APPENDIX B

API Support

This chapter discusses API support for the Clean Access Manager. Topics include:

Overview

Cisco NAC Appliance provides a utility script called **cisco_api.jsp** that allows you to perform certain operations using HTTPS POST. The actual Cisco NAC Appliance API for your Clean Access Manager is accessed via **https://<cam-IP-or-hostname>/admin/cisco_api.jsp.**

To access the web documentation page for the Cisco NAC Appliance API, login to your CAM web console and type “cisco_api.jsp” after “admin/” in your CAM console’s URL. This will redirect the browser to the web documentation page for the Cisco NAC Appliance API.

Note You must first log into the CAM web console before you can access the cisco_api.jsp documentation page.

To use this API, note the following:

- Competency with a scripting language (e.g. Java, Perl) is required and you must install the scripting software on the machine that runs these scripts.
- Cisco TAC does not support debugging of scripting packages (Java, Perl, etc.)

Note For general information on adding MAC address filters through the CAM web console interface, see Global Device and Subnet Filtering, page 3-10.

Authentication Requirements

Authentication over SSL is required to access the API. Two authentication methods are supported:

- Session-Based Authentication

  With this method, the administrator uses the *adminlogin* and *adminlogout* functions to create a cookie-based session with the server. The *adminlogin* function logs in the admin user and if successful, the HTTP response from the server will contain the session cookie to be used for the duration of the session. The *adminlogout* function logs out the admin user and invalidates the session. However, if the *adminlogout* function is not used, the CAM terminates the session by the configured or default admin session timeout.

- Function-Based Authentication

  If you do not want to used session-based authentication, you can use function-based authentication. With this method, the admin authenticates by passing his or her admin account credentials in every call to the API using the *admin* and *passwd* arguments in the request URL. If authenticating by function, you must add the *admin* and *passwd* parameters to all functions that you are using in your existing script. In this case, you do not use the *adminlogin* and *adminlogout* functions.

Administrator Operations

Use the *adminlogin* and *adminlogout* functions to create a shell script for session-based authentication using a session ID cookie. If you decide not to use session-based authentication, you will need to include the *admin* and *passwd* arguments within each API call instead.

adminlogin

The *adminlogin* function logs in the admin and starts the cookie-based session.

Required In Parameters:

- op: adminlogin
- admin: Administrator account username
- passwd: Administrator account password.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

<any subsequent operation>

The HTTP session cookie obtained through the adminlogin needs to be passed back as part of the HTTP request in any subsequent operation.

Required In Parameters:

- op: <ANY operation>
- <any operation specific parameters>

adminlogout

The *adminlogout* function logs out the administrator and invalidates the session.

Required In Parameters:

- op: adminlogout

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

Device Filter Operations

The following APIs perform operations on the CAM's Device Filter List (devices which bypass the user login requirement).

- addmac, page B-3
- removemac, page B-4
- checkmac, page B-4
- getmaclist, page B-5

Note See also changeuserrole, page B-9.

addmac

The *addmac* function adds one or more MAC addresses to the Device Filters list.

Required In Parameters:

- op: addmac
- mac: Specifies an exact MAC address or a range.
  Supported formats: 00:01:12:23:34:45 or 00:01:12:* or 00:01:12:23:34:45-11:22:33:44:55:66

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Optional In Parameters:

- ip: Specifies an IPv4 address for an exact MAC address. If you use a wildcard or range to specify a MAC address range, do not use the "ip" parameter. Supported format: 192.168.0.10
- type: Specifies one of the following strings: deny (default), allow, userole, check, or ignore.
- role: Specifies a role name. The role parameter is not required for the unauthenticated role (default) but is required for "userole" or "check".
- desc: Provides a description.
- ssip: Specifies the IP address used for configuring a Clean Access Server to Clean Access Manager. The default is global.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

removemac

The *removemac* function removes one or more MAC addresses from the Device Filters list.

Required In Parameters:

- op: removemac
- mac: Specifies one or more MAC addresses to delete from the device filters list. The MAC addresses must exactly match the display format including wildcards. You can specify multiple MAC addresses with a comma separated list.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Optional In Parameter:

- ssip: Specifies the IP address to use for configuring Clean Access Server to Clean Access Manager. The default is global.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

checkmac

The *checkmac* function queries the Device Filters list to check if a particular MAC address exists.

Required In Parameters:

- op: checkmac
- mac: Specifies the MAC address, which must exactly match the display format (00:01:12:23:34:45).

Optional In Parameter:

- ssip: Specifies the Clean Access Server IP address. By default, the *checkmac* function only checks global filters. If ssip provided, the Clean Access Server filters are also checked.

Out Parameters: <!--error=mesg--> comment

- Success:

  Either:

```
<!--error=0-->
<!--found=false-->
```

  Or:

```
<!--error=0-->
<!--found=true-->
<!--MAC=0A:13:07:9B:82:60,[IP=x.x.x.x,][CAS=y.y.y.y,]TYPE=ALLOW,[ROLE=zzz,]DESCRIPTION
=My Filter-->
```

In the device filter string:

- "IP=x.x.x.x" is only given for filters with an IP address configured.
- "CAS=y.y.y.y" is only given for server specific filters.
- "ROLE=zzz" is only given for filters with ROLE/CHECK types.
- For a specified single MAC address, the *checkmac* function returns the first matched filters, which can be a single MAC address filter or a MAC address wildcard/range filter.

- Failure: error string

getmaclist

The *getmaclist* function fetches the entire Device Filters list.

Required In Parameter:

- op: getmaclist

Out Parameters: <!--error=mesg--> comment

- Success:

```
<!--error=0-->
<!--count=number_of_filters-->
<!--MAC=0A:13:07:9B:82:60,[IP=x.x.x.x,][CAS=y.y.y.y,]TYPE=ALLOW,[ROLE=zzz,]DESCRIPTION
=My Filter--
...
```

In the device filter string:

- "IP=x.x.x.x" is only given for filters with an IP address configured.
- "CAS=y.y.y.y" is only given for server specific filters.
- "ROLE=zzz" is only given for filters with ROLE/CHECK types.

- Failure: error string

Certified Devices List Operations

The following APIs perform actions on the Certified Device list (devices which have met posture assessment requirements).

addcleanmac

The *addcleanmac* function adds one or more MAC addresses to the Certified Devices list as exempted devices.

Required In Parameters:

- op: addcleanmac

- mac: Specifies the MAC addresses to add. Supported formats 00:01:12:23:34:45 or 00-01-12-23-34-45 or 000112233445

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Optional In Parameter:

- ssip: Default is global. Specifies the IP address used for configuring Clean Access Server to Clean Access Manager.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

removecleanmac

The *removecleanmac* function removes one or more MAC addresses from the Certified Devices list.

Required In Parameters:

- op: removecleanmac
- mac: Specifies one or more MAC addresses to remove. Supported formats 00:01:12:23:34:45 or 00-01-12-23-34-45 or 000112233445

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Optional In Parameter:

- ssip: Default is global. Provide the IP address used for configuring Clean Access Server to Clean Access Manager.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: one or more error strings can appear if ssip is not provided and if a MAC address cannot be deleted from more than one Clean Access Server.

clearcertified

The *clearcertified* function deletes all of the existing entries from the Clean Access Certified Devices list.

Required In Parameter:

- op: clearcertified

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

User Operations

The following APIs perform user management operations:

- kickuser, page B-7
- kickuserbymac, page B-7
- kickoobuser, page B-8
- queryuserstime, page B-8
- renewuserstime, page B-8
- changeuserrole, page B-9
- changeloggedinuserrole, page B-9

Note See also getlocaluserlist, page B-10, addlocaluser, page B-10, and deletelocaluser, page B-11.

kickuser

The *kickuser* function terminates the active session of one or more currently logged-in in-band users, and removes the user from the In-Band Online Users list.

Required In Parameters:

- op: kickuser
- ip: Specifies one IP address or a comma separated list of IP addresses.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

kickuserbymac

The *kickuserbymac* function terminates the active session by MAC address of one or more logged-in in-band users and removes the user(s) from the In-Band Online Users list.

Required In Parameters:

- op: kickuserbymac
- mac: Specifies one MAC address or a comma separated list of MAC addresses.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

kickoobuser

The *kickoobuser* function terminates the active session of one or more OOB users and removes the user(s) from the Out-of-Band Online Users list.

Required In Parameters:

- op: kickoobuser
- mac: Specifies a MAC address or a comma separated list of MAC addresses.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

queryuserstime

The *queryuserstime* function queries the remaining session time for logged-in users. This function returns a list of logged-in users in roles with configured session timeouts.

Required In Parameters:

- op: queryuserstime

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; another <!--list=iplist--> comment with an IP list and session time remaining for each IP entry
- Failure: error string

renewuserstime

The *renewuserstime* function renews the logged-in users session timeout by a session.

Required In Parameters:

- op: renewuserstime
- list: Specifies a comma-separated list of IP addresses. Supported format: 10.1.10.10, 10.1.10.11, 10.1.10.12

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

changeuserrole

The *changeuserrole* function changes in-band user access permissions for a logged-in user by removing the user from the Online Users list and adding the user's MAC address to the Device Filters list with a new role.

Required In Parameters:

- op: changeuserrole
- ip: Specifies the IP address of a user who is logged in.
- role: Specifies the role to which the user is to be moved.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

changeloggedinuserrole

The *changeloggedinuserrole* function changes access permissions for a logged-in in-band user by changing that user's current role to a new role.

Required In Parameters:

- op: changeloggedinuserrole
- ip: Specifies the IP address of a logged-in user. To specify multiple users, use a comma-separated IP list.
- role: Specifies a new role for the user.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

Guest Access Operations

The following APIs allow administrators to create, delete, and view local user accounts on the CAM:

- getlocaluserlist, page B-10
- addlocaluser, page B-10
- deletelocaluser, page B-11

Local users are those internally validated by the CAM as opposed to an external authentication server. These APIs are intended to support guest access for dynamic token user access generation, providing the ability to:

- Use a webpage to access Cisco NAC Appliance API to insert a visitor username/password combination, such as jdoe@visitor.com/jdoe112805, and then assign a role, such as *guest1day*.
- Delete all guest users associated with the guest access role for that day.
- List all usernames associated with the guest access role.

These APIs support most implementations of guest user access dynamic token/password generation and allow the removal of those users for a guest role.

You must create the front-end generation password/token. For accounting purposes, Cisco NAC Appliance provides RADIUS accounting functionality only.

getlocaluserlist

The *getlocaluserlist* function returns a list of local user accounts with user name and role name.

Required In Parameters:

- op: getlocaluserlist

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; <!--count=10--> shows the number of users returned and is followed by same number of comments of form <!--NAME=jdoe,ROLE=Student-->
- Failure: error string

addlocaluser

The *addlocaluser* function adds a new local user account.

Required In Parameters:

- op: addlocaluser

- username: Specifies a new local user account user name.
- userpass: Specifies the user password for the new local user account.
- userrole: Specifies the role for the new local user account.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

deletelocaluser

The *deletelocaluser* function deletes one or all local user accounts.

Required In Parameters:

- op: deletelocaluser
- qtype: Specifies the data type: 'name' or 'all'
- qval: Specifies the exact username in single quotes or an empty string (‘’) to indicate “all.”

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

Report Operations

You can create scripts to compile lists of information or reports with the following report functions:

getversion

The *getversion* function returns the version number of the CAM.

Required In Parameters:

- op: getversion

Out Params:

- Comment of form <!--version=version--> is returned.

getuserinfo

Given an IP address, MAC address, or username, the *getuserinfo* function retrieves the following user information:

- *IP* in IPv4 format
- MAC address
- *Name* is the username
- *Provider* can be the LDAP server
- *Role* is the current role assigned to the user
- *Origrole* is the original role assigned to the user
- *VLAN* is the original VLAN tag
- *NEWVLAN* is the current VLAN tag
- Operating system of the user's system

If multiple users match the criteria, the system returns a list of users. If you enter "all" as the qtype Parameter, all information for all users is retrieved.

Required In Parameters:

- op: getuserinfo
- qtype: Specifies one of the following strings: ip, mac, name, or all.
- qval: Specifies an IP address, MAC address, or username depending on the qtype parameter; enter an empty string ('') to indicate "all."

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; <!--count=10--> shows the number of users returned and is followed by a corresponding number of comments <!--IP=10.1.10.12,MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP Server,ROLE=Student,ORIGROLE=Student,VLAN=1024,NEWVLAN=1024,OS=Windows XP-->
- Failure: error string

getoobuserinfo

Given an IP address, MAC address or username, the *getoobuserinfo* function retrieves information about the logged-in out-of-band (OOB) users, or given the qtype "all", the system generates a list of information about all logged-in OOB users. If multiple users match the criteria, the system generates a list of users.

Required In Parameters:

- op: getoobuserinfo
- qtype: Specifies the method of identifying one or more users: ip, mac, name, all.
- qval: Specifies an IP or MAC address or a username; enter an empty string (‘’) to indicate “all”.

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; <!--count=10--> shows the number of users returned and is followed by a matching number of comments of form <!--IP=10.1.10.12,MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP Server,ROLE=Student,AUTHVLAN=10,ACCESSVLAN=1024,OS=Windows XP,SWITCHIP=10.1.10.1,PORTNUM=18-->
- Failure: error string

getcleanuserinfo

Given a MAC address or username, the *getcleanuserinfo* function returns information about certified users. If there are multiple users matching the criteria, the system generates a list of certified users.

Required In Parameters:

- op: getcleanuserinfo
- qtype: Specifies the method of identifying the user: mac, name, all.
- qval: Specifies MAC address or username; enter an empty string (‘’) to indicate “all.”

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; <!--count=10--> shows the number of users returned and is followed by a matching number of comments of form <!--MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP Server,ROLE=Student,VLAN=10-->
- Failure: error string

getreports

The *getreports* function returns a report that contains customized content. You can also use this function to compile a list of users with certain software installed.

Required In Parameters:

op: getreports

Note If you do not use session-based authentication, the *admin* and *passwd* arguments are required. See Authentication Requirements, page B-2.

Optional Query Parameters:

Table B-1 lists the query Parameters for the *getreports* function.

Table B-1 Query Parameters for the getreports function

Parameter Name	Allowed Values	Description
status	One of the following values: • any (default) • success • failure	Reports only information for the specified status.
user	A string; empty single quotes (‘’) is the default	Reports information about the specified user.
agentType	One of the following values: • any (default) • web • win • mac	Reports information originating from the specified Agent type: Cisco NAC Agent, Clean Access Agent, or Cisco NAC Web Agent.
ip	One valid IPv4 address, such as 10.20.30.40; empty single quotes is the default	Reports information about the specified IP address.
mac	One valid MAC address, such as 00:01:12:23:34:45; empty single quotes is the default	Reports information about the specified MAC address.

Table B-1 Query Parameters for the getreports function (continued)

Parameter Name	Allowed Values	Description
os	One of the following values: • To indicate any OS, enter empty single quotes ('') (default) • WINDOWS_VISTA_ALL (Windows Vista) • WINDOWS_VISTA_HOME_BASIC (Windows Vista Home Basic) • WINDOWS_VISTA_BUSINESS (Windows Vista Business) • WINDOWS_VISTA_ULTIMATE (Windows Vista Ultimate) • WINDOWS_VISTA_ENTERPRISE (Windows Vista Enterprise) • WINDOWS_XP (Windows XP) • WINDOWS_PRO_XP (Windows XP Pro/Home) • WINDOWS_TPC_XP (Windows XP Tablet PC Edition) • WINDOWS_MCE_XP (Windows XP Media Center Edition) • WINDOWS_2K (Windows 2000) • WINDOWS_ME (Windows ME) • WINDOWS_98 (Windows 98) **Note** Cisco NAC Appliance no longer officially supports Windows ME or Windows 98 client login.	Reports information about the specified OS.

Table B-1 Query Parameters for the getreports function (continued)

Parameter Name	Allowed Values	Description
timeRange	timeFrom, timeTo • timeFrom can be one of the following values: – timestamp (format: yyyy-mm-dd hh:mm:ss) – negative integer representing the number of hours before now – past • timeTo can be one of the following values: – timestamp (format: yyyy-mm-dd hh:mm:ss) – negative integer representing the number of hours before now – now – -48, -24 (the day before last) – -24, now (within last day) – 2007-01-01 00:00:00, 2007-02-28 23:59:59 (Between Jan 1st and Feb 28th) Default: past, now (any time: all possible reports)	Reports information collected within the specified time range.
showText	One of the following values: • true— Returns the text. • false—Does not return the text. (default)	Indicates whether or not to return the report text.
orderBy	One of the following values: • user • ip • mac • os • time (default)	Specifies the report organization.
orderDir	One of the following values: • asc—Indicates ascending order. (default) • desc—Indicates descending order.	Specifies ascending or descending order for the data.
instSoft	One of the following values: • Empty single quotes (‘’) indicates “any” (default) • AV—Indicates AntiVirus installed • AS—Indicates AntiSpyware installed • UNKNOWN AV/AS—Indicates an unknown AV/AS	Restricts to reports containing this type of installed software.

Table B-1 Query Parameters for the getreports function (continued)

Parameter Name	Allowed Values	Description
reqName	Name of the AV or AS software requirement; empty quotes “any” (default)	Restricts to reports containing this software requirement.
reqStatus	One of the following values: • any (default) • success • failure	Restricts to reports where the software requirement is of this status (only if reqName is used).

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0; <!--count=count--> shows the number of reports returned; the reports follow the count comment and are of the form: <!--status=status,user=user,agentType=agentType,ip=ip,mac=mac,os=os,time=time,text=text-->
- Failure: error string