User Access, Authentication, and Accounting

This chapter contains the following sections:

Access Rights Workflow Dependencies

The Cisco Application Centric Infrastructure (ACI) RBAC rules enable or restrict access to some or all of the fabric. For example, in order to configure a leaf switch for bare metal server access, the logged in administrator must have rights to the `infra` domain. By default, a tenant administrator does not have rights to the `infra` domain. In this case, a tenant administrator who plans to use a bare metal server connected to a leaf switch could not complete all the necessary steps to do so. The tenant administrator would have to coordinate with a fabric administrator who has rights to the `infra` domain. The fabric administrator would set up the switch configuration policies that the tenant administrator would use to deploy an application policy that uses the bare metal server attached to an ACI leaf switch.

User Access, Authorization, and Accounting

Application Policy Infrastructure Controller (APIC) policies manage the authentication, authorization, and accounting (AAA) functions of the Cisco Application Centric Infrastructure (ACI) fabric. The combination of user privileges, roles, and domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a granular fashion. These configurations can be implemented using the REST API, the CLI, or the GUI.

Note There is a known limitation where you cannot have more than 32 characters for the login domain name. In addition, the combined number of characters for the login domain name and the user name cannot exceed 64 characters.

Multiple Tenant Support

A core Application Policy Infrastructure Controller (APIC) internal data access control system provides multitenant isolation and prevents information privacy from being compromised across tenants. Read/write restrictions prevent any tenant from seeing any other tenant's configuration, statistics, faults, or event data. Unless the administrator assigns permissions to do so, tenants are restricted from reading fabric configuration, policies, statistics, faults, or events.

User Access: Roles, Privileges, and Security Domains

The APIC provides access according to a user's role through role-based access control (RBAC). An Cisco Application Centric Infrastructure (ACI) fabric user is associated with the following:

- A predefined or custom role, which is a set of one or more privileges assigned to a user
- A set of privileges, which determine the managed objects (MOs) to which the user has access
- For each role, a privilege type: no access, read-only, or read-write
- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access

Roles and Privileges

A privilege controls access to a particular function within the system. The ACI fabric manages access privileges at the managed object (MO) level. Every object holds a list of the privileges that can read from it and a list of the privileges that can write to it. All objects that correspond to a particular function will have the privilege for that function in its read or write list. Because an object might correspond to additional functions, its lists might contain multiple privileges. When a user is assigned a role that contains a privilege, the user is given read access to the associated objects whose read list specifies read access, and write access to those whose write list specifies write access.

As an example, 'fabric-equipment' is a privilege that controls access to all objects that correspond to equipment in the physical fabric. An object corresponding to equipment in the physical fabric, such as 'eqptBoard,' will have 'fabric-equipment' in its list of privileges. The 'eqptBoard' object allows read-only access for the 'fabric-equipment' privilege. When a user is assigned a role such as 'fabric-admin' that contains the privilege 'fabric-equipment,' the user will have access to those equipment objects, including read-only access to the 'eqptBoard' object.

Note Some roles contain other roles. For example, '-admin' roles such as tenant-admin, fabric-admin, access-admin are groupings of roles with the same base name. For example, ‘access-admin’ is a grouping of 'access-connectivity', 'access-equipment', 'access-protocol', and 'access-qos.' Similarly, tenant-admin is a grouping of roles with a ‘tenant‘ base, and fabric-admin is a grouping of roles with a ‘fabric‘ base.

The 'admin' role contains all privileges.

For more details about roles and privileges see APIC Roles and Privileges Matrix.

Security Domains

A security domain is a tag associated with a certain subtree in the ACI MIT object hierarchy. For example, the default tenant “common” has a domain tag `common`. Similarly, the special domain tag `all` includes the entire MIT object tree. An administrator can assign custom domain tags to the MIT object hierarchy. For example, an administrator could assign the “solar” domain tag to the tenant named solar. Within the MIT, only certain objects can be tagged as security domains. For example, a tenant can be tagged as a security domain but objects within a tenant cannot.

Note Security Domain password strength parameters can be configured by creating **Custom Conditions** or by selecting **Any Three Conditions** that are provided.

Creating a user and assigning a role to that user does not enable access rights. It is necessary to also assign the user to one or more security domains. By default, the ACI fabric includes two special pre-created domains:

- `All`—allows access to the entire MIT
- `Infra`— allows access to fabric infrastructure objects/subtrees, such as fabric access policies

Note For read operations to the managed objects that a user's credentials do not allow, a "DN/Class Not Found" error is returned, not "DN/Class Unauthorized to read." For write operations to a managed object that a user's credentials do not allow, an HTTP 401 Unauthorized error is returned. In the GUI, actions that a user's credentials do not allow, either they are not presented, or they are grayed out.

A set of predefined managed object classes can be associated with domains. These classes should not have overlapping containment. Examples of classes that support domain association are as follows:

- Layer 2 and Layer 3 network managed objects
- Network profiles (such as physical, Layer 2, Layer 3, management)
- QoS policies

When an object that can be associated with a domain is created, the user must assign domain(s) to the object within the limits of the user's access rights. Domain assignment can be modified at any time.

If a virtual machine management (VMM) domain is tagged as a security domain, the users contained in the security domain can access the correspondingly tagged VMM domain. For example, if a tenant named solar

is tagged with the security domain called sun and a VMM domain is also tagged with the security domain called sun, then users in the solar tenant can access the VMM domain according to their access rights.

Login Domains

A login domain defines the authentication domain for a user. Login domains can be set to the Local, LDAP, RADIUS, TACACS+, DUO, SAML, RSA, or OAuth 2 authentication mechanisms. When accessing the system from REST, the CLI, or the GUI, the APIC enables the user to select the correct authentication domain.

For example, in the REST scenario, the username is prefixed with a string so that the full login username looks as follows:

```
apic:<domain>\<username>
```

If accessing the system from the GUI, the APIC offers a drop-down list of domains for the user to select. If no `apic: domain` is specified, the default authentication domain servers are used to look up the username.

Starting in ACI version 1.0(2x), the login domain fallback of the APIC defaults local. If the default authentication is set to a non-local method and the console authentication method is also set to a non-local method and both non-local methods do not automatically fall back to local authentication, the APIC can still be accessed via local authentication.

To access the APIC fallback local authentication, use the following strings:

- From the GUI, use *apic:fallback\\username*.
- From the REST API, use *apic#fallback\\username*.

Note Do not change the fallback login domain. Doing so could result in being locked out of the system.

Creating Login Domain Using the GUI

Authentication by an external server for SAML and OAuth 2 is based on user group map rule information, in addition to the standard CiscoAVpair-based authentication.

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The login domain name, realm, and remote server provider are available to define the authentication domain for the user.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the Navigation pane, choose **Authentication**.

Step 3 In the Work pane, choose the **Login Domains** tab.

Step 4 Click the **Actions** button > **Create Login Domain**.

Step 5 In the **Create Login Domain** screen, in the **General** pane, specify the following:

- The user configured domain name.
- Description of the login domain.
- The realm to verify the identity of an entity (person or device) accessing the fabric devices.The options available in the **Realm** drop-down list are discussed here:
  - **a.** A RADIUS provider group for a group of remote servers supporting the RADIUS protocol for authentication.
  - **b.** A TACACS+ provider group for a group of remote servers supporting the TACACS+ protocol for authentication.
  - **c.** An LDAP provider group for a group of remote servers supporting the LDAP protocol for authentication.
  - **d.** A RSA provider group for a group of remote servers supporting the RSA protocol for authentication.
  - **e.** A SAML provider remote server supporting the SAML protocol for authentication.
  - **f.** An OAuth 2 provider remote server supporting the OAuth 2 protocol for authentication.

Note If LDAP, RADIUS, or TACACS+ is specified as the default security method and the associated provider group specified in this dialog is not available to provide authentication during a user login, fallback local authentication is not executed by the Cisco APIC server unless is specifically configured to do so.

If Cisco APIC requires proxy servers to reach identity providers, then configure the corresponding proxy addresses. The proxy setting configuration is found under **System** > **System Settings** > **Proxy Policy**. In the **Proxy Policy** pane, enter the required URL in the **HTTP URL** or **HTTPS URL** fields.

Step 6 Fill in the details for the displayed options. The displayed options are dynamic and based on the selected **Realm**.

When the selected **Realm** is RADIUS or LDAP, the following options are displayed:

- Select either **Default** or **Duo** for the **Realm Subtype** .
- In the **Settings** pane, click **Add RADIUS (or LDAP) Provider** to select or create a provider if you selected the **Default** option above. If you have selected the **Duo** option, click **Add RADIUS (or LDAP) Duo Provider** to select or create a provider.

When the selected **Realm** is TACACS+ or RSA, the following options are displayed:

- In the **Settings** pane, click **Add RSA (or TACACS+) Provider** to select or create a provider.

When the selected **Realm** is SAML or OAuth 2, the following options are displayed:

- In the **Settings** pane, click **Select SAML (or OAuth 2) Provider** to select or create a provider.
- For **SAML (or OAuth 2) Authorization Choice**, select either **CiscoAVPair** or **GroupMap**.

- When **CiscoAVPair** is selected, the authorization is based on the CiscoAVpair value/ string configured on the external authentication server. On receiving the CiscoAVPair value from external IDP, Cisco APIC assigns the privileges accordingly to the remote user.
- When **GroupMap** is selected, the authorization is based on the group information configured on the external authentication server. On receiving the user group information from the external IDP, Cisco APIC matches the user group name configured on Cisco APIC and assigns the privileges to the remote user accordingly.

Two additional parameters are required for authorization using **GroupMap**, they are:

- Enter the **Group Attribute**. The group attribute entered here should match the group attribute on the external authentication server. For SAML, the group attribute should match the name of the group assertion in the response sent by the SAML IdP server. For OAuth2, the group attribute should match the group claim in the JWT (JSON Web Token) sent by the OAuth2 server.

  `Example: memberOf (used in Active directory), Groups or groups (used in ping ID/Okta)`

  Also, for OAuth2, to receive group information from IDP properly, ensure corresponding scope is configured in the OAuth2 provider configuration. Example: `openid profile groups`.
- Add a **User Group Map Rule**, by clicking **Add User Group Map Rule** .

  In the **Add User Group Map Rule** window, enter the following details:

  a. In the **Name** field, enter a name for the user group map rule.

  b. In the **Description** field, enter a description.

  c. In the **User Group** field, enter the name of the user group to which the user belongs.

  Ensure that the user group entered here matches the user group on the external server. This is used by Cisco APIC to validate the authentication information received from the external server. Privileges are set based on the user group to which the user belongs.

  d. To set **User Privileges**, click **Add User Privileges**.

  e. To add a security domain, click **Select Security Domain** to choose a security domain from the displayed list.

  f. Click **Add Role** to select a role and associate a privilege type (read or write); click the tick mark to associate the privilege to the role.

  To add more roles, click **Add Role**, and associate privileges.

  g. Click **Add** (on **Add User Privileges** window).

  h. Click **Apply** (on the **Add User Group Map Rule** window).

Step 7 Click **Save** (on the **Create Login Domain** screen).

Creating a Provider

Use this procedure to create a provider for the authentication/authorization protocols.

Before you begin

The relevant prerequisites before creating a provider for an authentication/authorization protocol is discussed under the relevant protocol sections.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the Navigation pane, choose **Authentication**.

Step 3 In the Work pane, choose **Providers**.

Step 4 Click **Actions** > **Create Provider**.

Step 5 In the **Create Provider** screen that is displayed, enter the **Hostname/IP Address**, **Description**, and choose a **Realm** from the drop-down list. The choices available for **Realm** are:

- RADIUS
- TACACS+
- LDAP
- SAML
- RSA
- OAuth 2

The options for configuring a provider are dynamic and change as per the selected **Realm**. The options available for each **Realm** are discussed in detail, in the subsequent steps.

Step 6 (Optional) Applicable only for RADIUS: Choose a **Realm Subtype**. The options are **Default** or **Duo**. Specify the following:

- Password for the RADIUS server; enter the password again for confirmation.
- Click **Select Reachability EPG** to choose an endpoint group.
- The service port number for RADIUS. The range is 1 to 65535. The default value is 1812.
- The authentication protocol, options are **PAP**, **CHAP**, **MS-CHAP**. You will see this option only if you chose **Default** for the **Realm Subtype**.
- The timeout for communication with a RADIUS server. The range is from 0 to 60 seconds. The default is 5 seconds (Realm Subtype: Default); default is 30 seconds (Realm Subtype:Duo).
- The number of retries when contacting the RADIUS endpoint.
- Select the **Enabled** check box to enable periodic server monitoring; enter a user name and password for the same.

This step is for RADIUS provider configuration. You can now proceed to step 12.

Step 7 (Optional step; applicable only for TACACS+) Specify the following:

- Password for the TACACS+ server; enter the password again for confirmation.
- Click **Select Reachability EPG** to choose an endpoint group.

- The service port number for TACACS+. The range is 1 to 65535. The default value is 49.
- The authentication protocol, options are—PAP, CHAP, MS-CHAP.
- The timeout for communication with a TACACS+ server. The range is from 0 to 60 seconds. The default is 5 seconds.
- The number of retries when contacting the TACACS+ endpoint.
- Select the **Enabled** check box to enable periodic server monitoring; enter a user name and password for the same.

This step is for TACACS+ provider configuration. You can now proceed to step 12.

Step 8 (Optional step; applicable only for LDAP) Choose a Realm Subtype, options are— **Default** or **Duo**. Specify the following:

- The Root Distinguished Name (DN) of the LDAP directory.
- The LDAP Base DN, which is the container name and path in the LDAP server where the APIC searches for the remote user account. This is where the password is validated. Filter is used to locate the attribute that the APIC requests to use for the *Cisco AVPair*.
- Password for the LDAP server; enter the password again for confirmation.
- The service port number for LDAP. The range is 1 to 65535. The default value is 389.
- Click **Select Reachability EPG** to choose an endpoint group.
- The timeout for communication with an LDAP server. The range is from 0 to 60 seconds. The default is 30 seconds.
- The number of retries when contacting the LDAP endpoint.
- Select the **Enable** check box to enable SSL.
- SSL Certificate Validation Level. The options are:
  - Permissive—A debugging knob to help diagnose DUO LDAP SSL Certificate issues.
  - Strict—A level that should be used when in production.
- LDAP Attribute.
- Authentication Method. The options are:
  - LDAP Bind
  - Password Compare
- Filter Type. Filters are a key element in defining the criteria used to identify entries in search requests. Example: (cn=*), which means any entry that contains one or more *cn* values. The options are:
  - Default
  - Microsoft Active Directory
  - Custom

- LDAP Filter. This field is auto-filled based on the selected Filter Type (unless you have chosen the Custom option Filter Type). If you have chosen Default, the filter is `cn=Suserid`; if you have chosen Microsoft Active Directory, the filter is `sAMAccountName=Suserid`.
- Select the **Enabled** check box to enable periodic server monitoring; enter a user name and password for the same.

This step is for LDAP provider configuration. You can now proceed to step 12.

Step 9 (Optional step; applicable only for RSA) Specify the following:

- Password for the RSA server; enter the password again for confirmation.
- Click **Select Reachability EPG** to choose an endpoint group.
- The service port number for RSA. The range is 1 to 65535. The default value is 1812.
- The timeout for communication with a RSA server. The range is from 0 to 60 seconds. The default is 5 seconds.
- The number of retries when contacting the RSA endpoint.
- Select the **Enabled** check box to enable periodic server monitoring; enter a user name and password for the same.

This step is for RSA provider configuration. You can now proceed to step 12.

Step 10 (Optional step; applicable only for SAML) Specify the following:

- **Identity Provider** (IdP). The options are—ADFS, OKTA, PING IDENTITY.
- **Metadata URL provided by IDP**.

  In case of ADFS, **Metadata URL provided by IDP** is of the format, *https://<FQDN of ADFS>/FederationMetadata/2007-06/FederationMetadata.xml*.

  For Ping ID, copy the metadata URL link from the configuration section of the Ping ID server (under the SAML application).
- Enter the **IdP Entity ID** for the SAML-based service.
- Click **Select Certificate Authority** to select a certificate authority if IdP is signed by a private CA.
- GUI Redirect Banner. This can be a URL or a message. This information is displayed before the user is redirected to the Identity Provider login page for authentication.
- The timeout for communication with a SAML server. The range is from 0 to 60 seconds. The default is 5 seconds.
- Select the Signature Algorithm from the drop down list.
- Put a check in the **Enabled** check box, to enable all/any of these— Encrypted SAML Assertions, Assertions in SAML Response Signed, SAML Auth Requests Signed, SAML Response Message Signed.

This step is for SAML provider configuration. You can now proceed to step 12.

Step 11 (Optional step; applicable only for OAuth 2) Specify the following:

- Client ID—Client identifier of the APIC application on IdP.

- Client Secret for the APIC application. Enter the client secret again for confirmation.
- Username Claim. Username attribute in the token. Example: email, sub.
- Scope. List of OAuth 2 scopes. Example: "openid profile". To receive user group information, add the corresponding scope configured in the IdP provider. Example: "openid profile groups".
- Choose to **Enable** or **Disable** the OIDC Protocol.
- Put a check in the **Enabled** check box to Verify Token Signature.
- JWKS Endpoint. The JSON Web Key Sets (JWKS) to verify the token. This field is displayed only if you have enabled Verify Token Signature.
- Authorization Endpoint. The IdP endpoint authorization URL. Get the authorization endpoint from the IdP server. This field is displayed only when the OIDC protocol is disabled.
- Token Endpoint. The IdP endpoint token URL. Get the token endpoint from the IdP server. This field is displayed only when the OIDC protocol is disabled.
- Issuer URL. Get the issuer URL from the IdP server.This field is displayed only when the OIDC protocol is enabled.
- Click **Select Certificate Authority** to select a ceritificate authority if IdP is signed by a private CA.
- Click **Select Reachability EPG** to choose an endpoint group.
- The timeout for communication with an OAuth 2 server. The range is from 0 to 60 seconds. The default is 5 seconds.
- GUI Redirect Banner. This can be a URL or a message. This information is displayed before the user is redirected to the Identity Provider login page for authentication.

This step is for OAuth 2 provider configuration. You can now proceed to step 12.

Step 12 Click **Save**.

Configuring a Local User

In the initial configuration script, the admin account is configured and the admin is the only user when the system starts. The APIC supports a granular, role-based access control system where user accounts can be created with various roles including non-admin users with fewer privileges.

Configuring a Local User Using the GUI

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- As appropriate, the security domain(s) that the user will access are defined. For example, if the new user account will be restricted to accessing a tenant, the tenant domain is tagged accordingly.
- An APIC user account is available that will enable the following:

- Creating the TACACS+ provider.
- Creating the local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, click **Users**.

In the **Work** pane, verify that you are in the **Local Users** tab.

Step 3 In the **Work** pane, click the task icon drop-down list and select **Create Local User**.

Step 4 In the **STEP 1 > User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.

The login ID must meet the following guidelines:

- Must be unique within APIC.
- Must begin with a letter.
- Can contain between 1 and 32 characters.
- Can include alphanumeric characters, underscores, dashes, and dots.

After creating a user account, you cannot change the login ID. You must delete the user account and create a new one.

b) In the **Password** field, enter the password.

At the time a user sets their password, the APIC validates it against the following criteria:

- Minimum password length is 8 characters.
- Maximum password length is 64 characters.
- Has fewer than three consecutive repeated characters.
- Must have characters from at least three of the following characters types: lowercase, uppercase, digit, symbol.
- Does not use easily guessed passwords.
- Cannot be the username or the reverse of the username.
- Cannot be any variation of cisco, isco or any permutation of these characters or variants obtained by changing the capitalization of letters therein.

c) In the **Confirm Password** field, confirm the password.

d) (Optional) For Certificate based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.

e) Click **Next**.

Step 5 You can activate or deactivate the user account by using the **Account Status** control, and you can set an expiration date by using the **Account Expires** control.

Step 6 In the **STEP 2 > Security** dialog box, under **Security Domain**, choose the desired security domain for the user, and click **Next**.

Step 7 In the **STEP 3 > Roles** dialog box, perform the following actions:

a) Click the **+** to associate the user with a domain.
b) From the drop-down lists, choose a **Role Name** and a **Role Privilege Type** for the user.
c) click **Update**

You can provide read-only or read/write privileges.

Step 8 click **Finish**

Configuring SSH Public Key Authentication Using the GUI

Before you begin

- Create a local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.
- Generate a public key using the Unix command **ssh-keygen**.

  The default login domain must be set to **local**

Procedure

Step 1 On the menu bar, choose **Admin** > **Users** and confirm you are in the **Local** tab.

Step 2 In the **Work** pane, click the name of the user that you previously created.

A window is displayed on the right with information about the user.

Step 3 Click the **Details** icon,  , and the user details are displayed on a new screen.

Scroll down to see the SSH Authorization details.

Step 4 Click the **Edit** icon,  , and the **Edit Local User** screen is displayed. You can change the SSH details as required.

Note To create the SSH Private Key File for downloading to a remote location, in the menu bar, expand **Firmware** > **Download Tasks**.

Step 5 Click **Save**.

Configuring a Remote User

Instead of configuring local users, you can point the APIC at the centralized enterprise credential datacenter. The APIC supports Lightweight Directory Access Protocol (LDAP), active directory, RADIUS, and TACACS+.

Note When an APIC is in minority (disconnected from the cluster), remote logins can fail because the ACI is a distributed system and the user information is distributed across APICS. Local logins, however, continue to work because they are local to the APIC.

Starting with the 3.1(1) release, **Server Monitoring** can be configured through RADIUS, TACACS+, LDAP, and RSA to determine whether the respective AAA servers are alive or not. Server monitoring feature uses the respective protocol login to check for server aliveness. For example, a LDAP server will use ldap login and a Radius server will use radius login with server monitoring to determine server aliveness.

To configure a remote user authenticated through an external authentication provider, you must meet the following prerequisites:

- The DNS configuration should have already been resolved with the hostname of the RADIUS server.
- You must configure the management subnet.

AV Pair on the External Authentication Server

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+.

To configure a Cisco AV Pair on an external authentication server, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair format is as follows:

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)
```

Starting with Cisco APIC release 2.1, if no UNIX ID is provided in AV Pair, the APIC allocates the unique UNIX user ID internally.

Note The APIC Cisco AV-pair format is compatible and can co-exist with other Cisco AV-pair formats. APIC will pick up the first matching AV-pair from all the AV-pairs.

Starting with release 3.1(x), the AV Pair shell:domains=all//admin allows you to assign Read-only privileges to users and provide them access to the switches and run commands.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Security domain with only writeRoles:

```
shell:domains=domainA/writeRole1|writeRole2/
```

- Example 2: A Cisco AV Pair that contains a single Security domain with only readRoles:

```
shell:domains=domainA//readRole1|readRole2
```

Note The "/" character is a separator between writeRoles and readRoles per Security domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

An example configuration for an open RADIUS server (/etc/raddb/users) is as follows:

```
aaa-network-admin Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = all/aaa/read-all(16001)"
```

Configuring an AV Pair on the External Authentication Server

The numerical value within the parentheses in the attribute/value (AV) pair string is used as the UNIX user ID of the user who is logged in using Secure Shell (SSH) or telnet.

Note Beginning with the 6.0(2) release, telnet is not supported.

Procedure

Configure an AV pair on the external authentication server.

The Cisco AV pair definition is as follows (Cisco supports AV pairs with and without UNIX user IDs specified):

Example:

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(8101)

These are the boost regexes supported by APIC:
uid_regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$");
regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$");
```

The following is an example:

```
shell:domains = coke/tenant-admin/read-all,pepsi//read-all(16001)
```

Configuring APIC for TACACS+ Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The TACACS+ server host name or IP address, port, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the TACACS+ Provider.

For configuring a TACACS+ provider, see Creating a Provider , on page 6.

For toggling in-band or out-of-band management in the APIC GUI:

In the Navigation pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the Work Pane select either **inband** or **ooband**.

Step 2 Create the **Login Domain** for TACACS+.

For the detailed procedure, see Creating Login Domain Using the GUI, on page 4.

What to do next

This completes the APIC TACACS+ configuration steps. Next, if a RAIDUS server will also be used, configure the APIC for RADIUS. If only a TACACS+ server will be used, go to the ACS server configuration topic below.

Configuring APIC for RADIUS Access

Before you begin

- The ACI fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The RADIUS server host name or IP address, port, authorization protocol, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the RADIUS provider.

For configuring a RADIUS provider, see Creating a Provider , on page 6.

For toggling in-band or out-of-band management in the APIC GUI:

In the Navigation pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the Work Pane select either **inband** or **ooband**.

Step 2 Create the **Login Domain** for RADIUS.

For the detailed procedure, see Creating Login Domain Using the GUI, on page 4.

What to do next

This completes the APIC RADIUS configuration steps. Next, configure the RADIUS server.

Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the APIC

Before you begin

- The Cisco Secure Access Control Server (ACS) version 5.5 is installed and online.

Note ACS v5.5 was used to document these steps. Other versions of ACS might support this task but the GUI procedures might vary accordingly.

- The Cisco Application Policy Infrastructure Controller (Cisco APIC) RADIUS or TACACS+ keys are available (or keys for both if both will be configured).
- The APICs are installed and online; the APIC cluster is formed and healthy.
- The RADIUS or TACACS+ port, authorization protocol, and key are available.

Procedure

Step 1 Log in to the ACS server to configure the APIC as a client.

a) Navigate to **Network Resources** > **Network Devices Groups** > **Network Devices and AAA Clients**.

b) Specify the client name, the APIC in-band IP address, select the TACACS+ or RADIUS (or both) authentication options.

Note If the only RADIUS or TACACS+ authentication is needed, select only the needed option.

c) Specify the authentication details such as Shared Secret (key), and port as appropriate for the authentication option(s).

Note The **Shared Secret**(s) must match the APIC **Provider** key(s).

Step 2 Create the Identity Group.

a) Navigate to **Users and Identity Stores** > **Internal Groups** option.

b) Specify the **Name**, and **Parent Group** as appropriate.

Step 3 Map users to the Identity Group.

a) In the **Navigation** pane, click the **Users and Identity Stores** > **Internal Identity Stores** > **Users** option.
b) Specify the user **Name**, and **Identity Group** as appropriate.

Step 4 Create the Policy Element.

a) Navigate to the **Policy Elements** option.
b) For RADIUS, specify the Authorization and Permissions > Network Access > Authorization Profiles **Name**. For TACACS+, specify the Authorization and Permissions > Device Administration > Shell Profile **Name** as appropriate.
c) For RADIUS, specify the **Attribute** as `cisco-av-pair`, **Type** as string, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate. For TACACS+, specify the **Attribute** as `cisco-av-pair`, **Requirement** as Mandatory, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate.

   The syntax of the **Value** field determines whether write privileges are granted:

   - For read/write privileges, the syntax is `shell:domains = <domain>/<role>/`.
   - For read-only privileges, the syntax is `shell:domains = <domain>// <role>`.

   For example, if the *cisco-av-pair* has a value of `shell:domains = solar/admin/,common// read-all`, then `solar` is the security domain, `admin` is the role that gives write privileges to this user in the security domain called `solar`, `common` is the tenant common, and `read-all` is the role with read privileges that gives this user read privileges to all of the tenant common.

Step 5 Create a service selection rule.

a) For RADIUS, create a service selection rule to associate the Identity Group with the Policy Element by navigating to **Access Policies** > **Default Device Network Access Identity** > **Authorization** and specifying the rule **Name**, **Status**, and **Conditions** as appropriate, and **Add** the `Internal Users:UserIdentityGroup in ALL Groups:<identity group name>`.
b) For TACACS+, create a service selection rule to associate the Identity Group with the Shell Profile by navigating to **Access Policies** > **Default Device Admin Identity** > **Authorization**. Specify the rule **Name**, **Conditions**, and **Select** the **Shell Profile** as appropriate.

What to do next

Use the newly created RADIUS and TACACS+ users to log in to the APIC. Verify that the users have access to the correct APIC security domain according to the assigned RBAC roles and privileges. The users should not have access to items that have not been explicitly permitted. Read and write access rights should match those configured for that user.

Configuring Windows Server 2012 LDAP for APIC Access with Cisco AVPair

Before you begin

- First, configure the LDAP server, then configure the Cisco Application Policy Infrastructure Controller (Cisco APIC) for LDAP access.
- The Microsoft Windows Server 2012 is installed and online.
- The Microsoft Windows Server 2012 Server Manager ADSI Edit tool is installed. To install ADSI Edit, follow the instructions in the Windows Server 2012 Server Manager help.
- `CiscoAVPair` attribute specifications: Common Name = **`CiscoAVPair`**, LDAP Display Name = **`CiscoAVPair`**, Unique X500 Object ID = 1.3.6.1.4.1.9.22.1, Description = **`CiscoAVPair`**, Syntax = **`Case Sensitive String`**.

Note For LDAP configurations, best practice is to use **`CiscoAVPair`** as the attribute string. If customer faces the issue using Object ID 1.3.6.1.4.1.9.22.1, an additional Object ID 1.3.6.1.4.1.9.2742.1-5 can also be used in the LDAP server.

- A Microsoft Windows Server 2012 user account is available that will enable the following:
  - Running ADSI Edit to add the `CiscoAVPair` attribute to the Active Directory (AD) Schema.
  - Configuring an Active Directory LDAP user to have `CiscoAVPair` attribute permissions.
- Port 636 is required for configuring LDAP integration with SSL/TLS.

Procedure

Step 1 Log in to an Active Directory (AD) server as a domain administrator.

Step 2 Add the `CiscoAVPair` attribute to the AD schema.

a) Navigate to **Start** > **Run**, type **`mmc`** and press **Enter**.
The Microsoft Management Console (MMC) opens.
b) Navigate to **File** > **Add/Remove Sanp-in** > **Add**.
c) In the **Add Standalonee Snap-in** dialog box, select the **Active Directory Schema** and click **Add**.
The MMC **Console** opens.
d) Right-click the **Attributes** folder, select the **Create Attribute** option.
The **Create New Attribute** dialog box opens.
e) Enter **`CiscoAVPair`** for the **Common Name** , **`CiscoAVPair`** for the **LDAP Display Name**, **`1.3.6.1.4.1.9.22.1`** for the **Unique X500 Object ID**, and select **`Case Sensitive String`** for the **Syntax**.
f) Click **OK** to save the attribute.

Step 3 Update the **User Properties** class to include the **CiscoAVPair** attribute.

a) In the MMC **Console**, expand the **Classes** folder, right-click the **user** class, and choose **Properties**. The **user Properties** dialog box opens.
b) Click the **Attributes** tab, and click **Add** to open the **Select Schema Object** window.
c) In the **Select a schema object:** list, choose **`CiscoAVPair`**, and click **Apply**.
d) In the MMC **Console**, right-click the **Active Directory Schema**, and select **Reload the Schema**.

Step 4 Configure the **`CiscoAVPair`** attribute permissions.

Now that the LDAP includes the **`CiscoAVPair`** attributes, LDAP users need to be granted Cisco APIC permission by assigning them Cisco APIC RBAC roles.

a) In the ADSI Edit dialog box, locate a user who needs access to the Cisco APIC.
b) Right-click on the user name, and choose **Properties**. The **<user> Properties** dialog box opens.
c) Click the **Attribute Editor** tab, select the *CiscoAVPair* attribute, and enter the *Value* as **`shell:domains = <domain>/<role>/,<domain>// role`**.

   For example, if the *CiscoAVPair* has a value of `shell:domains = solar/admin/,common// read-all(16001)`, then `solar` is the security domain, `admin` is the role for this user that gives write privileges to this user in the security domain called `solar`, `common` is the Cisco Application Centric Infrastructure (Cisco ACI) tenant common, and `read-all(16001)` is the role with read privileges that gives this user read privileges to all of the Cisco ACI tenant common.

d) Click **OK** to save the changes and close the **<user> Properties** dialog box.

The LDAP server is configured to access the Cisco APIC.

What to do next

Configure the Cisco APIC for LDAP access.

Configuring APIC for LDAP Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The LDAP server host name or IP address, port, bind DN, Base DN, and password are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, configure the LDAP Provider.

For configuring an LDAP provider, see Creating a Provider , on page 6.

For toggling in-band or out-of-band management in the APIC GUI:

In the Navigation pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the Work Pane select either **inband** or **ooband**.

Step 2 Create the **Login Domain** for LDAP.

For the detailed procedure, see Creating Login Domain Using the GUI, on page 4.

What to do next

This completes the APIC LDAP configuration steps. Next, test the APIC LDAP login access.

Changing the Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs

Procedure

Step 1 On the menu bar, choose **Admin** > **Authentication** > **AAA** > **Policy** tab.

Step 2 From the **Remote user login policy** drop-down list, choose **Assign Default Role**.

The default value is **No Login**. The **Assign Default Role** option assigns the minimal read-only privileges to users that have missing or bad Cisco AV Pairs. Bad AV Pairs are those AV Pairs that fail the parsing rules.

About Signature-Based Transactions

The APIC controllers in a Cisco ACI fabric offer different methods to authenticate users.

The primary authentication method uses a username and password and the APIC REST API returns an authentication token that can be used for future access to the APIC. This may be considered insecure in a situation where HTTPS is not available or enabled.

Another form of authentication that is offered utilizes a signature that is calculated for every transaction. The calculation of that signature uses a private key that must be kept secret in a secure location. When the APIC receives a request with a signature rather than a token, the APIC utilizes an X.509 certificate to verify the signature. In signature-based authentication, every transaction to the APIC must have a newly calculated signature. This is not a task that a user should do manually for each transaction. Ideally this function should be utilized by a script or an application that communicates with the APIC. This method is the most secure as it requires an attacker to crack the RSA/DSA key to forge or impersonate the user credentials.

Note Additionally, you must use HTTPS to prevent replay attacks.

Before you can use X.509 certificate-based signatures for authentication, verify that the following pre-requisite tasks are completed:

1. Create an X.509 certificate and private key using OpenSSL or a similar tool.
2. Create a local user on the APIC. (If a local user is already available, this task is optional).
3. Add the X.509 certificate to the local user on the APIC.

Guidelines and Limitations

Follow these guidelines and limitations:

- Local users are supported. Remote AAA users are not supported.
- The APIC GUI does not support the certificate authentication method.
- WebSockets and eventchannels do not work for X.509 requests.
- Certificates signed by a third party are not supported. Use a self-signed certificate.

Generating an X.509 Certificate and a Private Key

Procedure

Step 1 Enter an OpenSSL command to generate an X.509 certificate and private key.

Example:

```
$ openssl req -new -newkey rsa:1024 -days 36500 -nodes -x509 -keyout userabc.key -out
userabc.crt -subj '/CN=User ABC/O=Cisco Systems/C=US'
```

Note
- Once the X.509 certificate is generated, it will be added to the users profile on the APIC, and it is used to verify signatures. The private key is used by the client to generate the signatures.
- The certificate contains a public key but not the private key. The public key is the primary information used by the APIC to verify the calculated signature. The private key is never stored on the APIC. You must keep it secret.

Step 2 Display the fields in the certificate using OpenSSL.

Example:

```
$ openssl x509 -text -in userabc.crt
        Certificate:
            Data:
                Version: 3 (0x2)
                Serial Number:
                    c4:27:6c:4d:69:7c:d2:b6
                Signature Algorithm: sha1WithRSAEncryption
                Issuer: CN=User ABC, O=Cisco Systems, C=US
                Validity
                    Not Before: Jan 12 16:36:14 2015 GMT
                    Not After : Dec 19 16:36:14 2114 GMT
                Subject: CN=User ABC, O=Cisco Systems, C=US
                Subject Public Key Info:
                    Public Key Algorithm: rsaEncryption
```

```
                RSA Public Key: (1024 bit)
                    Modulus (1024 bit):
                        00:92:35:12:cd:2b:78:ef:9d:ca:0e:11:77:77:3a:
                        99:d3:25:42:94:b5:3e:8a:32:55:ce:e9:21:2a:ff:
                        e0:e4:22:58:6d:40:98:b1:0d:42:21:db:cd:44:26:
                        50:77:e5:fa:b6:10:57:d1:ec:95:e9:86:d7:3c:99:
                        ce:c4:7f:61:1d:3c:9e:ae:d8:88:be:80:a0:4a:90:
                        d2:22:e9:1b:25:27:cd:7d:f3:a5:8f:cf:16:a8:e1:
                        3a:3f:68:0b:9c:7c:cb:70:b9:c7:3f:e8:db:85:d8:
                        98:f6:e3:70:4e:47:e2:59:03:49:01:83:8e:50:4a:
                        5f:bc:35:d2:b1:07:be:ec:e1
                    Exponent: 65537 (0x10001)
            X509v3 extensions:
                X509v3 Subject Key Identifier:
                    0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                X509v3 Authority Key Identifier:
                    keyid:0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                    DirName:/CN=User ABC/O=Cisco Systems/C=US
                    serial:C4:27:6C:4D:69:7C:D2:B6

                X509v3 Basic Constraints:
                    CA:TRUE
        Signature Algorithm: sha1WithRSAEncryption
            8f:c4:9f:84:06:30:59:0c:d2:8a:09:96:a2:69:3d:cf:ef:79:
            91:ea:cd:ae:80:16:df:16:31:3b:69:89:f7:5a:24:1f:fd:9f:
            d1:d9:b2:02:41:01:b9:e9:8d:da:a8:4c:1e:e5:9b:3e:1d:65:
            84:ff:e8:ad:55:3e:90:a0:a2:fb:3e:3e:ef:c2:11:3d:1b:e6:
            f4:5e:d2:92:e8:24:61:43:59:ec:ea:d2:bb:c9:9a:7a:04:91:
            8e:91:bb:9d:33:d4:28:b5:13:ce:dc:fe:c3:e5:33:97:5d:37:
            cc:5f:ad:af:5a:aa:f4:a3:a8:50:66:7d:f4:fb:78:72:9d:56:
            91:2c
[snip]
```

Configuring a Local User

Creating a Local User and Adding a User Certificate Using the GUI

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users** and **Local Users** in the **Work** pane.

Step 3 In the **Work** pane, verify that you in the **Local Users** tab.

The admin user is present by default

Step 4 In the **Work** pane, click on task icon drop-down list and select **Create Local User**.

Step 5 In the **Security** dialog box, choose the desired security domain for the user, and click **Next**.

Step 6 In the **Roles** dialog box, click the radio buttons to choose the roles for your user, and click **Next**.

You can provide read-only or read/write privileges.

Step 7 In the **User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.

b) In the **Password** field, enter the password.
c) In the **Confirm Password** field, confirm the password.
d) (Optional) For Certificate based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.
e) Click **Finish**.

Step 8 In the **Navigation** pane, click the name of the user that you created. In the **Work** pane, expand the **+** sign next to your user in the **Security Domains** area.
The access privileges for your user are displayed.

Step 9 In the **Work** pane, in the **User Certificates** area, click the user certificates **+** sign, and in the **Create X509 Certificate** dialog box, perform the following actions:
a) In the **Name** field, enter a certificate name.
b) In the **Data** field, enter the user certificate details.
c) Click **Submit**.
The X509 certificate is created for the local user.

Creating a Local User Using Python SDK

Procedure

Create a local user.

Example:

```
#!/usr/bin/env python
from cobra.model.pol import Uni as PolUni
from cobra.model.aaa import UserEp as AaaUserEp
from cobra.model.aaa import User as AaaUser
from cobra.model.aaa import UserCert as AaaUserCert
from cobra.model.aaa import UserDomain as AaaUserDomain
from cobra.model.aaa import UserRole as AaaUserRole
from cobra.mit.access import MoDirectory
from cobra.mit.session import LoginSession
from cobra.internal.codec.jsoncodec import toJSONStr

APIC = 'http://10.10.10.1'
username = 'admin'
password = 'p@$$w0rd'

session = LoginSession(APIC, username, password)
modir = MoDirectory(session)
modir.login()

def readFile(fileName=None, mode="r"):
   if fileName is None:
      return ""
   fileData = ""
   with open(fileName, mode) as aFile:
      fileData = aFile.read()
   return fileData

# Use a dictionary to define the domain and a list of tuples to define
# our aaaUserRoles (roleName, privType)
# This can further be abstracted by doing a query to get the valid
```

```
# roles, that is what the GUI does

userRoles = {'all': [
               ('aaa', 'writePriv'),
               ('access-admin', 'writePriv'),
               ('admin', 'writePriv'),
               ('fabric-admin', 'writePriv'),
               ('nw-svc-admin', 'writePriv'),
               ('ops', 'writePriv'),
               ('read-all', 'writePriv'),
               ('tenant-admin', 'writePriv'),
               ('tenant-ext-admin', 'writePriv'),
               ('vmm-admin', 'writePriv'),
           ],
}

uni = PolUni('') # '' is the Dn string for topRoot
aaaUserEp = AaaUserEp(uni)
aaaUser = AaaUser(aaaUserEp, 'userabc', firstName='Adam',
                  email='userabc@cisco.com')

aaaUser.lastName = 'BC'
aaaUser.phone = '555-111-2222'
aaaUserCert = AaaUserCert(aaaUser, 'userabc.crt')
aaaUserCert.data = readFile("/tmp/userabc.crt")
# Now add each aaaUserRole to the aaaUserDomains which are added to the
# aaaUserCert
for domain,roles in userRoles.items():
   aaaUserDomain = AaaUserDomain(aaaUser, domain)
   for roleName, privType in roles:
       aaaUserRole = AaaUserRole(aaaUserDomain, roleName,
                                 privType=privType)
print toJSONStr(aaaUser, prettyPrint=True)

cr = ConfigRequest()
cr.addMo(aaaUser)
modir.commit(cr)
# End of Script to create a user
```

Using a Private Key to Calculate a Signature

Before you begin

You must have the following information available:

- HTTP method - GET, POST, DELETE
- REST API URI being requested, including any query options
- For POST requests, the actual payload being sent to the APIC
- The private key used to generate the X.509 certificate for the user
- The distinguished name for the user X.509 certificate on the APIC

Procedure

Step 1 Concatenate the HTTP method, REST API URI, and payload together in this order and save them to a file.

This concatenated data must be saved to a file for OpenSSL to calculate the signature. In this example, we use a filename of payload.txt. Remember that the private key is in a file called userabc.key.

Example:

GET example:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

POST example:

```
POST http://10.10.10.1/api/mo/tn-test.json{"fvTenant": {"attributes": {"status": "deleted",
 "name": "test"}}}
```

Step 2 Verify that the payload.txt file contains the correct information.

For example, using the GET example shown in the previous step:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

Your payload.txt file should contain only the following information:

```
GET/api/class/fvTenant.json?rsp-subtree=children
```

Step 3 Verify that you didn't inadvertently create a new line when you created the payload file.

Example:

```
# cat -e payload.txt
```

Determine if there is a **$** symbol at the end of the output, similar to the following:

```
GET/api/class/fvTenant.json?rsp=subtree=children$
```

If so, then that means that a new line was created when you created the payload file. To prevent creating a new line when generating the payload file, use a command similar to the following:

```
echo -n "GET/api/class/fvTenant.json?rsp-subtree=children" >payload.txt
```

Step 4 Calculate a signature using the private key and the payload file using OpenSSL.

Example:

```
openssl dgst -sha256 -sign userabc.key payload.txt > payload_sig.bin
```

The resulting file has the signature printed on multiple lines.

Step 5 Convert the signature to base64 format:

Example:

```
openssl base64 -A -in payload_sig.bin -out payload_sig.base64
```

Step 6 Strip the signature of the new lines using Bash.

Example:

```
$ tr -d '\n' < payload_sig.base64
P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8fIXXl4V79Zl7
Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f7q
IcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=
```

Note This is the signature that will be sent to the APIC for this specific request. Other requests will require to have their own signatures calculated.

Step 7 Place the signature inside a string to enable the APIC to verify the signature against the payload.

This complete signature is sent to the APIC as a cookie in the header of the request.

Example:

```
APIC-Request-Signature=P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8f
IXXl4V79Zl7Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f
7qIcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=;
APIC-Certificate-Algorithm=v1.0; APIC-Certificate-Fingerprint=fingerprint;
APIC-Certificate-DN=uni/userext/user-userabc/usercert-userabc.crt
```

Note The DN used here must match the DN of the user certified object containing the x509 certificate in the next step.

Step 8 Use the CertSession class in the Python SDK to communicate with an APIC using signatures.

The following script is an example of how to use the CertSession class in the ACI Python SDK to make requests to an APIC using signatures.

Example:

```
#!/usr/bin/env python
# It is assumed the user has the X.509 certificate already added to
# their local user configuration on the APIC
from cobra.mit.session import CertSession
from cobra.mit.access import MoDirectory

def readFile(fileName=None, mode="r"):
    if fileName is None:
        return ""
    fileData = ""
    with open(fileName, mode) as aFile:
        fileData = aFile.read()
    return fileData

pkey = readFile("/tmp/userabc.key")
csession = CertSession("https://ApicIPOrHostname/",
                       "uni/userext/user-userabc/usercert-userabc", pkey)

modir = MoDirectory(csession)
resp = modir.lookupByDn('uni/fabric')
pring resp.dn
# End of script
```

Note The DN used in the earlier step must match the DN of the user certified object containing the x509 certificate in this step.

Accounting

Cisco Application Centric Infrastructure (ACI) fabric accounting is handled by these two managed objects that are processed by the same mechanism as faults and events:

- The `aaaSessionLR` managed object tracks user account login and logout sessions on the Cisco Application Policy Infrastructure Controller (APIC) and switches, and token refresh. The Cisco ACI fabric session alert feature stores information such as the following:

- Username
- IP address initiating the session
- Type (telnet, HTTPS, REST, and so on)

Note Beginning with the 6.0(2) release, telnet is not supported.

- Session time and length
- Token refresh: A user account login event generates a valid active token which is required in order for the user account to exercise its rights in the Cisco ACI fabric.

Note Token expiration is independent of login; a user could log out but the token expires according to the duration of the timer value it contains.

- The `aaaModLR` managed object tracks the changes users make to objects and when the changes occurred.
- If the AAA server is not pingable, it is marked unavailable and a fault is seen.

Both the `aaaSessionLR` and `aaaModLR` event logs are stored in Cisco APIC shards. After the data exceeds the pre-set storage allocation size, it overwrites records on a first-in first-out basis.

Note In the event of a destructive event such as a disk crash or a fire that destroys a Cisco APIC cluster node, the event logs are lost; event logs are not replicated across the cluster.

The `aaaModLR` and `aaaSessionLR` managed objects can be queried by class or by distinguished name (DN). A class query provides all the log records for the whole fabric. All `aaaModLR` records for the whole fabric are available from the GUI at the **Fabric** > **Inventory** > **POD** > **History** > **Audit Log** section, The Cisco APIC GUI **History** > **Audit Log** options enable viewing event logs for a specific object identified in the GUI.

The standard syslog, callhome, REST query, and CLI export mechanisms are fully supported for `aaaModLR` and `aaaSessionLR` managed object query data. There is no default policy to export this data.

There are no pre-configured queries in the Cisco APIC that report on aggregations of data across a set of objects or for the entire system. A fabric administrator can configure export policies that periodically export `aaaModLR` and `aaaSessionLR` query data to a syslog server. Exported data can be archived periodically and used to generate custom reports from portions of the system or across the entire set of system logs.

Routed Connectivity to External Networks as a Shared Service Billing and Statistics

The Cisco Application Policy Infrastructure Controller (APIC) can be configured to collect byte count and packet count billing statistics from a port configured for routed connectivity to external networks as a shared service. The external networks are represented as external L3Out endpoint group (l3extInstP managed object)

in Cisco Application Centric Infrastructure (ACI). Any EPG in any tenant can share an external L3Out EPG for routed connectivity to external networks. Billing statistics can be collected for each EPG in any tenant that uses an external L3Out EPG as a shared service. The leaf switch where the external L3Out EPG is provisioned forwards the billing statistics to the Cisco APIC where they are aggregated. Accounting policies can be configured to export these billing statics periodically to a server.