle. After download, the controller then overwrites the current, older trustpool bundle file. As a practice, you may want to update the trustpool bundle before a new certificate from a CA is to be imported using the **Certificate** window or the **Proxy Gateway Certificate** window, or whenever the **Update** button is active and not grayed out.

Related Topics

Importing a Trustpool Bundle

Security and Cisco Network Plug and Play

With the Cisco Network Plug and Play (PnP) application, the Cisco APIC-EM responds to HTTPS requests from supported Cisco network devices and permits these devices to download and install an image and desired configuration. Before a device can download these files from the controller, the initial interaction between the controller and device involves the establishment of a trust relationship.

In certain Cisco Network Plug and Play scenarios, your network configuration may also have a proxy gateway present between the controller and PnP-enabled devices. For example, in an IWAN deployment a branch router may communicate with the Cisco APIC-EM through a proxy gateway at the DMZ at initial provisioning. Depending upon whether there is a proxy gateway present or not, the trust information provided by the controller at the initial transaction with the devices may correspond to either the proxy gateway's or to the controller's certificate issuer (if the corresponding server certificates are not valid CA signed). On the other hand, in either proxy or non-proxy cases, if the certificate is a simple self-signed certificate, then that certificate will be downloaded by the device into its trust store.

Note Using a self-signed certificate for either the Cisco APIC-EM or the proxy gateway is strongly discouraged. We strongly recommend using a publicly verifiable CA issued certificate to be installed on the controller, as well as the proxy gateway if one is present.

With a valid CA issued certificate for the controller or the proxy gateway (if present), the PnP-enabled devices can download the trustpool bundle (ios.p7b) containing all the well known CA root certificates. This permits the devices to establish secure connections to the controller or to the proxy gateway for further provisioning and operation of those devices. If such a certificate is not a valid CA issued or self-signed, then the devices will have to download the issuing CA's or self-signed certificate to proceed further with a secure connection to the controller or a proxy gateway in front of the controller. The Cisco APIC-EM facilitates automatic downloads of the relevant trusted certificates on the devices, depending on the nature of the certificate installed on it. However; when a proxy gateway is present, the controller provides a provisioning GUI to facilitate similar pre-provisioning.

Related Topics

Importing a Proxy Gateway Certificate

Configuring the TLS Version Using the CLI

Northbound REST API requests from the external network to the Cisco APIC-EM (from northbound REST API based apps, browsers, and network devices connecting to the controller using HTTPS) are made secure using the Transport Layer Security protocol (TLS). The Cisco APIC-EM supports TLS versions 1.0, 1.1, and 1.2.

By default, the minimum TLS version that a client can use to communicate with the controller is version 1.0. If your network device IOS/XE versions can support a higher version than version 1.0, then it is strongly recommended to configure the minimum TLS version of the controller to that higher version, but first ensure that all of your network devices under Cisco APIC-EM control can support the higher version.

Important With the controller TLS version set to 1.2, a client initiating a lower TLS connection version (for example, versions 1.0 or 1.1) will be rejected and any communications from this client will fail. With the controller TLS version set to 1.0, a client initiating a higher TLS connection version (for example, versions 1.1 or 1.2) will be permitted. Any versions lower than TLS 1.0 (such as SSLv3 and SSLv2) are not supported by the Cisco APIC-EM.

You configure the TLS version for the controller by logging into the host (physical or virtual) and using the CLI.

Before you begin

You must have successfully deployed the Cisco APIC-EM and it must be operational.

You must have grapevine SSH access privileges to perform this procedure.

Important This security feature applies to ports 443 and 14141 on the Cisco APIC-EM. Performing this procedure may disable traffic on port 14141 to the controller infrastructure for a few seconds. For this reason, you should configure TLS infrequently and only during off-peak hours or a maintenance time period.

Step 1 Using a Secure Shell (SSH) client, log into the host (physical or virtual) with the IP address that you specified using the configuration wizard.

Note The IP address to enter for the SSH client is the IP address that you configured for the network adapter. This IP address connects the host to the external network.

Step 2 When prompted, enter your Linux username ('grapevine') and password for SSH access.

Step 3 Enter the **grape config display** command at the prompt to display the default TLS minimum version.

```
$ grape config display

PROPERTY                          VALUE
------------------------------------------------------------------------
client_grow_timeout               150
client_heartbeat_timeout          120
client_idle_timeout               60
enable_policy                     True
enable_secure_tunnel              True
enable_service_rollback           False
host_cpu_threshold                0.9
host_datastore_threshold          1.0
host_heartbeat_timeout            120
host_memory_threshold             0.00999999977648
https_proxy
https_proxy_password
https_proxy_username
load_multiplier                   1.0
max_spare_capacity                1
policy_startup_delay              120
tls_minimum                       1_0

(grapevine)
```

The above command output indicates that the current TLS minimum version is 1.0.

Step 4 Enter the **grape config update tls_minimum 1_2** command at the prompt to update to TLS version 1.2

```
$ grape config update tls_minimum 1_2
Config updated successfully

(grapevine)
```

To update the TLS version to 1.1, you would enter the **grape config update tls_mimimum 1_1** command.

Step 5 Enter the **grape config display** command at the prompt a second time to view the new TLS minimum version.

```
$ grape config display

PROPERTY                          VALUE
------------------------------------------------------------------------
client_grow_timeout               150
client_heartbeat_timeout          120
client_idle_timeout               60
enable_policy                     True
enable_secure_tunnel              True
enable_service_rollback           False
host_cpu_threshold                0.9
host_datastore_threshold          1.0
host_heartbeat_timeout            120
host_memory_threshold             0.00999999977648
https_proxy
https_proxy_password
https_proxy_username
load_multiplier                   1.0
max_spare_capacity                1
policy_startup_delay              120
tls_minimum                       1_2

(grapevine)
```

The TLS minimum version should display *1_2*, which indicates the TLS 1.2 version.

Related Topics

External Network Security, on page 2
Device Management Network Security , on page 2

Configuring IPSec Tunneling for Multi-Host Communications

The default tunneling protocol used for inter-host communications in a multi-host cluster is Internet Protocol Security (IPsec). The previous default tunneling protocol (in earlier controller release versions) was Generic Routing Encapsulation (GRE). Communications between the hosts in a multi-host cluster can be made more secure using IPSec. If you current tunneling configuration between hosts is GRE, then you can enable secure tunneling with IPSec with the configuration wizard.

Perform the steps described in the following procedure to enhance security for communications between the hosts. The steps are organized as follow:

1. Break up or disassemble your existing multi-host cluster (steps 1-6).

2. Enable IPSec tunneling on the last host that was in your cluster (steps 7-11).
3. Reassemble your multi-host cluster around that host where you enabled IPSec tunneling. (steps 11-21).

Note Do not enable or disable the secure tunnel mode (IPSec tunneling) while the Cisco APIC-EM is in a multi-host cluster. The configuration wizard does not support such a change while in a multi-host cluster.

Before you begin

You must have successfully deployed the Cisco APIC-EM and it must be operational.

The current tunneling protocol is GRE, and not IPSec.

You must have grapevine SSH access privileges to perform this procedure.

Step 1 Using a Secure Shell (SSH) client, log into one of the hosts in your cluster.

When prompted, enter your Linux username ('grapevine') and password for SSH access.

Step 2 Enter the **grape config display** command to view and confirm your current GRE tunneling configuration.

```
$ grape config display
```

The **enable_secure_tunnel** value will be set to **false** for a GRE configuration.

Step 3 Enter the following command to access the configuration wizard.

```
$ config_wizard
```

Step 4 Review the **Welcome to the APIC-EM Configuration Wizard!** screen and choose the option to remove the host from the cluster:

- **Remove this host from its APIC-EM cluster**

Step 5 A message appears with an option to **[proceed]** and remove this host from the cluster.

Choose **proceed>>** to begin. After choosing **proceed>>**, the configuration wizard begins to remove this host from the cluster.

At the end of this process, this host is removed from the cluster.

Step 6 Repeat the above steps (steps 1-4) on the second host in your cluster. This will break up your multi-host cluster.

Important Make a note of the final host in the cluster that you have just broken up or disassembled. You must perform the next steps (enabling IPSec tunneling) on that final host. For example, with 3 hosts in a cluster (A, B, and C) and you first remove host A, then remove host B, then you must enable IPSec on host C.

Step 7 Using a Secure Shell (SSH) client, log into the last host in your cluster and run the **config_wizard** command.

```
$ config_wizard
```

Step 8 Review the current configuration values in the configuration wizard and click **next>>**, until you access the **INTER-HOST COMMUNICATION** screen.

Step 9 Configure IPSec tunneling for communications between the hosts in a multi-host cluster by selecting *yes*.

By entering 'yes', you are configuring IPSec tunneling with this step.

Step 10 Click **next>>** until the last step of the configuration wizard process is reached.

Step 11 Click **proceed>>** to have the configuration wizard save and apply your configuration changes to your Cisco APIC-EM deployment.

At the end of the configuration process, a **CONFIGURATION SUCCEEDED!** message appears.

Next, proceed to log into the other hosts previously in your multi-host cluster and use the configuration wizard to reassemble the cluster (with IPSec tunneling configured between the hosts).

Step 12 Using a Secure Shell (SSH) client, log into one of the other hosts in your cluster.

When prompted, enter your Linux username ('grapevine') and password for SSH access.

Step 13 Enter the following command to access the configuration wizard.

```
$ config_wizard
```

Step 14 Review the **Welcome to the APIC-EM Configuration Wizard!** screen and choose the **Create a new APIC-EM cluster** option.

Note Joining this other (second) host to the host with the enabled IPSec tunneling, automatically configures IPSec tunneling on this other (second) host.

Step 15 Proceed to recreate the cluster using the configuration wizard.

For additional information about this step and process, see Configuring Cisco APIC-EM as a Multi-Host Cluster Using the Wizard.

Step 16 At the end of the configuration process, click **proceed>>** to have the configuration wizard save and apply your configuration changes.

A **CONFIGURATION SUCCEEDED!** message appears.

Step 17 Using a Secure Shell (SSH) client, log into the third host and use the configuration wizard to join the new multi-host cluster.

When prompted, enter your Linux username ('grapevine') and password for SSH access.

Step 18 Enter the following command to access the configuration wizard.

```
$ config_wizard
```

Step 19 Review the **Welcome to the APIC-EM Configuration Wizard!** screen and choose the **Add this host to an existing APIC-EM cluster** option.

Note Adding this host to the new multi-host cluster with the enabled IPSec tunneling, automatically configures IPSec tunneling on this host.

Step 20 Proceed to add this host to the cluster using the configuration wizard.

For additional information about this step and process, see Configuring Cisco APIC-EM as a Multi-Host Cluster Using the Wizard..

Step 21 At the end of the configuration process, click **proceed>>** to have the configuration wizard save and apply your configuration changes.

A **CONFIGURATION SUCCEEDED!** message appears.

At the end of this step, you have updated your cluster and configured IPSec tunneling.

Related Topics

Internal Network Security, on page 2

Password Requirements

The Cisco APIC-EM password policy governs password values in logins to the controller GUI, SSH logins to the Grapevine root, northbound API requests, and logins to the Grapevine console for troubleshooting. The Cisco APIC-EM rejects a password that does not conform to the password policy. If a password is rejected, the controller provides an error message that describes the reason for the rejection.

A new or changed password must meet the following criteria:

- Eight character minimum length.
- Does NOT contain a tab or a line break.
- Does contain characters from at least three of the following categories:
  - Uppercase alphabet
  - Lowercase alphabet
  - Numeral
  - Special characters

    Special characters include the space character or any of the following characters or character combinations:

```
! @ # $ % ^ & * ( ) - =  + _ { } [ ] \\ | ; : " ' , < . > ? /
:: #! ./ ;; >> << () **
```

For example, `Splunge!` is a valid password because it meets the eight-character minimum length, contains at least one uppercase alphabetic character, contains at least one lowercase alphabetic character, and contains at least one special character (`!`).

Related Topics

Configuring Password Policies

Cisco APIC-EM Ports Reference

The following tables list the Cisco APIC-EM ports that permit incoming traffic, as well as the Cisco APIC-EM ports that are used for outgoing traffic. You should ensure that these ports on the controller are open for both incoming and outgoing traffic flows.

Note Ensure that proper protections exist in your network for accessing ports 22 and 14141. For example, you can configure a proxy gateway or secure subnets to access these ports.

Table 2: Cisco APIC-EM Incoming Traffic Port Reference

Port Number	Permitted Traffic	Protocol (TCP or UDP)
22	SSH	TCP
67	bootps	UDP
80	HTTP	TCP
123	NTP	UDP
162	SNMP	UDP
443 [1]	HTTPS	TCP
500	ISAKMP In order for deploying multiple hosts across firewalls in certain deployments, the IPSec ISAKMP (Internet Security Association and Key Management Protocol) UDP port 500 has to be allowed to be traversed.	UDP
14141	Grapevine APIs	TCP
16026	SCEP	TCP

[1] You can configure the TLS version for this port using the Cisco APIC-EM. For more information, see Configuring the TLS Version Using the CLI, on page 15

Table 3: Cisco APIC-EM Outgoing Traffic Port Reference

Port Number	Permitted Traffic	Protocol (TCP or UDP)
22	SSH (to the network devices)	TCP
23	Telnet (to the network devices)	TCP

Port Number	Permitted Traffic	Protocol (TCP or UDP)
53	DNS	UDP
80	Port 80 may be used for an outgoing proxy configuration. Additionally, other common ports such as 8080 may also be used when a proxy is being configured by the Cisco APIC-EM configuration wizard (if a proxy is already in use for your network). **Note** To access Cisco supported certificates and trust pools, you can configure your network to allow for outgoing IP traffic from the controller to Cisco addresses at the following URL: http://www.cisco.com/security/pki/	TCP
123	NTP	UDP
161	SNMP agent	UDP
443 [2]	HTTPS	TCP
500	ISAKMP In order for deploying multiple hosts across firewalls in certain deployments, the IPSec ISAKMP ((Internet Security Association and Key Management Protocol) UDP port 500 has to be allowed to be traversed.	UDP

[2] You can configure the TLS version for this port using the Cisco APIC-EM. For more information, see Configuring the TLS Version Using the CLI, on page 15