Cisco Identity Services Engine API Reference Guide, Release 2.x

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: OL-26134-01

Cisco Identity Services Engine API Reference Guide, Release 2.x

CONTENTS

Preface

Overview of Cisco Identity Services Engine

Cisco Identity Services Engine (ISE), as a next-generation identity and access control policy platform enables enterprises to enforce compliance, enhance infrastructure security, and streamline their service operations. The unique architecture of Cisco ISE allows enterprises to gather real-time contextual information from networks, users, and devices in order to make proactive governance decisions by tying identity to various network elements including access switches, wireless LAN controllers (WLCs), virtual private network (VPN) gateways, and data center switches.

Cisco ISE is a key component of the Cisco Security Group Access Solution. Cisco ISE is a consolidated policy-based access control solution that:

- Combines authentication, authorization, accounting (AAA), posture, profiler, and guest management services into one appliance
- Enforces endpoint compliance by checking the device posture of all endpoints accessing the network, including 802.1X environments
- Provides support for discovery, profiling, policy-based placement, and monitoring of endpoint devices on the network
- Enables consistent policy in centralized and distributed deployments allowing services to be delivered where they are needed
- Employs advanced enforcement capabilities including Security Group Access (SGA) through the use of Security Group Tags (SGTs) and Security Group (SG) Access Control Lists (ACLs)
- Supports scalability to support a number of deployment scenarios from small office to large enterprise environments

The Cisco ISE architecture supports standalone and distributed deployments, allowing you to configure and manage your network from a centralized portal. For more information on the capabilities of Cisco ISE, see the *Cisco Identity Services Engine Admin Guide*.

Purpose

This application programming interface (API) reference guide provides only a brief high-level overview of the capabilities afforded by the supported APIs. The purpose of this API reference guide is to provide a developer, system or network administrator, or system integrator with basic guidelines for using the outlined APIs within the Cisco ISE deployment.

The REST API calls use queries to determine the following types of data:

- Number of active sessions
- Types of active sessions
- Authentication status of active session
- MAC addresses in use
- NAS IP addresses in use
- Node versions and types
- Reasons for node session failures

The External RESTful Services APIs and related API calls can be used to perform CRUD (Create, Read, Update, Delete) operations on Cisco ISE resources. External RESTful Services is based on HTTP protocol and REST methodology.

Note For more information about the Cisco ISE network, its nodes and personas, concepts of operation or usage, or how to use the Cisco ISE user interface, see the *Cisco Identity Services Engine Admin Guide*.

Audience

This API reference guide is intended for experienced system administrators who administer Cisco ISE appliances within a network environment, system integrators who may want to make use of the APIs, or third-party partners who have with the responsibility for managing or troubleshooting Cisco ISE deployments. As a prerequisite to using this API reference guide, you should have a basic understanding of troubleshooting and diagnostic practices and how to make and interpret API calls.

Document Conventions

This section outlines the conventions used throughout this document.

Caution Means *reader be careful*. You are capable of doing something that might result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

This API reference guide uses the following conventions to convey instructions and information.

Item	Convention
Commands, keywords, special terminology, and options that should be chosen during procedures	**boldface** font
Variables for which you supply values and new or important terminology	*italic* font
Displayed session and system information, paths, and file names	`screen` font
Information you enter	**`boldface screen`** font
Variables you enter	*`italic screen`* font
Menu items and button names	**boldface** font
Indicates menu items to choose, in the order in which you choose them.	**Option > Network Preferences**

Related Documentation

This section provides information on release-specific documentation, as well as platform-specific documentation.

General product information for Cisco ISE is available at http://www.cisco.com/go/ise. End-user documentation is available on Cisco.com at http://www.cisco.com/en/US/products/ps11640/tsd_products_support_series_home.html.

Platform-Specific Documentation

- Cisco Secure ACS
  http://www.cisco.com/en/US/products/ps9911/tsd_products_support_series_home.html
- Cisco NAC Appliance
  http://www.cisco.com/en/US/products/ps6128/tsd_products_support_series_home.html
- Cisco NAC Profiler
  http://www.cisco.com/en/US/products/ps8464/tsd_products_support_series_home.html
- Cisco NAC Guest Server
  http://www.cisco.com/en/US/products/ps10160/tsd_products_support_series_home.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, refer to the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

PART 1

Cisco ISE Monitoring REST APIs

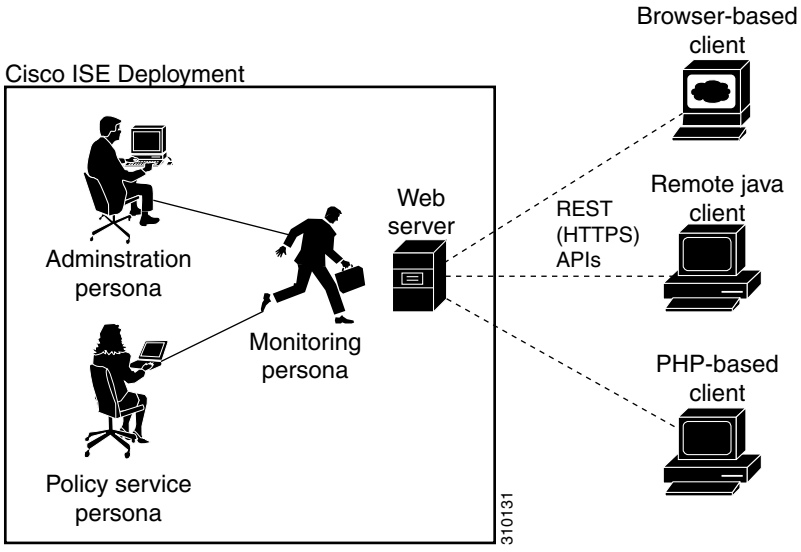
Introduction to the Monitoring REST API

The Monitoring REST API allows allow you to gather session and node-specific information by using Monitoring nodes in your network. A session is defined as the duration between when you access a desired node and complete the operations needed to gather information.

Monitoring REST API calls allow you to locate, monitor, and accumulate important real-time, session-based information stored in individual endpoints in a network. You can access this information through a Monitoring node.

The real-time, session-based information that you gather can help understand Cisco ISE operations and assist in diagnosing conditions or issues. It can also be used to troubleshoot error conditions or an activity or behavior that may be affecting monitoring operations. As shown in Figure 1-1, the Monitoring REST API calls are used to access the Monitoring node and retrieve important session-based information that is stored in the Cisco ISE deployment endpoints.

Figure 1-1 Monitoring REST API Calls in a Distributed Deployment



To perform operations using the Monitoring REST APIs, the users must be assigned to one of the following Admin Groups and must be authenticated against the credentials stored in the Cisco ISE internal database (internal admin users):

- Super Admin
- System Admin
- MnT Admin

The following Monitoring REST API categories are supported:

- Session Management
- Troubleshooting
- Change of Authorization (CoA)

You can use these APIs to gather information about endpoints being monitored by the Monitoring persona. For the remainder of this guide, “Monitoring node” will be used to describe the Monitoring persona of a Cisco ISE node.

Any attempt to use these categories to gather information about the Policy Service persona of a Cisco ISE appliance will result in an error. For more information about Cisco ISE nodes and personas, see *Cisco Identity Services Engine Admin Guide*.

Verifying a Monitoring Node

Before you Begin

Before you can successfully invoke the API calls on a Monitoring node, you need to verify that the node you want to monitor is valid.

Note To be able to use a public Monitoring REST API, you must first authenticate with Cisco ISE using valid credentials.

Step 1 Enter valid login credentials (Username and Password) in the Cisco ISE Login window, and click **Login**.

The Cisco ISE dashboard and user interface appears.

Step 2 Choose **Administration > System > Deployment**.

The Deployment Nodes page appears, which lists all configured nodes that are deployed.

Step 3 In the Roles column of the Deployment Nodes page, verify that the role for the target node that you want to monitor is listed as a Monitoring node.

Supported API Calls

The following tables describe the different types of API calls and provide an example of the API call format:

- Table 1-1 on page 1-3—defines API calls for session management.
- Table 1-2 on page 1-6—defines API calls for troubleshooting.
- Table 1-3 on page 1-7—defines CoA API calls.

If you intend to use a generic programmatic interface to authenticate with the Monitoring REST API supported by Cisco ISE, you need to first create a REST-based client that bridges Cisco ISE and the specific tool you use. You then use this REST client to authenticate with the Cisco ISE Monitoring REST APIs, marshal and submit the API requests to the Monitoring nodes, and then unmarshal the API responses and pass them on to the specified tool.

Table 1-1 Cisco ISE Session Management API Calls

API Call Category	Description and Example
Session Counters	
ActiveCount	Lists the number of active sessions. https://<ISEhost>/admin/API/mnt/Session/ActiveCount **Note** You must add the HTTP authorization header with the authorization credentials to view the number of active sessions.
PostureCount	Lists the number of Postured endpoints. https://<ISEhost>/admin/API/mnt/Session/PostureCount **Note** Posture is a service that aids in checking the state (or posture) for all the endpoints that connect to a Cisco ISE network. Cisco ISE utilizes NAC Agent for checking the posture compliance of a device.
ProfilerCount	Lists the number of active Profiler service sessions. https://<ISEhost>/admin/API/mnt/Session/ProfilerCount **Note** Profiler is a service that aids in identifying, locating, and determining the capabilities of all attached endpoints on a Cisco ISE network.

Table 1-1 Cisco ISE Session Management API Calls (continued)

API Call Category	Description and Example
Session List **Note** A session list includes the MAC address, network access device (NAD) IP address, username, and session ID information associated with a session.	
ActiveList	Lists all active sessions. https://<ISEhost>/admin/API/mnt/Session/ActiveList **Note** In this release of Cisco ISE, the maximum number of active authenticated endpoint sessions that can be displayed is limited to 250,000.
AuthList	Lists all currently active authenticated sessions. https://<ISEhost>/admin/API/mnt/Session/AuthList/<parameteroptions> You can specify the following parameter options that will return different values: • null/null—Lists all active authenticated sessions. • null/endtime—Lists all active authenticated sessions after the specified end time. • starttime/null—Lists all active authenticated sessions before the specified start time. • starttime/endtime—Lists all active authenticated sessions between the specified start time and end time. Enter the date and time for the start time and end time in the following format: YYYY-MM-DD hh:mm:ss.s where: • YYYY—four-digit year • MM—two-digit month (01=January, and so on) • DD—two-digit day of the month (01 through 31) • hh—two-digit hour (00 through 23) (a.m. and p.m. are not allowed) • mm—two-digit minute (00 through 59) • ss—two-digit second (00 through 59) • s—one or more digits representing a decimal fraction of a second **Note** Every Cisco ISE node is configured with a time zone. Recommended time zone is UTC. See Sample Data Returned from the AuthList API Call with the null/null Option, page 2-9, for samples that show all four parameter options.

Table 1-1 Cisco ISE Session Management API Calls (continued)

API Call Category	Description and Example
Session Attributes **Note** This is a timestamp-based search for the latest session that contains the specified search attribute.	
MACAddress	Searches the database for the latest session that contains the specified MAC address. https://<ISEhost>/admin/API/mnt/Session/MACAddress/<macaddress> **Note** XX:XX:XX:XX:XX:XX is the MAC address format and is not case sensitive (for example, 0a:0B:0c:0D:0e:0F). **Note** The MAC address serves as the only unique key to finding the correct session you want to monitor. Use the ActiveList API call to list all active sessions and their MAC addresses, from which you can base your MAC address search.
UserName	Searches the database for the latest session that contains the specified username. https://<ISEhost>/admin/API/mnt/Session/UserName/<username> **Note** Usernames must conform to the same Cisco ISE password policy used for network usernames. The only invalid character for the Monitoring REST APIs is the backslash (\) character. For details, see "User Password Policy" in *Cisco Identity Services Engine User Guide, Release 1.1*.
IPAddress	Searches the database for the latest session that contains the specified NAS IP address (IPv4 or IPv6 address). https://<ISEhost>/admin/API/mnt/Session/IPAddress/<nasipaddress> **Note** xxx.xxx.xxx.xxx is the NAS IP address format (for example, 10.10.10.10) or https://<ISEhost>/admin/API/mnt/Session/IPAddress/<nasipv6address> **Note** xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx is the NAS IPv6 address format (for example, 2001:cdba:0:0:0:0:3247:9651)
Audit Session ID	Searches the database for the latest session that contains the specified audit session ID. https://<ISEhost>/admin/API/mnt/Session/Active/SessionID/<audit-session-id>/0 **Note** Use the ActiveList API call to list all active sessions and their audit session IDs, from which you can base your audit session ID search. Alternatively, you can obtain the audit session ID from the Live Sessions page in the Admin portal.

For specific details about Cisco ISE API calls for session management, see Chapter 2, "Session Management Query APIs".

Table 1-2 Cisco ISE Troubleshooting API Calls - Troubleshooting

API Call	Description and Example
Version	Lists the node version and type. https://<ISEhost>/admin/API/mnt/Version Node type can be any of the following values (0-3): 0—STAND_ALONE_MNT_NODE 1—ACTIVE_MNT_NODE 2—STAND_BY_MNT_NODE 3—NOT_AN_MNT_NODE **Note** STAND_ALONE_MNT_NODE means it is a Monitoring node that does not function in any distributed deployment. ACTIVE_MNT_NODE means it is a primary node in a primary-secondary relationship in a distributed deployment. STAND_BY_MNT_NODE means it is a secondary node in a primary-secondary pair in a distributed deployment. NOT_AN_MNT_NODE means it is not a Monitoring node. See *Cisco Identity Services Engine User Guide, Release 1.1* for details about the supported ISE nodes and personas.
FailureReasons	Lists the reasons for failure. https://<ISEhost>/admin/API/mnt/FailureReasons Each failure reason displays an error code (failureReason id), a brief description (code), a failure reason (cause), and a possible response (resolution), as shown in the following example: <failureReason id="100009"> <code> 100009 WEBAUTH_FAIL <cause> This may or may not be indicating a violation. <resolution> Please review and resolve this issue according to your organization's policy. **Note** The FailureReasons API call to be called only once to gather the information from the Monitoring node. You should store the contents of any returned failure reasons into your own file system or database. The returned contents of these API calls are intended to be used for reference purposes. If you experience any issues during authentication, you should compare the failure reason code provided in the authentication response with the list of failure reasons that you have stored in your own file system or database. For a complete list of Cisco ISE failure reasons, see Appendix A, "Cisco ISE Failure Reasons Report".

Table 1-2 Cisco ISE Troubleshooting API Calls - Troubleshooting (continued)

API Call	Description and Example
AuthStatus	Lists the authentication status for all sessions. https://<ISEhost>/admin/API/mnt/AuthStatus/MACAddress/<macaddress>/<numberofseconds>/<numberofrecordspermacaddress>/All **Note** The seconds parameter <numberofseconds> is user-configurable, the range is from 0 to 432000 seconds (5 days).
Get Session Accounting Status	
AcctStatus	Lists the accounting status of all sessions within a specific period of time. https://<ISEhost>/admin/API/mnt/AcctStatusTT/MACAddress/<macaddress>/<numberof seconds> **Note** The seconds parameter <numberofseconds> is user-configurable, with the range is from 0 to 432000 seconds (5 days).

For specific details about Cisco ISE API calls for troubleshooting, see Chapter 2, "Session Management Query APIs".

Table 1-3 Cisco ISE Change of Authorization API Calls

API Call	Description and Example
Reauth	Sends a session reauthentication command and type. https://<ISEhost>/admin/API/mnt/CoA/Reauth/<serverhostname>/<macaddress>/<reauthtype>/<nasipaddress>/<destinationipaddress> Where <*ISEhost*> denotes the ip address of the ISE host, <*serverhostname*> denotes the name of the ISE server, <*nasipaddress*> denotes the identifying ip address of NAS, and <*destinationipaddress*> denotes the ip address of the destination. Reauth type can be any of the following values (0-2): 0—REAUTH_TYPE_DEFAULT 1—REAUTH_TYPE_LAST 2—REAUTH_TYPE_RERUN **Note** If you do not know the NAS IP address, you can enter the required values up to that point and the API will use these values in its search query. However, you must know the MAC address to perform this API call, but you can leave other parameters starting from NAS IP address as null. If the NAS IP address is provided then it's necessary to also provide the Destination IP address. This API call can only be executed on a Monitoring ISE node, which submits the requests to perform CoA remotely. The Administration ISE node is not involved or required to execute these CoA API calls.

Table 1-3 Cisco ISE Change of Authorization API Calls (continued)

API Call	Description and Example
Session Disconnect	
Disconnect	Sends a session disconnect command and port option type. https://<ISEhost>/admin/API/mnt/CoA/Disconnect/<serverhostname>/<macaddress>/<disconnecttype>/<nasipaddress>/<destinationipaddress> Port option type can be any of the following values (0-2): 0—DYNAMIC_AUTHZ_PORT_DEFAULT 1—DYNAMIC_AUTHZ_PORT_BOUNCE 2—DYNAMIC_AUTHZ_PORT_SHUTDOWN **Note** If you do not know the NAS IP address, enter the required values up to that point and the API will use these values in its search query. However, you must know the MAC address to perform this API call, but you can leave other parameters as null.

For details about Cisco ISE Change of Authorization API calls, see Chapter 4, "Change of Authorization REST APIs".

HTTP PUT API Calls

Similar to AuthStatus API call in Table 1-2, there is an HTTP PUT version of an API call that allows clients to retrieve account status. The Monitoring REST API supports both HTTP PUT and HTTP GET calls, with the examples in this guide documenting HTTP GET calls. HTTP PUT addresses the need for calls that require parameter inputs. The following schema file example is a request for account status:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="acctRequest" type="mnTRESTAcctRequest"/>
<xs:complexType name="mnTRESTAcctRequest">
    <xs:complexContent>
      <xs:extension base="mnTRESTRequest">
        <xs:sequence>
          <xs:element name="duration" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="mnTRESTRequest" abstract="true">
    <xs:sequence>
      <xs:element name="valueList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="value" type="xs:string" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="searchCriteria" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Session Management Query APIs

This chapter describes the session management API calls that provide the means for retrieving important session-related information from within the Cisco Monitoring ISE node in your Cisco ISE deployment.

Session Counter API Calls

The following session counter API calls let you quickly gather a current count of session-related information on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Active sessions (ActiveCount)—An active session is one that is authenticated onto the network.
- Postured sessions (PostureCount)—Postured state is asserted when posture is concluded (Compliant/Noncompliant). Posture is optional, for example, IP-phone/printer would not go to Postured state. Postured state is a short lived interim state, since after Postured, it moves to Started state when accounting start is set.
- Profiled sessions (ProfilerCount)

These various states are meant to troubleshoot if an endpoint gets stuck in any of the phases.

Active Sessions Counter

You can use the ActiveCount API call to retrieve a count of all currently active sessions.

Note You must add the HTTP authorization header with the authorization credentials to view the number of active sessions.

ActiveCount API Output Schema

This sample schema file is the output of the ActiveCount API call for retrieving a count of the active sessions on the target Monitoring persona of an ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="activeCount"/>
  <xs:complexType name="activeCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
```

```
  </xs:complexType>
</xs:schema>
```

Invoking the ActiveCount API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the ActiveCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>):

`https://acme123/`admin/API/mnt`/Session/ActiveCount`

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents the target Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the ActiveCount API Call

The following example illustrates the data returned (number of active sessions) when you invoke an ActiveCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<sessionCount>
<count>5</count>
</sessionCount>
```

Posture Sessions Counter

You can use the PostureCount API call to retrieve a current count of all currently active Posture sessions.

PostureCount API Output Schema

This sample schema file is the output of the PostureCount API call for retrieving a count of the current active Posture sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="postureCount"/>

  <xs:complexType name="postureCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the PostureCount API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the PostureCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/Session/<specific-api-call>):

```
https://acme123/admin/API/mnt/Session/PostureCount
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents the target Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the PostureCount API Call

The following example illustrates the data returned (number of current active Posture sessions) when you invoke a PostureCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionCount>
<count>3</count>
</sessionCount>
```

Profiler Sessions Counter

You can use the ProfilerCount API call to retrieve a count of all currently active Profiler sessions.

ProfilerCount API Output Schema

This sample schema file is the output of the ProfilerCount API call for retrieving a count of the current active Profiler sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="profilerCount"/>

  <xs:complexType name="profilerCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the ProfilerCount API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the ProfilerCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/Session/<specific-api-call>):

```
https://acme123/admin/API/mnt/Session/ProfilerCount
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the ProfilerCount API Call

The following example illustrates the data returned (number of active Profiler sessions) when you invoke a ProfilerCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionCount>
<count>1</count>
</sessionCount>
```

Simple Session List API Calls

The following simple session list API calls let you quickly gather session-related information such as the MAC address, the network access device (NAD) IP address, user name, and session ID associated with a current active session on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Active sessions list (ActiveList)
- Authenticated sessions list (AuthList)

Active Sessions List

You can use the ActiveList API call to list all currently active sessions.

Note The maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.

ActiveList API Output Schema

This sample schema file is the output of the ActiveList API call for retrieving a list of the current active sessions (and session-related information) on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

<xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
<xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>
```

Invoking the ActiveList API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the ActiveList API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/Session/<specific-api-call>):

```
https://acme123/admin/API/mnt/Session/ActiveList
```

Note You must carefully enter each API call in the URL Address field of a target node, because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the ActiveList API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an ActiveList API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="5">
-
<activeSession>
```

```
<calling_station_id>00:0C:29:FA:EF:0A</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<calling_station_id>70:5A:B6:68:F7:CC</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<acct_session_id>00000032</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<acct_session_id>0000002C</acct_session_id>
<audit_session_id>0ACB6BA10000002A165FD0C8</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>ipepvpnuser</user_name>
<calling_station_id>172.23.130.89</calling_station_id>
<nas_ip_address>10.203.107.45</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>A2000070</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Authenticated Sessions List

You can use the AuthList API call to retrieve a list of all currently active authenticated sessions.

Note The maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.

AuthList API Output Schema

This sample schema file is the output of the AuthList API call for retrieving a list of all currently active authenticated sessions within a specified period of time (or for no specified time using the "null/null" parameter) on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>

</xs:schema>
```

Invoking the AuthList API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the AuthList API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/Session/<specific-api-call>):

Note The first of the following two examples uses a defined starttime and null parameter, which displays a list of the currently active sessions that were authenticated after the specified start time. The second example uses the null/null parameter that displays a list of all currently active authenticated sessions. See Sample Data Returned from the AuthList API Call with the null/null Option, page 2-9, which displays samples of the four parameter setting types for this API call.

```
https://acme123/admin/API/mnt/Session/AuthList/2010-12-14 15:33:15/null

https://acme123/admin/API/mnt/Session/AuthList/null/null
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the AuthList API Call with the null/null Option

The following example illustrate the list of currently active authenticated sessions that is returned when you invoke an AuthList API call using the null/null option:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<audit_session_id>0acb6b0c000000174D07F487</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
```

```
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Sample Data Returned from the AuthList API Call with the endtime/null Option

The following example illustrate the list of currently active authenticated sessions that is returned when you invoke an AuthList API call using the endtime/null option:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
```

```
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Sample Data Returned from the AuthList API Call with the null/starttime Option

The following example illustrate the list of currently active authenticated sessions that is returned when you invoke an AuthList API call using the null/starttime option:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
```

```
</activeSession>
</activeSessionList>
```

Sample Data Returned from the AuthList API Call with the statttime/endtime Option

The following example illustrate the list of currently active authenticated sessions that is returned when you invoke an AuthList API call using the starttime/endtime option:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Detailed Session Attribute API Calls

The following detailed session attribute API calls let you quickly search the latest session for key information, such as the following:

- MAC address session search (MACAddress)
- User name session search (UserName)
- NAS IP address session search (IPAddress associated with a target Monitoring ISE node)
- Endpoint IP address session search (EndPointIPAddress)
- Audit session ID search (Audit Session ID)

MAC Address Session Search

You can use the MACAddress API call to retrieve a specified MAC address from a current, active session. This API call lists a variety of session-related information drawn from node database tables.

MACAddress API Output Schema

This sample schema file is the output of the MACAddress API call for retrieving a specified MAC address from the current active sessions:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
      <xs:element name="response" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
      <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
      <xs:element name="use_case" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
      <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
      <xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
      <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
      <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
      <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
      <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
      <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
      <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
      <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
      <xs:element name="response_time" type="xs:long" minOccurs="0"/>
      <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
      <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
      <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
      <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
      <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
      <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
      <xs:element name="security_group" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="started" type="xs:anyType" minOccurs="0"/>
      <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
      <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
      <xs:element name="type" type="xs:long" minOccurs="0"/>
      <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="vlan" type="xs:string" minOccurs="0"/>
      <xs:element name="dacl" type="xs:string" minOccurs="0"/>
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
```

```
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>
```

Invoking the MACAddress API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the MACAddress API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>/<macaddress>):

```
https://acme123/admin/API/mnt/Session/MACAddress/0A:0B:0C:0D:0E:0F
```

Note Make sure that you specify the MAC address using the XX:XX:XX:XX:XX:XX format.

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the MACAddress API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an MACAddress API call:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hunter_thompson</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
```

```
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=00-14-BF-5A-0C-03; User-Name=00-14-BF-5A-0C-03;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>00:14:BF:5A:0C:03</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Ethernet</nas_port_type>
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
```

```
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

User Name Session Search

You can use the UserName API call to retrieve a specified user name from a current, active session. This API will list a variety of session-related information drawn from node database tables.

UserName API Output Schema

This sample schema file is the output of the UserName API call for retrieving a specified user name from the current active sessions:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="response" type="xs:string" minOccurs="0"/>
<xs:element name="service_type" type="xs:string" minOccurs="0"/>
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
<xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="response_time" type="xs:long" minOccurs="0"/>
<xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="acct_id" type="xs:long" minOccurs="0"/>
<xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
<xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
<xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_class" type="xs:string" minOccurs="0"/>
<xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
<xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
<xs:element name="termination_action" type="xs:string" minOccurs="0"/>
<xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
<xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
<xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
<xs:element name="security_group" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
<xs:element name="started" type="xs:anyType" minOccurs="0"/>
<xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
<xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
<xs:element name="type" type="xs:long" minOccurs="0"/>
<xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
```

```
      <xs:element name="vlan" type="xs:string" minOccurs="0"/>
      <xs:element name="dacl" type="xs:string" minOccurs="0"/>
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>
```

Invoking the UserName API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the UserName API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>/<username>):

```
https://acme123/admin/API/mnt/Session/UserName/graham_hancock
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the UserName API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke a UserName API call:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
```

```
-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>graham_hancock</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=graham_hancock; User-Name=graham_hancock;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>graham_hancock</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Ethernet</nas_port_type>
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
```

```
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

NAS IP Address Session Search

You can use the IPAddress API call to retrieve data for a specified NAS IP address (IPv4 or IPv6 address) from a current session. This API will list a variety of session-related information drawn from node database tables.

IPAddress API Output Schema

This sample schema file is the output of the IPAddress API call for retrieving a specified NAS IP address (IPv4 or IPv6 address) from the current active sessions:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
<xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
<xs:element name="auth_id" type="xs:long" minOccurs="0"/>
<xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="message_code" type="xs:string" minOccurs="0"/>
<xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
<xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
<xs:element name="identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="response" type="xs:string" minOccurs="0"/>
<xs:element name="service_type" type="xs:string" minOccurs="0"/>
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
<xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="response_time" type="xs:long" minOccurs="0"/>
<xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="acct_id" type="xs:long" minOccurs="0"/>
<xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
<xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
<xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_class" type="xs:string" minOccurs="0"/>
<xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
<xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
<xs:element name="termination_action" type="xs:string" minOccurs="0"/>
<xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
<xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
<xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
      <xs:element name="security_group" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="started" type="xs:anyType" minOccurs="0"/>
      <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
      <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
      <xs:element name="type" type="xs:long" minOccurs="0"/>
      <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="vlan" type="xs:string" minOccurs="0"/>
      <xs:element name="dacl" type="xs:string" minOccurs="0"/>
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>
```

Invoking the NAS IPAddress API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the IPAddress API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>/<nasipaddress>):

```
https://acme123/admin/API/mnt/Session/IPAddress/10.10.10.10
```

Note Make sure that you specify IPv4 address/IPv6 address (NAS IP Address) using the xxx.xxx.xxx.xxx format or Compressed format respectively.

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the IPAddress API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an IPAddress API call:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>ipepvpnuser</user_name>
<nas_ip_address>10.10.10.10</nas_ip_address>
<nas_ipv6_address>2001:cdba::357:965</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>200:cdba:0000:0000:0000:0000:3157:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3247:9651</ipv6_address>
<ipv6_address>2001:cdba::3257:962</ipv6_address>
</framed_ipv6_address>
<calling_station_id>172.23.130.90</calling_station_id>
<nas_port>1015</nas_port>
<identity_group>iPEP-VPN-Group</identity_group>
<network_device_name>iPEP-HA-Routed</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>PAP_ASCII</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T19:57:29.885Z</auth_acs_timestamp>
<authentication_method>PAP_ASCII</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,15041,15004,15013,24210,24212,22037,15036,15048,15048,
15004,15016,11002
</execution_steps>
<audit_session_id>0acb6be4000000044D091DA9</audit_session_id>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<auth_id>1291240762083580</auth_id>
<auth_acsview_timestamp>2010-12-15T19:57:29.887Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/693</acs_session_id>
<service_selection_policy>iPEP-VPN</service_selection_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=ipepvpnuser; State=ReauthSession:0acb6be4000000044D091DA9;
Class=CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693;
Termination-Action=RADIUS-Request; }
</response>
<service_type>Framed</service_type>
-
<cisco_av_pair>
```

```
audit-session-id=0acb6be4000000044D091DA9,ipep-proxy=true
</cisco_av_pair>
<acs_username>ipepvpnuser</acs_username>
<radius_username>ipepvpnuser</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Virtual</nas_port_type>
<selected_azn_profiles>iPEP-Unknown-Auth-Profile</selected_azn_profiles>
<tunnel_details>Tunnel-Client-Endpoint=(tag=0) 172.23.130.90</tunnel_details>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-Protocol=PPP,Proxy-State=Cisco Secure
ACS9e733142-070a-11e0-c000-000000000000-2906094480-3222,CPMSessionID=0acb6be4000000044D091
DA9,CPMSessionID=0acb6be4000000044D091DA9,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Model Name=Unknown,Software Version=Unknown,Device
IP Address=10.203.107.228,Called-Station-ID=172.23.130.94
</other_attributes>
<response_time>20</response_time>
<acct_id>1291240762083582</acct_id>
<acct_acs_timestamp>2010-12-15T19:57:30.281Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T19:57:30.283Z</acct_acsview_timestamp>
<acct_session_id>F1800007</acct_session_id>
<acct_status_type>Start</acct_status_type>
-
<acct_class>
CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693
</acct_class>
<acct_delay_time>0</acct_delay_time>
<framed_protocol>PPP</framed_protocol>
<started xsi:type="xs:boolean">true</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

Endpoint IP Address Session Search

You can use the EndPointIPAddress API call to retrieve session directory information from a current, active session. This section provides a schema file output example, a procedure for searching the node database for the latest active session that contains the specified IP address by invoking the EndPointIPAddress API call, and a sample of the endpoint-related data returned after this API call is issued. This API call lists a variety of session directory information drawn from node database tables.

EndPointIPAddress API Output Schema

This sample schema file is the output of the EndPointIPAddress API call for retrieving session directory information about a specified endpoint from the current active sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="sessionParameters" type="restsdStatus"/>
<xs:complexType name="restsdStatus">
<xs:sequence>
<xs:element name="passed" type="xs:anyType" minOccurs="0"/>
<xs:element name="failed" type="xs:anyType" minOccurs="0"/>
<xs:element name="user_name" type="xs:string" minOccurs="0"/>
<xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
<xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port" type="xs:string" minOccurs="0"/>
<xs:element name="identity_group" type="xs:string" minOccurs="0"/>
<xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
<xs:element name="acs_server" type="xs:string" minOccurs="0"/>
<xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
<xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
<xs:element name="access_service" type="xs:string" minOccurs="0"/>
<xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
<xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
<xs:element name="radius_response" type="xs:string" minOccurs="0"/>
<xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
<xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
<xs:element name="auth_id" type="xs:long" minOccurs="0"/>
<xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="message_code" type="xs:string" minOccurs="0"/>
<xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
<xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
<xs:element name="identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="response" type="xs:string" minOccurs="0"/>
<xs:element name="service_type" type="xs:string" minOccurs="0"/>
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
<xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="response_time" type="xs:long" minOccurs="0"/>
<xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="acct_id" type="xs:long" minOccurs="0"/>
<xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
<xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
<xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_class" type="xs:string" minOccurs="0"/>
<xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
<xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
<xs:element name="termination_action" type="xs:string" minOccurs="0"/>
<xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
<xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
<xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
<xs:element name="security_group" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
<xs:element name="started" type="xs:anyType" minOccurs="0"/>
<xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
<xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
<xs:element name="type" type="xs:long" minOccurs="0"/>
<xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="vlan" type="xs:string" minOccurs="0"/>
<xs:element name="dacl" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
<xs:element name="interface_name" type="xs:string" minOccurs="0"/>
<xs:element name="reason" type="xs:string" minOccurs="0"/>
<xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Invoking the EndPointIPAddress API Call

Note Ensure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node.

To issue the EndPointIPAddress API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the EndPointIPAddress API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/Session/EndPointIPAddress/<endpoint_ip>):

```
https://acme123/ise/mnt/api/Session/EndPointIPAddress/A.B.C.D
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the EndPointIPAddress API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an EndPointIPAddress API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>00:0C:29:95:A5:C1</user_name>
<nas_ip_address>10.77.152.139</nas_ip_address>
<calling_station_id>00:0C:29:95:A5:C1</calling_station_id>
<nas_port>50109</nas_port>
<identity_group>RegisteredDevices</identity_group>
<network_device_name>switch</network_device_name>
<acs_server>ise248</acs_server>
<authen_protocol>Lookup</authen_protocol>
<framed_ip_address>10.20.40.10</framed_ip_address>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2012-03-13T17:02:22.169+05:30</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15048,15004,15041,15006,15013,24209,24211,22037,15036,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0A4D988B000000E337B8D983</audit_session_id>
<nas_port_id>GigabitEthernet1/0/9</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1331101769985927</auth_id>
<auth_acsview_timestamp>2012-03-13T17:02:22.171+05:30</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>ise248/120476308/97</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<authorization_policy>wired_redirect</authorization_policy>
<identity_store>Internal Endpoints</identity_store>
-
<response>
{UserName=00:0C:29:95:A5:C1; User-Name=00-0C-29-95-A5-C1;
State=ReauthSession:0A4D988B000000E337B8D983;
Class=CACS:0A4D988B000000E337B8D983:ise248/120476308/97;
Termination-Action=RADIUS-Request; Tunnel-Type=(tag=1) VLAN; Tunnel-Medium-Type=(tag=1)
802; Tunnel-Private-Group-ID=(tag=1) 30;
cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
```

```
cisco-av-pair=url-redirect=https://ise248.cisco.com:8443/guestportal/gateway?sessionId=0A4
D988B000000E337B8D983&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-cwa_wired-4f570619;
cisco-av-pair=profile-name=WindowsXP-Workstation; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0A4D988B000000E337B8D983</cisco_av_pair>
<acs_username>00:0C:29:95:A5:C1</acs_username>
<radius_username>00:0C:29:95:A5:C1</radius_username>
<selected_identity_store>Internal Endpoints</selected_identity_store>
<authentication_identity_store>Internal Endpoints</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Ethernet</nas_port_type>
<selected_azn_profiles>wired_cwa_redirect</selected_azn_profiles>
<response_time>17</response_time>
<destination_ip_address>10.77.152.248</destination_ip_address>
-
<other_attributes>
ConfigVersionId=15,DestinationPort=1812,Protocol=Radius,Framed-MTU=1500,EAP-Key-Name=,cisc
o-nas-port=GigabitEthernet1/0/9,CPMSessionID=0A4D988B000000E337B8D983,EndPointMACAddress=0
0-0C-29-95-A5-C1,EndPointMatchedProfile=WindowsXP-Workstation,HostIdentityGroup=Endpoint
Identity Groups:RegisteredDevices,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Device IP
Address=10.77.152.139,Called-Station-ID=EC:C8:82:55:2E:09
</other_attributes>
<acct_id>1331101769985928</acct_id>
<acct_acs_timestamp>2012-03-13T17:02:22.365+05:30</acct_acs_timestamp>
<acct_acsview_timestamp>2012-03-13T17:02:22.366+05:30</acct_acsview_timestamp>
<acct_session_id>000000FC</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>16411</acct_session_time>
<acct_input_octets>3053882</acct_input_octets>
<acct_output_octets>2633472</acct_output_octets>
<acct_input_packets>20166</acct_input_packets>
<acct_output_packets>20297</acct_output_packets>
<acct_class>CACS:0A4D988B000000E337B8D983:ise248/120476308/97</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
<vlan>30</vlan>
<dacl>#ACSACL#-IP-cwa_wired-4f570619</dacl>
<endpoint_policy>WindowsXP-Workstation</endpoint_policy>
</sessionParameters>
```

Audit Session ID Search

You can use the Audit Session ID API call to retrieve a specified audit session from a current, active session. This API call lists a variety of session-related information drawn from node database tables.

Audit Session ID API Output Schema

This sample schema file is the output of the Audit Session ID API call for retrieving a specified audit session ID from the current active sessions:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="sessionParameters" type="restsdStatus"/>

<xs:complexType name="restsdStatus">
  <xs:sequence>
    <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
    <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
    <xs:element name="user_name" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
    <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
    <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
    <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
    <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
    <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
    <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
    <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
    <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
    <xs:element name="access_service" type="xs:string" minOccurs="0"/>
    <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
    <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
    <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
    <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
    <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
    <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
    <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="message_code" type="xs:string" minOccurs="0"/>
    <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
    <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
    <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
    <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
    <xs:element name="response" type="xs:string" minOccurs="0"/>
    <xs:element name="service_type" type="xs:string" minOccurs="0"/>
    <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
    <xs:element name="use_case" type="xs:string" minOccurs="0"/>
    <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
    <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
    <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
    <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
    <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
    <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
    <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
    <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
    <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
    <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
    <xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
    <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
    <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
    <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
    <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
    <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
    <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
    <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
    <xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
    <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
    <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
    <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
    <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
    <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
    <xs:element name="response_time" type="xs:long" minOccurs="0"/>
```

```
      <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
      <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
      <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
      <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
      <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
      <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
      <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
      <xs:element name="security_group" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="started" type="xs:anyType" minOccurs="0"/>
      <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
      <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
      <xs:element name="type" type="xs:long" minOccurs="0"/>
      <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="vlan" type="xs:string" minOccurs="0"/>
      <xs:element name="dacl" type="xs:string" minOccurs="0"/>
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="nas_ipv6_address" type="xs:string"/>
  <xs:complexType name="framed_ipv6_address_list">
    <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>

</xs:schema>
```

Invoking the Audit Session ID API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the Audit Session ID API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/Session/Active/SessionID/*<audit-session-id>*/0):

```
https://acme123/admin/API/mnt/Session/Active/SessionID/0A000A770000006B609A13A9/
0
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the Audit Session ID API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an Audit Session ID API call:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-<activeSessionList noOfActiveSession="1">
    -<activeSession>
        <calling_station_id>00:50:56:10:13:02</calling_station_id>
        <session_state_bit>0</session_state_bit>
        <session_source>0</session_source>
        <acct_session_time>0</acct_session_time>
        <nas_ip_address>10.0.10.119</nas_ip_address>
        <nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
        <framed_ipv6_address>
        <ipv6_address>200:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
        <ipv6_address> 2001:cdba:0:0:0:0:3257:9651</ipv6_address>
        <ipv6_address>2001:cdba::3257:9652</ipv6_address>
        </framed_ipv6_address>
        <nas_port_id>GigabitEthernet1/0/15</nas_port_id>
        <auth_method>dot1x</auth_method>
        <auth_protocol>PEAP (EAP-MSCHAPv2)</auth_protocol>
        <posture_status>Compliant</posture_status>
        <endpoint_policy>Undetermined</endpoint_policy>
        <server>acme123</server>
        <paks_in>0</paks_in>
        <paks_out>0</paks_out>
        <bytes_in>0</bytes_in>
        <bytes_out>0</bytes_out>
    </activeSession>
</activeSessionList>
```

Stale Sessions

Some devices, such as Wireless Lan Controllers (WLCs), may allow stale sessions to linger. In such cases, you can use the HTTP **DELETE** API call to manually delete the inactive sessions. To do so, use **cURL**, a free 3rd-party command line tool for transferring data with URL (HTTP, HTTPS) syntax.

ISE no longer tracks those sessions. This is to mitigate the case when ISE lost connectivity to the network for an extended period of time, and missed a pile of accounting stops from the WLC/NAD. You can clear such stale information from ISE using this API.

Note GNU Wget, the free utility for retrieving files using HTTP and HTTPS, does not support the HTTP **DELETE** API call.

Removing Stale Sessions

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

Note API calls are case-sensitive, and must be entered carefully. The variable <mntnode> represents a Cisco Monitoring ISE node.

Step 4 To manually delete a stale session for a MAC address, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/MACAddress/<madaddress>
```

Step 5 To manually delete a stale session for a session ID, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/SessionID/<sid#>
```

Step 6 To manually delete all sessions on the Monitoring node, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/All
```

Related Topics

- Verifying a Monitoring Node, page 1-2

Query APIs for Troubleshooting

This chapter provides examples and describes how to use the individual Cisco Prime Network Control System (NCS) REST API calls.

Cisco Prime NCS API Calls

The Cisco Prime NCS API calls provide a mechanism for retrieving key troubleshooting information about the target Cisco Monitoring ISE node sessions that include node version and type, failure reasons, authentication status, and accounting status.

Troubleshooting Cisco ISE using the Query API Calls

Cisco Prime NCS troubleshooting API calls send status requests to the target Cisco Monitoring ISE node in your Cisco ISE deployment and retrieve the following diagnostic-related information:

- Node version and type (using the Version API call)
- Failure reasons (using the FailureReasons API call)
- Authentication status (using the AuthStatus API call)
- Accounting status (using the AcctStatus API call)

Node Version and Type API Call

You can use the Version API call to test the REST programmatic interface (PI) service and the credentials of each node. This section provides a schema file output example, a procedure for requesting the version of the Cisco ISE software and the node type by invoking this API call, and a sample of the node version and type that is returned after this API call is issued.

The node types can be any of the following:

- STANDALONE_MNT_NODE = 0
- ACTIVE_MNT_NODE= 1
- BACKUP_MNT_NODE = 2
- NOT_AN_MNT_NODE = 3

Version API Output Schema

This sample schema file is the output of the Version API call after sending it to the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="product" type="product"/>

  <xs:complexType name="product">
    <xs:sequence>
      <xs:element name="version" type="xs:string" minOccurs="0"/>
  <xs:element name="type_of_node" type="xs:int"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Version API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

If your login is unsuccessful, click the **Problem logging in?** link in the Login page and follow the instructions in Step 2.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the Version API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>):

```
https://acme123/admin/API/mnt/Version
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the Version API Call

The following example illustrates the data returned when you invoke a Version API call on a target Cisco Monitoring ISE node. This API call returns the following two values for the target node:

- Node version (this example displays 1.0.3.032).
- Type of Cisco Monitoring ISE node (this example displays a “1”, which means an active Cisco Monitoring ISE node).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<product name="Cisco Identity Services Engine">
<version>1.0.3.032</version>
<type_of_node>1</type_of_node>
</product>
```

Failure Reasons API Call

You can use the FailureReasons API call to return a list of failure reasons returned in the authentication status check done on the target node. This section provides a schema file output example, a procedure for requesting a list of all failure reasons logged by the Cisco Monitoring ISE node by invoking this API call, and a sample of the failure reasons returned after this API call is issued. Each failure reason that is returned consists of the following elements shown in Table 3-1.

Note For details about using the Cisco ISE Failure Reasons Editor to access the complete list of failure reasons, see Cisco ISE Failure Reasons Report, page A-1.

Table 3-1 Product Documentation for Cisco Identity Services Engine

Failure Reason Elements	Example
Failure reason ID	`<failureReason id="11011">`
Code	`<11011 RADIUS listener failed>`
Cause	`<Could not open one or more of the ports used to receive RADIUS requests>`
Resolution	`<Ensure that the ports 1812, 1813, 1645 and 1646 are not being used by another process on the system>`

Note You can also check for failure reason reports using the Cisco ISE user interface (click **Monitor > Reports > Catalog > Failure Reasons**), which will display failure reason reports.

FailureReasons API Output Schema

This sample schema file is the output of the FailureReasons API call after sending the request to a target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="failureReasonList" type="failureReasonList"/>

  <xs:complexType name="failureReasonList">
    <xs:sequence>
```

```
      <xs:element name="failureReason" type="failureReason" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="failureReason">
    <xs:sequence>
      <xs:element name="code" type="xs:string" minOccurs="0"/>
      <xs:element name="cause" type="xs:string" minOccurs="0"/>
      <xs:element name="resolution" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the FailureReasons API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

If your login is unsuccessful, click the **Problem logging in?** link in the Login page and follow the instructions in Step 2.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the FailureReasons API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>):

```
https://acme123/admin/API/mnt/FailureReasons
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the FailureReasons API Call

The following example illustrates the data returned when you invoke a FailureReasons API call on a target Cisco Monitoring ISE node. This API call returns a list of failure reasons from the target node, and each failure reason is defined by a failure ID, a failure code, a cause, and a resolution (if known).

Note The following FailureReasons API call example only displays a small sample of data that can be returned.

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<failureReasonList>
-
<failureReason id="100001">
-
<code>
100001 AUTHMGR-5-FAIL Authorization failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100002">
-
<code>
100002 AUTHMGR-5-SECURITY_VIOLATION Security violation on the interface
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100003">
-
<code>
100003 AUTHMGR-5-UNAUTHORIZED Interface unauthorized
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100004">
-
<code>
100004 DOT1X-5-FAIL Authentication failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100005">
<code>100005 MAB-5-FAIL Authentication failed for client</code>
<cause>This may or may not be indicating a violation</cause>
-
```

```
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100006">
-
<code>
100006 RADIUS-4-RADIUS_DEAD RADIUS server is not responding
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100007">
-
<code>
100007 EPM-6-POLICY_APP_FAILURE Interface ACL not configured
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
```

Related Topics

- Verifying a Monitoring Node, page 1-2
- Appendix A, “Cisco ISE Failure Reasons Report”

Authentication Status API Call

You can use the AuthStatus API call to check the authentication status of sessions on the target node. The query associated with this API call requires at least one MAC address to be searched for a match, with a user-configurable limit of the most recent records for the specified MAC address returned.

This section provides a schema file output example, a procedure for sending a a request to search for session authentication status on a target Monitoring mode by invoking this API call, and a sample of the data returned after this API call is issued.

The AuthStatus API call lets you configure the following search-related parameters:

- Duration—Defines the number of seconds in which an attempt is made to search and retrieve the authentication status records associated with the designated MAC address. Valid user-configurable values range from 1 to 864000 seconds (10 days). If you enter a value of 0 seconds, this specifies a default duration of 10 days.
- Records—Defines the number of session records to be searched per MAC address. Valid user-configurable values range from 1 to 500 records. If you enter 0, this specifies a default setting of 200 records.

Note If you specify the value 0 for both the duration and the records parameters, this API call returns only the very latest authentication session record associated with the designated MAC address(es).

Here is an example of the generic form of a URL with the Duration and Records attributes:

https://10.10.10.10/admin/API/mnt/AuthStatus/MACAddress/01:23:45:67:89:98/900000/2/All

- Attributes—Defines the number of attributes in the authentication status table that are returned from an authentication status search using the AuthStatus API call. Valid values include 0 (the default), All, or user_name+acs_timestamp (see the AuthStatus schema example, AcctStatus API Output Schema, page 3-12).
  - If you enter “0”, the attributes defined in Table 3-2 are returned. These are listed in the restAuthStatus section of the output schema.
  - If you enter “All”, a fuller set of attributes are returned. These are listed in the fullRESTAuthStatus section of the output schema.
  - If you enter the values listed in the schema for user_name+acs_timestamp, only those attributes are returned. The user_name and acs_timestamp attributes are listed in the restAuthStatus section of the output schema.

Table 3-2 Authentication Status Table Attributes

Attribute	Description
name=”passed” or name=”failed”	Authentication status results: • Passed • Failed
name=”user_name”	User name
name=”nas_ip_address”	IP address/hostname for the network access device
name=”nas_ipv6_address”	IPv6 address/hostname for the network access device
name=”failure_reason”	Reason for session authentication failure
name=”calling_station_id”	Source IP address
name=”nas_port”	Network access server port
name=”identity_group”	A logical group consisting of related users and hosts
name=”network_device_name”	Name of the network device
name=”acs_server”	Name of the Cisco ISE appliance
name=”eap_authentication”	Extensible Authentication Protocol (EAP) method used for authentication request
name=”framed_ip_address”	Address configured for a specific user
name=”framed_ipv6_address”	Address configured for a specific user
network_device_groups”	A logical group consisting of related network devices
name=”access_service”	Applied access service
name=”acs_timestamp”	Time stamp that is associated with the Cisco ISE authentication request
name=”authentication_method”	Identifies the method used in authentication
name=”execution_steps”	List of message codes for each diagnostic message logged while processing the request
name=”radius_response”	Type of RADIUS response (for example, VLAN or ACL)
name=”audit_session_id”	ID of the authentication session

Table 3-2 Authentication Status Table Attributes (continued)

Attribute	Description
name=”nas_identifier”	A network access server (NAS) associated with a specific resource
name=”nas_port_id”	ID of the NAS port used
name=”nac_policy_compliance”	Reflects Posture status (compliant or non-compliant)
name=”selected_azn_profiles”	Identifies the profile used in authorization
name=”service_type”	Indicates a framed user
name=”eap_tunnel”	Tunnel or outer method used for EAP authentication
name=”message_code”	Identifier of the audit message that defines the processed request result
name=”destination_ip_address”	Identifies the destination IP address

AuthStatus API Output Schema

This sample schema file is the output of the AuthStatus API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authStatusOutputList" type="fullRESTAuthStatusOutputList"/>

  <xs:complexType name="fullRESTAuthStatusOutputList">
    <xs:sequence>
      <xs:element name="authStatusList" type="fullRESTAuthStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatusList">
    <xs:sequence>
      <xs:element name="authStatusElements" type="fullRESTAuthStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="key" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatus">
    <xs:complexContent>
      <xs:extension base="restAuthStatus">
        <xs:sequence>
          <xs:element name="id" type="xs:long" minOccurs="0"/>
          <xs:element name="acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
          <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
          <xs:element name="response" type="xs:string" minOccurs="0"/>
          <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
          <xs:element name="use_case" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
          <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
          <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
          <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
```

```
            <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
            <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
            <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
            <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
            <xs:element name="authentication_identity_store" type="xs:string"

minOccurs="0"/>
            <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
            <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
            <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
            <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
            <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
            <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
            <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
            <xs:element name="selected_query_identity_stores" type="xs:string"
minOccurs="0"/>
            <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
            <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
            <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
            <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
            <xs:element name="response_time" type="xs:long" minOccurs="0"/>
            <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="restAuthStatus">
      <xs:sequence>
        <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
        <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
        <xs:element name="user_name" type="xs:string" minOccurs="0"/>
        <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
        <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
        <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
        <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
        <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
        <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
        <xs:element name="eap_authentication" type="xs:string" minOccurs="0"/>
        <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
        <xs:element name="access_service" type="xs:string" minOccurs="0"/>
        <xs:element name="acs_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
        <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
        <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
        <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
        <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
        <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
        <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
        <xs:element name="service_type" type="xs:string" minOccurs="0"/>
        <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
        <xs:element name="message_code" type="xs:string" minOccurs="0"/>
        <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="nas_ipv6_address" type="xs:string"/>
    <xs:complexType name="framed_ipv6_address_list">
      <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
      </xs:sequence>
```

```
  </xs:complexType>
  <xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>
```

Invoking the AuthStatus API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

If your login is unsuccessful, click the **Problem logging in?** link in the Login page and follow the instructions in Step 2.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the AuthStatus API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>/MACAddress/<macaddress>/<seconds>/<numberofrecordspermacaddress>/All):

```
https://acme123/admin/API/mnt/AuthStatus/MACAddress/00:50:56:10:13:02/120/100/All
```

Note The REST API calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the AuthStatus API Call

The following example illustrates the data returned when you invoke a AuthStatus API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<authStatusOutputList>
-
<authStatusList key="00:0C:29:46:F3:B8"><authStatusElements>
-
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>suser77</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
```

```
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>User Identity Groups:Guest</identity_group>
<acs_server>guest-240</acs_server>
<acs_timestamp>2012-10-05T10:50:56.515Z</acs_timestamp>
<execution_steps>5231</execution_steps>
<message_code>5231</message_code>
<id>1349422277270561</id>
<acsview_timestamp>2012-10-05T10:50:56.517Z</acsview_timestamp>
<identity_store>Internal Users</identity_store>
<response_time>146</response_time>
<other_attributes>ConfigVersionId=81,EndPointMACAddress=00-0C-29-46-F3-B8,PortalName=Defau
ltGuestPortal,
CPMSessionID=0A4D98D1000001F26F0C04D9,CiscoAVPair=</other_attributes>
</authStatusElements>
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>00:0C:29:46:F3:B8</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>2001:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9652</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>Guest_IDG</identity_group>
<network_device_name>switch</network_device_name>
<acs_server>guest-240</acs_server>
<authentication_method>mab</authentication_method>
<authentication_protocol>Lookup</authentication_protocol>
<acs_timestamp>2012-10-05T10:49:47.915Z</acs_timestamp>
<execution_steps>11001,11017,11027,15049,15008,15048,15048,15004,15041,15006,15013,24209,2
421
1,22037,15036,15048,15004,15016,11022,11002</execution_steps>
<response>{UserName
=00:0C:29:46:F3:B8; User-Name=00-0C-29-46-F3-B8;
State=ReauthSession:0A4D98D1000001F26F0C04D9;
Class=CACS:0A4D98D1000001F26F0C04D9:guest-240/138796808/76;
Termination-Action=RADIUS-Request; Tunnel-Type=(tag=1) VLAN;
Tunnel-Medium-Type=(tag=1) 802; Tunnel-Private-Group-ID=(tag=1) 2;
cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://guest-240.cisco.com:8443/guestportal/gateway?
sessionId=0A4D98D1000001F26F0C04D9&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-pre-posture-506e980a;
cisco-av-pair=profile-name=WindowsXP-Workstation;}</response
><audit_session_id>0A4D98D1000001F26F0C04D9</audit_session_id><nas_po
rt_id>GigabitEthernet1/0/17</nas_port_id><posture_status>Pending</posture_status>
<selected_azn_profiles>CWA_Redirect</selected_azn_profiles>
<service_type>Call Check</service_type>
<message_code>5200</message_code>
<nac_policy_compliance>Pending</nac_policy_compliance>
<id>1349422277270556</id>
<acsview_timestamp>2012-10-05T10:49:47.915Z</acsview_timestamp>
<identity_store>Internal Endpoints</identity_store>
<response_time>13</response_time>
<other_attributes>ConfigVersionId=81,DestinationPort=1812,Protocol=Radius,AuthorizationPol
icyMatchedRule=CWA_Redirect,
NAS-Port=50117,Framed-MTU=1500,NAS-Port-Type=Ethernet,EAP-Key-N
ame=,cisco-nas-port=GigabitEthernet1/0/17,AcsSessionID=guest-240/138796808/76,Us
eCase=Host Lookup,SelectedAuthenticationIdentityStores=Internal
Endpoints,ServiceSelectionMatchedRule=MAB,IdentityPolicyMatchedRule=Default,CPMS
essionID=0A4D98D1000001F26F0C04D9,EndPointMACAddress=00-0C-29-46-F3-B8,EndPointM
```

```
atchedProfile=WindowsXP-Workstation,ISEPolicySetName=Default,HostIdentityGroup=E
ndpoint Identity Groups:Guest_IDG,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Device IP
Address=10.77.152.209,Called-Station-ID=00:24:F7:73:9A:91,CiscoAVPair=audit-sess
ion-id=0A4D98D1000001F26F0C04D9</other_attributes>
-
</authStatusElements>
-
</authStatusList>
-
</authStatusOutputList>
```

Account Status API Call

You can use the AcctStatus API call to retrieve the latest device and session account information on the target node. This section provides a schema file output example, a procedure for sending a request for the latest device and session information by invoking this API call, and a sample of the data returned after this API call is issued. The AcctStatus API call lets you configure a time-related parameter:

- Duration—Defines the number of seconds in which an attempt is made to search and retrieve the latest account device records associated with the designated MAC address. Valid user-configurable values range from 1 to 432000 seconds (5 days). For example,
  - If you enter a value of 2400 seconds (40 minutes), this means that you want the latest account device records for the designated MAC address that are available in the past 40 minutes.
  - If you enter a value of 0 seconds, this specifies a default duration of 15 minutes (900 seconds). This means that you want the latest account device records for the designated MAC address that are available within this time period.

The AcctList API call provides the following account status data fields as API outputs (see Table 3-3):

Table 3-3 Accounting Status Data Fields

Data Field	Description
MAC address	MAC address of the client
audit-session-id	Authentication session ID
Packets in	Packets received count total
Packets out	Packets transmitted count total
Bytes in	Bytes received count total
Bytes out	Bytes transmitted count total
Session time	Duration of current sessions

AcctStatus API Output Schema

This sample schema file is the output of the AcctStatus API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="acctStatusOutputList" type="restAcctStatusOutputList"/>

  <xs:complexType name="restAcctStatusOutputList">
```

```
    <xs:sequence>
      <xs:element name="acctStatusList" type="restAcctStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="restAcctStatusList">
    <xs:sequence>
      <xs:element name="acctStatusElements" type="restAcctStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="username" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="restAcctStatus">
    <xs:sequence>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="paks_in" type="xs:long" minOccurs="0"/>
      <xs:element name="paks_out" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_in" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_out" type="xs:long" minOccurs="0"/>
      <xs:element name="session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="username" type="xs:string" minOccurs="0"/>
  <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the AcctStatus API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

If your login is unsuccessful, click the **Problem logging in?** link in the Login page and follow the instructions in Step 2.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the AcctStatus API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/<specific-api-call>/MACAddress/<macaddress>/<durationofcurrenttime>):

```
https://acme123/admin/API/mnt/AcctStatus/MACAddress/00:26:82:7B:D2:51/1200
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the AcctStatus API Call

The following example illustrates the data returned when you invoke an AcctStatus API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<acctStatusOutputList>
-
<acctStatusList macAddress="00:25:9C:A3:7D:48">
-
<acctStatusElements>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<audit_session_id>0acb6b0b0000000B4D0C0DBD</audit_session_id>
<paks_in>0</paks_in>
<paks_out>0</paks_out>
<bytes_in>0</bytes_in>
<bytes_out>0</bytes_out>
<session_time>240243</session_time>
<server>HAREESH-R6-1-PDP1</server>
</acctStatusElements>
</acctStatusList>
</acctStatusOutputList>
```

Change of Authorization REST APIs

This chapter provides examples and describes how to use the following individual Change of Authorization (CoA) REST API calls that are supported in this release of Cisco Identity Services Engine.

Introduction

The CoA API calls provide the means for sending session authentication and session disconnect commands to a specified Cisco Monitoring ISE node in your Cisco ISE deployment.

CoA Session Management API Calls

The CoA session management API calls allow you to send reauthentication and disconnect commands to a specified session on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Session reauthentication (Reauth)
- Session disconnection (Disconnect)

Session Reauthentication API Call

The Session Reauthentication API Call constitutes the following types:

- REAUTH_TYPE_DEFAULT = 0
- REAUTH_TYPE_LAST = 1
- REAUTH_TYPE_RERUN = 2

Reauth API Output Schema

This sample schema file is the output of the Reauth API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="remoteCoA" type="coAResult"/>
<xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Reauth API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the Reauth API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/CoA/<specific-api-call>/<macaddress>/<reauthtype>:

```
https://acme123/admin/API/mnt/CoA/Reauth/server12/00:26:82:7B:D2:51/1
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the Reauth API Call

The following example illustrates the data returned when you invoke a Reauth API call on a target Cisco Monitoring ISE node. Two possible results can be returned from invoking this command:

- True indicates that the command was successfully executed.
- False means that the command was not executed (due to a variety of conditions).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

Session Disconnect API Call

The Session Disconnect API call constitutes the following disconnect port option types:

- DYNAMIC_AUTHZ_PORT_DEFAULT = 0
- DYNAMIC_AUTHZ_PORT_BOUNCE = 1
- DYNAMIC_AUTHZ_PORT_SHUTDOWN = 2

Disconnect API Output Schema

This sample schema file is the output of the Disconnect API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="remoteCoA" type="coAResult"/>

  <xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Disconnect API Call

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, *https://<ise hostname or ip address>/admin/*).

Step 2 Enter the username and case-sensitive password, that was specified and configured during the initial Cisco ISE setup.

Step 3 Click **Login** or press **Enter**.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 4 Enter the Disconnect API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/admin/API/mnt/CoA/<Disconnect>/<serverhostname>/<macaddress>/<portoptiontype>/<nasipaddress>/<destinationipaddress>:

```
https://acme123/admin/API/mnt/CoA/Disconnect/server12/00:26:82:7B:D2:51/2/10.10.
10.10/192.168.1.1
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 5 Press **Enter** to issue the API call.

Related Topics

- Verifying a Monitoring Node, page 1-2

Sample Data Returned from the Disconnect API Call

The following example illustrates the data returned when you invoke a Disconnect API call on a target Cisco Monitoring ISE node. Two possible results can be returned by invoking this command:

- True indicates that the command was successfully executed.
- False means that the command was not executed (due to a variety of conditions).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

PART 2

Cisco ISE External RESTful Services APIs

CHAPTER 5

Introduction to ERS APIs

Prerequisites for Using the External RESTful Services API Calls

You must fulfill the following prerequisites before invoking an External RESTful Services API call:

- You must have enabled External RESTful Services from the GUI.
- You must have External RESTful Services Admin privileges.

You can use any REST client like JAVA, curl linux command, python or any other client to invoke External RESTful Services API calls.

External RESTful Services SDK

You can use the External RESTful Services SDK to start building your own tools. You can access the External RESTful Services SDK from the following URL: `https://<ISE-ADMIN-NODE>:9060/ers/sdk`.

External RESTful Services SDK can be accessed by the External RESTful Services Admin users only. The SDK consists the following components:

- Quick reference API documentation
- Complete list of all available API operations
- Schema files available for download
- Sample application in Java available for download
- Use cases in curl script format
- Use cases in python script format
- Instructions on using Chrome Postman

The following APIs are available in the SDK:

- Certificate template API
- Clear threats and vulnerabilities API
- Egress matrix cell API
- Endpoint API
- Endpoint certificate API
- Endpoints identity group API
- Guest location API

- Guest SMTP notification configuration API
- Guest SSID API
- Guest type API
- Guest user API
- Hotspot portal API
- IP-to-SGT mapping API
- IP-to-SGT mapping group API
- ISE service information API
- Identity group API
- Identity sequence API
- Internal user API
- My device portal API
- Native supplicant profile API
- Network device API
- Network device group API
- Node details API
- PSN node details with RADIUS service
- Portal API
- Portal theme API
- Profiler profile API
- SMS server API
- SXP connection API
- SXP local binding API
- SXP VPN API
- Security group API
- Security group ACL (SGACL) API
- Self registered portal API
- Sponsor group API
- Sponsor group member API
- Sponsor portal API
- Sponsored guest portal API

External RESTful Services API Authentication and Authorization

The External RESTful Services APIs are based on HTTPS protocol and REST methodology and uses port 9060.

The External RESTful Services APIs support basic authentication. The authentication credentials are encrypted and are part of the request header.

The ISE administrator must assign special privileges to a user to perform operations using the External RESTful Services APIs.

To perform operations using the External RESTful Services APIs (except for the Guest API), the users must be assigned to one of the following Admin Groups and must be authenticated against the credentials stored in the Cisco ISE internal database (internal admin users):

- External RESTful Services Admin—Full access to all ERS APIs (GET, POST, DELETE, PUT). This user can Create, Read, Update, and Delete ERS API requests.
- External RESTful Services Operator—Read Only access (GET request only).

If you do not have the required permissions and still try to perform operations using the External RESTful Services APIs, you will receive an error response.

Cisco ISE Failure Reasons Report

This appendix provides a procedure you can use to access the Cisco ISE Failure Reasons report. The Cisco ISE Failure Reason report allows you to view the list of failure reasons.

Introduction

The Cisco ISE Failure Reason report is an option in the Cisco ISE user interface that provides information about all of the failure reasons that could be encountered. You can use this to check on those that are returned as output from a Get Failure Reason Mapping call when using the Cisco ISE Query troubleshooting API.

The Cisco ISE Failure Reasons report lets you access the complete list of failure reasons defined by the Cisco ISE software that apply to Cisco Monitoring ISE node operations. The following procedure lets you view or edit the list of defined failure reasons. You must log into the Cisco ISE user interface of the target Cisco Monitoring ISE node to view and access the failure reasons. For details about logging in, see Verifying a Monitoring Node, page 1-2.

Viewing Failure Reasons

Step 1 Choose **Operations > Reports > Authentication Summary** report.

Step 2 In the navigation panel, expand **Monitoring** and select **Failure Reason Editor**.

Step 3 Choose Failure Reasons from the list of filters provided.

Step 4 Provide the failure reason that you are looking for.

Step 5 Click Run.

A list of failure reasons appears in the right panel.

Step 6 Click on any failure reason to get a detailed report in a new window.