



Internal Architecture

This appendix describes the architectural components of the Cisco Secure Access Control Server Release 4.0 Solution Engine, hereafter referred to as ACS. ACS is modular and flexible to fit the needs of simple and large networks. This appendix includes the following topics:

- [Windows Services, page F-1](#)
- [SQL Registry, page F-2](#)
- [CSAdmin, page F-2](#)
- [CSAuth, page F-3](#)
- [CSDBSync, page F-3](#)
- [CSLog, page F-3](#)
- [CSMon, page F-3](#)
- [CSTacacs and CSRADIUS, page F-6](#)

Windows Services

ACS includes the following service modules:

- **CSAdmin**
- **CSAuth**
- **CSDBSync**
- **CSLog**
- **CSMon**
- **CSTacacs**
- **CSRADIUS**

You can stop or restart ACS services as a group, except for **CSAdmin**, by using the ACS web interface. For more information, see [Service Control, page 8-1](#). You can start, stop, and restart individual ACS services from the appliance serial console. For more information about starting, stopping, and restarting services by using the serial console, see the *Installation and Setup Guide for Cisco Secure ACS Solution Engine*.

SQL Registry

In order to create a unified data storage model, ACS has moved from multiple data storages to a standard SQL-based relational database.

SQL Registry

The SQL registry contains table information on all user and configuration data. SQL data is not made available for viewing and is protected by an encrypted password.

SQL Tables	Description
ConfigKey	Information that is not stored in the other tables. The data corresponds to registry keys.
ConfigValue	Data corresponding to registry values.
DictKey	Tree of attribute keys. The data is corresponds to registry keys of the ACS dictionary.
DictValue	Values for attributes keys from the ACSDictionaryKeys table.
Host	Information regarding all hosts in ACS.
HostService	Additional data for hosts of type remote agent.
Admin	ACS administrators. The permissions for each administrator are represented as a bitset inside a binary blob.
NetworkModel	Network model section.
Users	All user-specific information that was previously stored in the <i>user.dat</i> file. This table structure represents ACS UDB_ACCOUNT structure. However, some fields will not appear.
VarsDB	Currently in use but will be moved to a new table.

CSAdmin

CSAdmin is the service that provides the web server for the ACS web interface. After ACS is installed, you must configure it from its web interface; therefore, **CSAdmin** must be running when you configure ACS.

Because the ACS web server uses port 2002, rather than the standard port 80 that is usually associated with HTTP traffic, you can use another web server on the same machine to provide other web services. We have not performed interoperability testing with other web servers, but unless a second web server is configured to use either port 2002 or one of the ports within the range specified in the HTTP Port Allocation feature, you should not encounter port conflicts for HTTP traffic. For more information about the HTTP Port Allocation feature, see [Access Policy, page 12-8](#).

Although you can start and stop services from within the ACS web interface, this does not include starting or stopping **CSAdmin**. If **CSAdmin** stops abnormally because of an external action, you can only restart the service by using the appliance serial console. For more information about starting, stopping, and restarting services by using the serial console, see the *Installation and Setup Guide for Cisco Secure ACS Solution Engine*.

CSAdmin is multithreaded, which enables several ACS administrators to access it at the same time. Therefore, **CSAdmin** is well suited for distributed, multiprocessor environments.

CSAuth

CSAuth is the authentication and authorization service. It permits or denies access to users by processing authentication and authorization requests. **CSAuth** determines if access should be granted and defines the privileges for a particular user. **CSAuth** is the ACS database manager.

To authenticate users, ACS can use the internal database or one of many external databases. When a request for authentication arrives, ACS checks the database that is configured for that user. If the user is unknown, ACS checks the database(s) configured for unknown users. For more information about how ACS handles authentication requests for unknown users, see [About Unknown User Authentication, page 16-3](#).

For more information about the various database types supported by ACS, see [Chapter 13, “User Databases.”](#)

When a user has authenticated, ACS obtains a set of authorizations from the user profile and the group to which the user is assigned. This information is stored with the username in the ACS internal database. Some of the authorizations included are the services to which the user is entitled, such as IP over PPP, IP pools from which to draw an IP address, access lists, and password-aging information. The authorizations, with the approval of authentication, are then passed to the **CSTacacs** or **CSRADIUS** modules to be forwarded to the requesting device.

CSDBSync

CSDBSync is the service used to synchronize the ACS database with third-party relational database management system (RDBMS) systems. **CSDBSync** synchronizes AAA client, AAA server, network device groups (NDGs) and Proxy Table information with data from a table in an external relational database. For information on RDBMS Synchronization, see [RDBMS Synchronization, page 9-16](#).

CSLog

CSLog is the service used to capture and place logging information. **CSLog** gathers data from the TACACS+ or RADIUS packet and **CSAuth**, and then manipulates the data to be placed into the comma-separated value (CSV) files. CSV files can be imported into spreadsheets that support this format.

CSMon

CSMon is a service that helps minimize downtime in a remote access network environment. **CSMon** works for TACACS+ and RADIUS and automatically detects which protocols are in use.

You can use the ACS web interface to configure the **CSMon** service. The ACS Active Service Management feature provides options for configuring **CSMon** behavior. For more information, see [ACS Active Service Management, page 8-13](#).

**Note**

CSMon is not intended as a replacement for system, network, or application management applications but is provided as an application-specific utility that can be used with other, more generic system management tools.

CSMon performs four basic activities, outlined in the following topics:

- [Monitoring, page F-4](#)
- [Recording, page F-5](#)
- [Notification, page F-5](#)
- [Response, page F-5](#)

Monitoring

CSMon monitors the overall status of ACS and the system on which it is running. CSMon actively monitors three basic sets of system parameters:

- **Generic host system state**—CSMon monitors the following key system thresholds:
 - Available hard disk space
 - Processor utilization
 - Physical memory utilization

All events related to generic host system state are categorized as warning events.

- **Application-specific performance**
 - **Application viability**—CSMon periodically performs a test login by using a special built-in test account (the default period is one minute). Problems with this authentication can be used to determine if the service has been compromised.
 - **Application performance thresholds**—CSMon monitors and records the latency of each test authentication request (the time it takes to receive a positive response). Each time this is performed, CSMon updates a variable containing the average response time value. Additionally, it records whether retries were necessary to achieve a successful response. By tracking the average time for each test authentication, CSMon can build up a picture of expected response time on the system in question. CSMon can therefore detect whether excess re-tries are required for each authentication or if response times for a single authentication exceed a percentage threshold over the average.
- **System resource consumption by ACS**—CSMon periodically monitors and records the usage by ACS of a small set of key system resources and compares it against predetermined thresholds for indications of atypical behavior. The parameters monitored include the following:
 - Handle counts
 - Memory utilization
 - Processor utilization
 - Thread used
 - Failed log-on attempts

CSMon cooperates with CSAuth to keep track of user accounts being disabled by exceeding their failed attempts count maximum. This feature is more oriented to security and user support than to system viability. If configured, it provides immediate warning of brute-force attacks by alerting the administrator to a large number of accounts becoming disabled. In addition, it helps support technicians anticipate problems with individual users gaining access.

Recording

CSMon records exception events in logs that you can use to diagnose problems.

- **CSMon Log**—Like the other ACS services, **CSMon** maintains a CSV log of its own for diagnostic recording and error logging. Because this logging consumes relatively small amounts of resources, **CSMon** logging cannot be disabled.
- **Windows Event Log**—**CSMon** can log messages to the Windows Event Log. Logging to the Windows Event Log is enabled by default but can be disabled.

Notification

CSMon can be configured to notify system administrators in the following cases:

- Exception events
- Response
- Outcome of the response

Notification for exception events and outcomes includes the current state of ACS at the time of the message. The default notification method is simple mail-transfer protocol (SMTP) e-mail, but you can create scripts to enable other methods.

Response

CSMon detects exception events that affect the integrity of the service. For information about monitored events, see [Monitoring, page F-4](#). These events are application-specific and hard-coded into ACS. The two types of responses are:

- **Warning events**—Service is maintained but some monitored threshold is breached.
- **Failure events**—One or more ACS components stop providing service.

CSMon responds to the event by logging the event, sending notifications (if configured) and, if the event is a failure, taking action. The two types of actions are:

- **Predefined actions**—These actions are hard-coded into the program and are always carried out when a triggering event is detected. Because these actions are hard-coded, they are integral to the application and do not need to be configured. These actions include running the **CSSupport** utility, which captures most of the parameters dealing with the state of the system at the time of the event.

If the event is a warning event, it is logged and the administrator is notified. No further action is taken. **CSMon** also attempts to fix the cause of the failure after a sequence of retries and individual service restarts.

- **Customer-Definable Actions**—If the predefined actions built into **CSMon** do not fix the problem, **CSMon** can execute an external program or script.

CSTacacs and CSRADIUS

The **CSTacacs** and **CSRADIUS** services communicate between the **CSAuth** module and the access device that is requesting authentication and authorization services. For **CSTacacs** and **CSRADIUS** to work properly, the system must meet the following conditions:

- **CSTacacs** and **CSRADIUS** services must be configured from **CSAdmin**.
- **CSTacacs** and **CSRADIUS** services must communicate with access devices such as access servers, routers, switches, and firewalls.
- The identical shared secret (key) must be configured both in ACS and on the access device.
- The access device IP address must be specified in ACS.
- The type of security protocol being used must be specified in ACS.

CSTacacs is used to communicate with TACACS+ devices and **CSRADIUS** to communicate with RADIUS devices. Both services can run at the same time. When only one security protocol is used, only the applicable service needs to be running; however, the other service will not interfere with normal operation and does not need to be disabled. For more information about TACACS+ AV pairs, see [Appendix B, “TACACS+ Attribute-Value Pairs.”](#) For more information about RADIUS+ AV pairs, see [Appendix C, “RADIUS Attributes.”](#)