TACACs+, RADIUS, LDAP, and SAML

This chapter contains the following sections:

Overview

This article provides step by step instructions on how to enable RADIUS, TACACS+, and LDAP users access the APIC. It assumes the reader is thoroughly familiar with the Cisco Application Centric Infrastructure Fundamentals manual, especially the User Access, Authentication, and Accounting chapter.

RADIUS

To configure users on RADIUS servers, the APIC administrator must configure the required attributes (`shell:domains`) using the `cisco-av-pair` attribute. The default user role is network-operator.

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If these options are not specified in the `cisco-av-pair` attribute, MD5 and DES are the default authentication protocols.

For example, SNMPv3 authentication and privacy protocol attributes can be specified as follows:

```
snmpv3:auth=SHA priv=AES-128
```

Similarly, the list of domains would be as follows:

```
shell:domains="domainA domainB …"
```

TACACS+ Authentication

Terminal Access Controller Access Control device Plus (TACACS+) is another remote AAA protocol that is supported by Cisco devices. TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the APIC can authorize access without authenticating.
- Uses TCP to send data between the AAA client and server, enabling reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. RADIUS encrypts passwords only.
- Uses the av-pairs that are syntactically and configurationally different than RADIUS but the APIC supports `shell:domains`.

Note The TACACS server and TACACs ports must be reachable by ping.

The XML example below configures the ACI fabric to work with a TACACS+ provider at IP address 10.193.208.9.

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

```
<aaaTacacsPlusProvider name="10.193.208.9"
           key="test123"
           authProtocol=”pap”/>
```

User IDs in the APIC Bash Shell

User IDs on the APIC for the Linux shell are generated within the APIC for local users. Users whose authentication credential is managed on external servers, the user ID for the Linux shell can be specified in the cisco-av-pair. Omitting the (16001) in the above cisco-av-pair is legal, in which case the remote user gets a default Linux user ID of 23999. Linux User IDs are used during bash sessions, allowing standard Linux permissions enforcement. Also, all managed objects created by a user are marked as created-by that user’s Linux user ID.

The following is an example of a user ID as seen in the APIC Bash shell:

```
admin@ifav17-ifc1:~> touch myfile
admin@ifav17-ifc1:~> ls -l myfile
-rw-rw-r-- 1 admin admin 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> ls -ln myfile
-rw-rw-r-- 1 15374 15374 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> id
uid=15374(admin) gid=15374(admin) groups=15374(admin)
```

Login Domains

A login domain defines the authentication domain for a user. Login domains can be set to the Local, LDAP, RADIUS, or TACACS+ 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.

LDAP/Active Directory Authentication

Similar to RADIUS and TACACS+, LDAP allows a network element to retrieve AAA credentials that can be used to authenticate and then authorize the user to perform certain actions. An added certificate authority configuration can be performed by an administrator to enable LDAPS (LDAP over SSL) trust and prevent man-in-the-middle attacks.

The XML example below configures the ACI fabric to work with an LDAP provider at IP address 10.30.12.128.

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

```
<aaaLdapProvider name="10.30.12.128"
          rootdn="CN=Manager,DC=ifc,DC=com"
          basedn="DC=ifc,DC=com"
          SSLValidationLevel="strict"
          attribute="AciCiscoAVPair"
          enableSSL="yes"
          filter="cn=$userid"
          port="636" />
```

Note For LDAP configurations, best practice is to use ciscoAVPair as the attribute string. This avoids problems related to the limitation LDAP servers not allowing overlapping object identifiers (OID); that is, the ciscoAVPair OID is already in use.

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.

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)
```

The first av-pair format has no UNIX user ID, while the second one does. Both are correct if all remote users have the same role and mutual file access is acceptable. If the UNIX user ID is not specified, ID 23999 is applied by the APIC system, and more than one role/read privilege is specified to any AV Pair user. This can cause users to have higher or lower permissions than configured through the group settings.

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.

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 Login domain with only writeRoles:

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

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

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

Note The "/" character is a separator between writeRoles and readRoles per Login 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)"
```

Best Practice for Assigning AV Pairs

As best practice, Cisco recommends that you assign unique UNIX user ids in the range 16000-23999 for the AV Pairs that are assigned to users when in bash shell (using SSH, Telnet or Serial/KVM consoles). If a situation arises when the Cisco AV Pair does not provide a UNIX user id, the user is assigned a user id of 23999 or similar number from the range that also enables the user's home directories, files, and processes accessible to remote users with a UNIX ID of 23999.

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.

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.

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**.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Providers**.
c) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider**.
d) Specify the TACACS+ host name (or IP address), port, authorization protocol, key, and management endpoint group.

Note If the APIC is configured for in-band management connectivity, choosing an out-of-band management endpoint group for TACACS+ access does not take effect. Alternatively, an out-of-band over an in-band management endpoint group can connect a TACACS+ server, but requires configuring a static route for the TACACS+ server. The Cisco ACS sample configuration procedure below uses an APIC in-band IP address.

Step 2 Create the **TACACS+ Provider Group**.

a) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider Group**.
c) Specify the TACACS+ provider group name, description, and providers as appropriate.

Step 3 Create the **Login Domain** for TACACS+.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

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.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **RADIUS Managment** > **RADIUS Providers**.
c) In the **Work** pane, choose **Actions** > **Create RADIUS Provider**.
d) Specify the RADIUS host name (or IP address), port, protocol, and management endpoint group.

Note If the APIC is configured for in-band management connectivity, choosing an out-of-band management endpoint group for RADIUS access does not take effect. Alternatively, an out-of-band over an in-band management endpoint group can connect a RADIUS server but requires configuring a static route for the RADIUS server. The Cisco ACS sample configuration procedure below uses an APIC in-band IP address.

Step 2 Create the RADIUS provider group.

a) In the **Navigation** pane, choose **RADIUS Managment** > **RADIUS Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create RADIUS Provider Group**.
c) Specify the RADIUS Provider Group name, description, and providers as appropriate.

Step 3 Create the login domain for RADIUS.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

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 Cisco APICs are installed and online; the Cisco 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 Cisco APIC as a client.

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

b) Specify the client name, the Cisco 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 Cisco 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.
For example, if the *cisco-av-pair* 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.

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 Cisco APIC. Verify that the users have access to the correct Cisco 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 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.

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.

Configuring a Remote User Using the NX-OS Style CLI

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+.

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.

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

Procedure

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

Step 2 In the **Navigation** pane, click **AAA Authentication**.

Step 3 In the **Work** pane, in the **Properties** area, 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.

Changing Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs Using the NX-OS Style CLI

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. To do so, an administrator adds a Cisco AV pair to the existing user record. 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+. One AV pair format contains a Cisco UNIX user ID and one does not. Both are correct if all remote users have the same role and mutual file access is acceptable. If the UNIX user ID is not specified, ID 23999 is applied by the APIC system, and more than one role/read privilege is specified to any AV Pair user. This can cause users to have higher or lower permissions than configured through the group settings. This topic explains how to change the bahavior if that is not acceptable.

To change the default behavior for remote users with missing or bad Cisco AV pairs using the NX-OS CLI:

Procedure

Step 1 In the NX-OS CLI, start in Configuration mode.

Example:

```
apic1#
apic1# configure
```

Step 2 Configure the aaa user default role.

Example:

```
apic1(config)# aaa user default-role
 assign-default-role  assign-default-role
 no-login             no-login
```

Step 3 Configure the aaa authentication login methods.

Example:

```
apic1(config)# aaa authentication
 login  Configure methods for login

apic1(config)# aaa authentication login
 console  Configure console methods
 default  Configure default methods
 domain   Configure domain methods

apic1(config)# aaa authentication login console
 <CR>

apic1(config)# aaa authentication login domain
 WORD      Login domain name
 fallback
```

About SAML

SAML is an XML-based open standard data format that enables administrators to access a defined set of Cisco collaboration applications seamlessly after signing into one of those applications. SAML describes the exchange of security related information between trusted business partners. It is an authentication protocol used by service providers to authenticate a user. SAML enables exchange of security authentication information between an Identity Provider (IdP) and a service provider.

SAML SSO uses the SAML 2.0 protocol to offer cross-domain and cross-product single sign-on for Cisco collaboration solutions. SAML 2.0 enables SSO across Cisco applications and enables federation between Cisco applications and an IdP. SAML 2.0 allows Cisco administrative users to access secure web domains to exchange user authentication and authorization data, between an IdP and a Service Provider while maintaining high security levels. The feature provides secure mechanisms to use common credentials and relevant information across various applications.

The authorization for SAML SSO Admin access is based on Role-Based Access Control (RBAC) configured locally on Cisco collaboration applications.

SAML SSO establishes a Circle of Trust (CoT) by exchanging metadata and certificates as part of the provisioning process between the IdP and the Service Provider. The Service Provider trusts the IdP's user information to provide access to the various services or applications.

Note Service providers are no longer involved in authentication. SAML 2.0 delegates authentication away from the service providers and to the IdPs.

The client authenticates against the IdP, and the IdP grants an Assertion to the client. The client presents the Assertion to the Service Provider. Since there is a CoT established, the Service Provider trusts the Assertion and grants access to the client.

Enabling SAML SSO results in several advantages:

- It reduces password fatigue by removing the need for entering different user name and password combinations.

- It transfers the authentication from your system that hosts the applications to a third party system. UsingSAML SSO, you can create a circle of trust between an IdP and a service provider. The service provider trusts and relies on the IdP to authenticate the users.
- It protects and secures authentication information. It provides encryption functions to protect authentication information passed between the IdP, service provider, and user. SAML SSO can also hide authentication messages passed between the IdP and the service provider from any external user.
- It improves productivity because you spend less time re-entering credentials for the same identity.
- It reduces costs as fewer help desk calls are made for password reset, thereby leading to more savings.

Basic Elements of SAML

- Client (the user's client): This is a browser-based client or a client that can leverage a browser instance for authentication. For example, a system administrator's browser.
- Service provider: This is the application or service that the client is trying to access.
- An Identity Provider (IdP) server: This is the entity that authenticates user credentials and issues SAML Assertions.
- Lightweight Directory Access Protocol (LDAP) users: These users are integrated with an LDAP directory, for example Microsoft Active Directory or OpenLDAP. Non-LDAP users reside locally on the Unified Communications server.
- SAML Assertion: It consists of pieces of security information that are transferred from IdPs to the service provider for user authentication. An assertion is an XML document that contains trusted statements about a subject including, for example, a username and privileges. SAML assertions are usually digitally signed to ensure their authenticity.
- SAML Request: This is an authentication request that is generated by a Unified Communications application. To authenticate the LDAP user, Unified Communications application delegates an authentication request to the IdP.
- Circle of Trust (CoT): It consists of the various service providers that share and authenticate against one IdP in common.
- Metadata: This is an XML file generated by an ACI application as well as an IdP. The exchange of SAML metadata builds a trust relationship between the IdP and the service provider.
- Assertion Consumer Service (ACS) URL: This URL instructs the IdPs where to post assertions. The ACS URL tells the IdP to post the final SAML response to a particular URL.

Note All in-scope services requiring authentication use SAML 2.0 as the SSO mechanism.

Supported IdPs and SAML Components

Supported IdPs

Identity Provider (IdP) is an authentication module that creates, maintains, and manages identity information for users, systems, or services and also provides authentication to other applications and service providers within a distributed network.

With SAML SSO, IdPs provide authentication options based on the user role or log in options for each of the Cisco collaboration applications. The IdPs store and validate the user credentials and generate a SAML response that allows the user to access the service provider protected resources.

Note You must be familiar with your IdP service, and ensure that it is currently installed and operational.

The APIC SAML SSO feature has been tested with following IdPs:

- https://technet.microsoft.com/en-us/library/cc772128(WS.10).aspx
- Okta Single Sign-On: https://www.okta.com/products/single-sign-on/

Note APIC expects both Response Message and Assertions to be signed for the SAML Response and does not support encrypted Assertions. Please configure the IdP as required.

SAML Components

A SAML SSO solution is based on a particular combination of assertions, protocols, bindings, and profiles. The various assertions are exchanged among applications and sites using the protocols and bindings, and those assertions authenticate the users among sites. The SAML components are as follows:

- SAML Assertion: It defines the structure and content of the information that is transferred from IdPs to service providers. It consists of packets of security information and contains statements that service providers use for various levels of access-control decisions.SAML SSO provides the following types of statements:
  - Authentication statements- These statements assert to the service provider about the method of authentication that occurs between the IdP and the browser at a particular time.
  - Attribute statements- These statements assert about certain attributes (name-value pairs) that are associated with the user. The attribute assertions contain specific information about the user. The service providers use attributes to make access-control decisions.
- SAML protocol: A SAML protocol defines how the SAML requests for and gets assertions. This protocol is responsible for the SAML request and response elements that consist of certain SAML elements or assertions. The SAML 2.0 contains the following protocols:
  - Assertion Query and Request Protocol
  - Authentication Request Protocol

- SAML binding: A SAML binding specifies the mapping of SAML assertion and/or protocol message exchanges with standard messaging formats or communication protocols like SOAP exchanges. ACI supports the following SAML 2.0 bindings:
  - HTTP Redirect (GET) Binding
  - HTTP POST Binding
- SAML profile: A SAML profile provides a detailed description of the combination of SAML assertions, protocols, and bindings to support well-defined use cases.

NTP Setup

In SAML SSO, Network Time Protocol (NTP) enables clock synchronization between the APIC and IdP. SAML is a time sensitive protocol and the IdP determines the time-based validity of a SAML assertion. If the IdP and the APIC clocks are not synchronized, the assertion becomes invalid and stops the SAML SSO feature. The maximum allowed time difference between the IdP and the APIC is 3 seconds.

Note For SAML SSO to work, you must install the correct NTP setup and make sure that the time difference between the IdP and the APIC does not exceed 3 seconds. If IdP and APIC clocks are not synchronized, the user will be redirected back to the APIC's login page even after successful authentication on IdP.

DNS Setup

Domain Name System (DNS) enables the mapping of host names and network services to IP addresses within a network or networks. DNS server(s) deployed within a network provide a database that maps network services to hostnames and, in turn, hostnames to IP addresses. Devices on the network can query the DNS server and receive IP addresses for other devices in the network, thereby facilitating communication between network devices.

In summary, APIC and Idp should be able to resolve each other's fully qualified domain names to IP addresses and should be resolvable by the client.

Certificate Authority

Cisco recommends using server certificates that are signed by one of the following types of Certificate Authority (CA):

- **Public CA—**A third-party company verifies the server identity and issues a trusted certificate.
- **Private CA—**You create and manage a local CA and issue trusted certificates.

The signing process varies for each product and can vary between server versions. It is beyond the scope of this document to provide detailed steps for every version of each server. Refer the appropriate server documentation for detailed instructions on how to get certificates signed by a CA.

If you get server certificates signed by a public CA, the public CA should already have a root certificate present in the trust store on the client computer. In this case, you do not need to import root certificates on the client computers. You should import root certificates if the certificates are signed by a CA that does not already exist in the trust store, such as a private CA. In SAML SSO, the IdP and service providers must have CA signed certificates with the correct domains in the CN or SAN. If the correct CA certificates are not validated, the browser issues a pop up warning.

If the APIC's trust store does not include the root certificate of the IdP, a new certificate authority should be created. This Certificate Authority should be used later while configuring the SAML Provider on APIC.

Configuring APIC for SAML Access

Note SAML based Authentication is only for APIC GUI and not for CLI/REST. Also, not applicable for LEAF Switches and SPINEs. SAML configuration cannot be done via APIC CLI.

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 SAML server host name or IP address, and the IdP's metadata URL are available..
- The APIC management endpoint group is available.
- Set up the following:
  - Time Synchronization and NTP: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#concept_9CE11B84AD78486AA7D83A7DE1CE2A77.
  - Configuring a DNS Service Policy to Connect with DNS Providers Using the Advanced GUI: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_750E077676704BFBB5B0FE74628D821E.
  - Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_F037F1B75FF74ED1BCA4F3C75A16C0FA.

Procedure

Step 1 In the APIC, create the **SAML Provider**.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **SAML Management** > **SAML Providers**.
c) In the **Work** pane, choose **Actions** > **Create SAML Provider**.
d) Specify the SAML host name (or IP address), and IdP metadata URL.

- In case of AD FS, IdP Metadata URL is of the format *https://<FQDN of ADFS>/FederationMetadata/2007-06/FederationMetadata.xml*.
- In case of Okta, to get the IdP Metadata URL, copy the link for **Identity Provider Metadata** in the **Sign On** section of the corresponding SAML Application from the Okta server.

e) Configure the Https Proxy if it is needed to access the IdP metadata URL.

f) Select the Certificate Authority if IdP is signed by a Private CA.

Step 2 Create the **SAML Provider Group**.

a) In the **Navigation** pane, choose **SAML Management** > **SAML Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create SAML Provider Group**.
c) Specify the SAML provider group name, description, and providers as appropriate.

Step 3 Create the **Login Domain** for SAML.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

Setting Up a SAML Application in Okta

To configure SAML in Okta, log in to your Okta organization as a user with administrative privileges.

Note If you don’t have an Okta organization, you can create a free Okta at:

https://www.okta.com/start-with-okta/

Procedure

Step 1 In Okta, click on the blue **Admin** button.

Step 2 Click on the **Add Applications** shortcut.

Step 3 Click on the green **Create New App** button, and perform the following actions:

a) In the **Create New App** dialog box, select the **SAML 2.0** option, then click the green **Create** button.
b) In the **General Settings** box, enter **Example SAML Application** in the **App name** field, then click the green **Next** button.
c) In the **Configure SAML** section A **SAML Settings** field, paste your SAML URL into the **Single sign on URL**, **Recipient URL**, and **Audience Restriction** fields.
The fields should be of the below format:

- https://<APIC_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>
- Use Requestable SSO URLs to configure cluster of APICs:
    - https://<APIC1_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>
    - https://<APIC2_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>
    - https://<APIC3_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>
- Name ID Format: Transient
- Response: Signed
- Assertion Signature: Signed

- Assertion Encryption: Unencrypted
- SAML Single Logout: Disabled
- authnContextClassRef: PasswordProtectedTransport
- SAML Issuer ID: http://www.okta.com/${org.externalKey}

d) In the **Attribute Statements** section, add the information to the **FirstName**, **LastName**, **Email**, and **CiscoAvpair** fields and click **Next**.

Note A custom attribute called **CiscoAvpair** needs to be created for the Okta User in the **Profile Editor**. For more information on CiscoAvpair, see AV Pair on the External Authentication Server, on page 4.

e) In the **Feedback** box, select **I'm an Okta customer adding an internal app**, and **This is an internal app that we have created**, then click **Finish**.

Step 4 The **Sign On** section of your newly created **Example SAML Application** application appears. Save this page and open it on a separate tab or browser window. You will return to this page later to copy the **Identity Provider metadata** link for your SAML configuration.

Note To copy the metadata link, right-click on the **Identity Provider metadata** link and select **Copy**.

Setting Up a Relying Party Trust in AD FS

Add relying party trust in AD FS Management Console:

Procedure

Step 1 Add relying party trust:

a) Login to AD FS Management Console on your AD FS server, Navigate to **ADFS** > **Trust Relationships** > **Relying Party Trusts** and right-click on **Add Relying Party Trust** and click **Start**.

b) Choose **Enter data about the relying party manually** or **Import data about relying party from a file (skip the steps d, e, f and g)** by importing the metadata file generated using the **Download SAML Metadata** option available on the corresponding login domain setup in APIC.

c) Enter your preferred **Display Name** for the relying party trust and click **Next**.

d) Choose AD FS Profile and click **Next**.

e) Click **Next** again.

f) Select **Enable support for the SAML 2.0 Web SSO Protocol** and enter **Relying party SAML2.0 SSO service UR** as *https://<APIC_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>* and click **Next**.

g) Enter the **Relying party trust identifier –** *https://<APIC_hostname>/api/aaaLoginSSO.json*

h) Choose **I do not want to configure multi-factor authentication settings for this relying party trust at this time** and click **Next**.

i) Choose **Permit all users to access this relying party** and click **Next**.

j) Select **Open the Edit Claim rules** dialog for this relying party trust when the wizard closes and click **Close**.

Step 2 Add the following **Claim** rules:

a) Send LDAP Attributes as claims:

- In the **Edit Claim Rules** window, click **Add Rule**.
- Select the **Claim Rule Template** as Send LDAP attributes as **Claims** and click **Next**.
- Enter a **Rule_*Name*** and select **Active Directory** as the Attribute Store.
- Select the reserved User Attribute for storing CiscoAvpair (For Ex: **Department**) as LDAP attribute type and map it to Outgoing Claim Manually Type as **CiscoAvpair**.
- Select **E-Mail-Addresses** on LDAP Attribute and map it to the Outgoing Claim Type **E-mail Address** and click **Finish**.

b) Transform an Incoming Claim:

- Click **Add Rule** again in the **Edit Claim Rules** window, and select **Transform an Incoming Claim as Claim Rule Template** and click **Next**.
- Select **E-Mail Address** as the Incoming claim type.
- Select **Name ID** as Outgoing claim type.
- Select **Transient Identifier** as Outgoing name ID format.

Step 3 To add a cluster of APICs, one can either setup multiple **Relying Party Trusts** or setup single **Relying Party Trust** and add multiple **Relying Party Identifiers** and **SAML Assertion Consumer Endpoints** to it.

a) Adding other APICs in a cluster with same relying party trusts created above.

1. Navigate to **ADFS Management Console** > **ADFS** > **Trust Relationships** > **Relying Party Trusts** and right-click on **CiscoAPIC** > **Properties**.
2. Click on Identifiers tab and add other APICs in cluster as: *https://<APIC2_hostname>/api/aaaLoginSSO.json*, *https://<APIC3_hostname>/api/aaaLoginSSO.json*
3. Click on **Endpoints** tab and Other two APICs by clicking on **Add SAML**. **Add SAML Post Binding**, Index as 1 and Enter trusted URL as: *https://<APIC2_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>*, and **Add SAML Post Binding** as: *https://<APIC3_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>*.

Step 4 Message and Assertion need to be signed in ADFS from powershell in ADFS server. For Signing Message and Assertion in ADFS Server:

a) Open Windows Powershell (should be run as Administrator) and execute the below command:

b) Set-AdfsRelyingPartyTrust -TargetName **RelyingpartytrustnameOfCiscoAPIC** -SamlResponseSignature **MessageAndAssertion**.