



Troubleshooting Cisco Access Registrar

This chapter provides information about techniques used when troubleshooting Cisco Access Registrar and highlights common problems.

Gathering Basic Information

Table 17-1 lists UNIX commands that provide basic and essential information to help you understand the CAR installation environment.

Table 17-1 UNIX Commands to Gather Information

UNIX Command	Information Returned
<code>/usr/bin/uname -r</code>	Solaris release level
<code>/usr/bin/uname -i</code>	Machine hardware name
<code>/usr/bin/uname -v</code>	Solaris version
<code>/usr/bin/uname -a</code>	All system information including hostname, operating system type and release, machine model and type
<code>/usr/sbin/prtconf</code>	System configuration information including memory capacity, machine type, and peripheral equipment
<code>/usr/sbin/df -k</code>	File system disk space usage including partitions, capacity, and space used
<code>/usr/bin/ps -ef</code>	Currently running processes
<code>/usr/sbin/psinfo -v</code>	Information about processors
<code>/usr/bin/pkginfo -l AICar1</code>	Software package information about CAR version number and installation directory



Note

More information about these commands and their options is available using the **man** command in a terminal window on the Sun workstation.

Troubleshooting Quick Checks

Many of the most common problems can be diagnosed by doing the following:

- Check disk space
- Check for resource conflicts
- Check the CAR log files

Disk Space

Running out of disk space can cause a number of problems including:

- Failure to process RADIUS requests
- Parts of the CAR configuration *disappearing* in **aregcmd**
- Failure to log into **aregcmd**

Check that the CAR installation partition (**\$INSTALL**) and **/tmp** are not at capacity.

Resource Conflicts

Resource conflicts are a common reason for the Cisco Access Registrar server failing to start. The most common resource conflicts are the following:

- Cisco Network Registrar is running on the CAR server
- Another application is also using ports 1645 and 1646
- A network management application is using the Sun SNMP Agent

No Co-Existence With Cisco Network Registrar

Cisco Network Registrar cannot coexist on a machine running CAR for this reason. You can determine if CNR is running by entering the following command line in a terminal window:

```
--> pkginfo | grep -i "network registrar"
```

Port Conflicts

The default ports used by the CAR server are ports 1645 and 1646. You should check to determine that no other applications are listening on the same ports as CAR.

You can check to see which TCP ports are in use by entering the following command line:

```
--> netstat -aP tcp
```

You can check to see which UDP ports are in use by entering the following command line:

```
--> netstat -aP udp
```



Note

If you configure the CAR server to use ports other than the default, you will have to specifically add ports 1645 and 1646 if you want to also use those ports.

Server Running Sun SNMP Agent

If you plan to use the CAR server's SNMP agent, you cannot use the Sun Microsystems SNMP agent that comes with the Solaris operating system.

CAR Log Files

Examining the CAR log files can help you diagnose most CAR issues. By default, the CAR log files are located in `/opt/AICar1/logs`. Table 17-2 lists the CAR log files and the information stored in each log.

Table 17-2 CAR Log Files

Log File	Information Recorded
<code>agent_server_1_log</code>	Log of the server agent process
<code>ar-status</code>	Log of CAR stop and start using the <code>arservagt</code> utility
<code>aregcmd_log</code>	Log of commands executed in <code>aregcmd</code> (very useful for tracing the steps that took place before a problem occurred)
<code>config_mcd_1_log</code>	Log of the mcd internal database
<code>name_radius_1_log</code>	Log of the radius server process
<code>name_radius_1_trace</code>	Debugging output of RADIUS request processing (only generated when the trace level, set in <code>aregcmd</code> , is greater than zero)

Using xtail to Monitor Log File Activity

A useful way of monitoring all of the log files is to run `xtail`, a utility provided with Cisco Access Registrar. The `xtail` program monitors one or more files and displays all data written to a file since command invocation.

Run `xtail` in a dedicated terminal window. It is very useful for monitoring multiple logfiles simultaneously, such as with a command line like the following:

```
> xtail $INSTALL/logs/*
```

Modifying the Trace Level

By modifying the trace level, you can gather more detailed information in the log files about what is happening in the Cisco Access Registrar server. There are five different trace levels. Each higher trace level also includes the information logged using lower trace levels. The different trace levels provide the following information:

- Level 0—No tracing occurs
- Level 1—Indicates when a packet is sent or received and when a status change occurs in a remote server (RADIUS Proxy and LDAP)
- Level 2—Information includes the following:
 - Which services and session managers are used to process

- Which client and vendor objects are being used to process a packet
- More details about remote servers (RADIUS Proxy and LDAP), packet transmission, and time-outs
- Details about poorly-formed packets.
- Level 3—Information includes the following:
 - Tracing of errors in Tcl scripts when referencing invalid RADIUS attributes
 - Which scripts have been run
 - Details about local userlist processing
- Level 4—Information includes the following:
 - Advanced duplication detection processing
 - Details about creating, updating, and deleting sessions
 - Tracing of all APIs called during the running of a script
- Level 5—Provides information about policy engine operations

Installation and Server Process Start-up

The installation process installs the CAR software to the specified installation directory and then starts the server processes. This process rarely fails but the following checks should always be performed:

- Ensure that there is an **installation success message** at the end of the **pkgadd** dialog, otherwise check the dialog for the problem
- Follow the installation instructions carefully especially when performing an upgrade. For example, when upgrading to 1.6R1, 1.6R2, or 1.6R3, a post-installation upgrade script needs to be run
- Pay attention to the information included in README files

At the end of a successful installation, **arstatus** should show the following four server processes:

```
> $INSTALL/usrbin/arstatus
AR RADIUS server running      (pid: 6285)
AR MCD lock manager running   (pid: 6284)
AR MCD server running         (pid: 6283)
AR Server Agent running       (pid: 6277)
```

If any of the above processes are not displayed, check the log file of the failed process to determine the reason. The MCD processes may fail to start if Cisco Network Registrar is installed on the same machine.

The manual method of starting and stopping the CAR processes is using the **arservagt** utility.

To start CAR processes: **arservagt start**

To stop CAR processes: **arservagt stop**

To restart CAR processes: **arservagt restart**

aregcmd and CAR Configuration

While troubleshooting, you should always use the **aregcmd trace** to turn on tracing. With tracing active, CAR generates debugging output to the log file **name_radius_1_trace**. The syntax is:

```
trace [<server>] [<level>]
```

When you do not specify a server, CAR sets the trace level for all servers in the current cluster. When you do not specify a trace level, the currently set level is used. The default trace level is 0.

Running and Stopped States

CAR can be in two states, running or stopped. In either state, all four CAR processes remain running. The state of CAR will be displayed when logging into **aregcmd** or by using the **aregcmd status** command:

```
--> status
```

```
Server 'Radius' is Running, its health is 10 out of 10\
```

The **start** and **stop** commands allow CAR to move between states. **Reload** is equivalent to a **stop** followed by a **start** if CAR is already running, and just a **start** if it is already stopped.

```
--> stop
```

```
Stopping Server 'Radius'...
Server 'Radius' is Stopped
```

```
--> start
```

```
Starting Server 'Radius'...
Server 'Radius' is Running, its health is 10 out of 10
```

```
--> reload
```

```
Reloading Server 'Radius'...
Server 'Radius' is Running, its health is 10 out of 10
```

During the transition from running to stopped, CAR stops processing new RADIUS requests and releases resources such as memory, network and database connections and open files.

During the transition from stopped to running, CAR reverses this process by opening a connection with its internal database, reading configuration data, claiming memory, establishing network connections, opening files, and initializing scripts. During this transition, problems can occur. CAR may fail to start and display the following:

```
--> reload
```

```
Reloading Server 'Radius'...
310 Command failed
```

CAR failed to move from stopped state to running:

```
--> status
```

```
Server 'Radius' is Stopped
```

This may occur for a number of reasons including the following:

- An invalid configuration
- Insufficient memory
- Listening ports already in use by another application
- Unable to open files
- Unable to initialize scripts

Check the **name_radius_1_log** file for the one of these indications.

RADIUS Request Processing

The main technique for troubleshooting RADIUS request processing in CAR is to examine the **name_radius_1_trace** log file with the trace level set to 5. Most issues are fairly self-explanatory. Some issues that can arise are:

- CAR has marked a remote server as *down*
- A resource manager has run out of resources (for example, user or group session limit has been reached or no more IP addresses are available)
- A configuration error (such as an accounting service not being set)
- A run time error in a script

Some issues are not immediately evident from the log files though, such as the following:

- Failure to save or reload CAR after a configuration change
- CAR is not listening on the correct UDP ports for RADIUS requests

Other Troubleshooting Techniques and Resources

aregcmd Stats Command

The **aregcmd** command **stats** provides statistics on request processing.

--> stats

```
Global Statistics for Radius:
serverStartTime = Tue Oct  2 10:28:02 2001
serverResetTime = Tue Oct  2 20:25:12 2001
serverState = Running
totalPacketsInPool = 1024
totalPacketsReceived = 0
totalPacketsSent = 0
totalRequests = 0
totalResponses = 0
totalAccessRequests = 0
totalAccessAccepts = 0
totalAccessChallenges = 0
totalAccessRejects = 0
totalAccessResponses = 0
totalAccountingRequests = 0
totalAccountingResponses = 0
totalStatusServerRequests = 0
totalAscendIPAAAllocateRequests = 0
totalAscendIPAAAllocateResponses = 0
totalAscendIPAREleaseRequests = 0
totalAscendIPAREleaseResponses = 0
totalUSRNASRebootRequests = 0
totalUSRNASRebootResponses = 0
totalUSRResourceFreeRequests = 0
totalUSRResourceFreeResponses = 0
totalUSRQueryResourceRequests = 0
totalUSRQueryResourceResponses = 0
totalUSRQueryReclaimRequests = 0
totalUSRQueryReclaimResponses = 0
totalPacketsInUse = 0
totalPacketsDrained = 0
```

```
totalPacketsDropped = 0
totalPayloadDecryptionFailures = 0
```

Core Files

A core file in the CAR installation directory is an indication that CAR has crashed and restarted. Check that the radius server process generated the core file using the UNIX **file** command:

```
> file core
```

```
core: ELF 32-bit MSB core file SPARC Version 1, from 'radius'
```

Check the timestamp on the core file and look for corresponding log messages in the **name_radius_1_log** file in **\$INSTALL/logs**. The word *assertion* commonly appears in core messages. Try to establish what caused the problem and contact Cisco TAC.

radclient

The CAR package provides a utility called **radclient** that allows RADIUS requests to be generated. Use **radclient** to test configurations and troubleshoot problems.

CAR Replication

For information about troubleshooting CAR replication refer to Chapter 6 of the Cisco Access Registrar Concepts and Reference Guide, Understanding Replication, (http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cnsar/1_7/referenc/repl_ref.htm).

For more information about using CAR replication, refer to [Chapter 7, “Using Replication.”](#)

