Contents

Introduction

This article discusses how to troubleshoot SNMP traps AAAAccSrvUnreachable and AAAAuthSrvUnreachable, which are triggered due to reachability issues with a Remote Authentication Dial-In User Service (RADIUS) server used to authenticate subscribers (or operators logging into the node, but that is not what is being discussed here). There are two approaches that can be used to determine when either of these traps will trigger. This article will explain what conditions trigger these traps and what troubleshooting approaches and data collection can be taken to determine root cause and solve them. It also discuss some potential remediation steps that can be considered.

Note that the RESULT of unreachability will be call failures or accounting failures, the same as if radius responses are rejections instead of acceptances. While success/failure (authentication) rate is measured independently of timeout/reachability (there are traps and alarms for this) and can certainly be analyzed in its own right, the focus of this article will be on the reachability problem and not the reject problem.

Example output from the LAB and actual tickets is used throughout to help drive home the discussions. What appears to be public IP addresses in this article are **fake** addresses.

Trap triggers

There are two different models/algorithms/approaches to choose from to determine the status of a radius server and when to try a different server if failures are occurring:

Consecutive failures in a aaamgr process approach

The original approach and the one used more often by operators involves keeping track of the number of failures that have occurred in a row for a particular aaamgr process. A aaamgr process is responsible for all radius message processing and exchange with a radius server, and many aaamgr process will exist in a chassis, each paired with sessmgr processes (which are main processes responsible for call control). (View all the aaamgr processes with the "show task resources" command) A particular aaamgr process will therefore be processing radius messages for many calls, not just a single call, and this algorithm involves tracking how many times in a row a particular aaamgr process has failed to get a response to the same request it has had to resend - an "Access-Request Timeout" as reported in "show radius counters".

The respective counter "Access-Request Current Consecutive Failures in a mgr", also from "show radius counters" is incremented when this occurs, and the "show radius accounting (or authentication) servers detail" command indicates the timestamps of the radius state change from Active to Not Responding (but no SNMP trap or logs are generated for just one failure). Here is an example for radius accounting:

If this counter reaches the value configured (Default = 4) without ever being reset, per configurable: (note the brackets [ ] are used to indicate optional qualifier and in these cases captures troubleshooting accounting (authentication is the default if accounting is not specified)

radius [accounting] detect-dead-server consecutive-failures 4

Then this server is marked "Down" for the period (minutes) configured:

radius [accounting] deadtime 10

An SNMP trap and logs is triggered as well, for example, for authentication and/or accounting respectively:

The traps indicate the server which is unreachable. Take note of any patterns. For instance, is it happening with one server or another or all servers, and what is the frequency of bouncing - is it happening continually or occasionally?

Also note that all it takes for this trap to be triggered is for one aaamgr to fail, and so the tricky part about this trap is that it does not indicate the extent of the issue. It could be very extensive or very minoir - that is up to the operator to determine, and approaches to figuring that out are discussed in this article.

show snmp trap statistics will report the number of times it has triggered since bootup, even if the older traps have long since been deleted. This example shows an accounting unreachable issue:

Note that the aaamgr reported in the above example is #231. This is the management aaamgr on ASR 5000 that resides on the System Management Card (SMC). What is deceiving in this output is that when an individual aaamgr or aaamgrs experience reachability issues, the instance number reported in the logs is the management aaamgr instance and not the particular instance(s) experiencing the issue. This is due to the fact that if many instances are experiencing reachability issues, then logging would fill up quickly if they were all reported as such, and so the design has been to report generically on the management instance, which if one did not know this, would certainly be deceiving. In the troubleshooting section further details will be provided on how to

determine which aaamgr(s) is/are failing. Starting in some versions of StarOS 17 and v18+, this behavior has been changed so that the corresponding aaamgr instance number having connectivity issues (as reported in SNMP traps) is reported in the logs with the particular id (Cisco CDETS CSCum84773), though still only the first occurrence (across mutliple aaamgrs) of this happening is reported.

The management aaamgr is the maximum sessmgr instance number + 1, and so on an ASR 5500 it is 385 for Data Processing Card (DPC) or 1153 (for DPC 2).

As a sidenote, the management aaamgr is responsible for handling operator/administrator logins as well as handling change of authorization requests initiated from RADIUS servers themselves.

Continuing, the "show radius accounting (or authentication) servers detail" command will indicate the timestamps of the state changes to Down that corresponds to the traps/logs (reminder: Not Responding defined earlier is only a single aaamgr getting a timeout, whereas Down is a single aaamgr getting enough consecutive timeouts per configuration to trigger Down)

If there is only one server configured, then it is not marked down, as that would be critical for successful call setup.

Worth mentioning is that there is another parameter that can be configured on the detect-dead-server config line called “response-timeout”. When specified, a server is marked down only when the consecutive failures and response-timeout conditions are both met. The response-timeout specifies a period of time when NO responses are received to ALL the requests sent to a particular server. (Note that this timer would be continually reset as responses are received.) This condition would be expected when either a server or the network connection is completely down, vs. partially compromised/degraded.

The use case for this would be a scenario where a burst in traffic causes the consecutive failures to trigger, but marking a server down immediately as a result is not desired. Rather, the server is only be marked down after a specific period of time passes where no responses are received, effectively representing true server un-reachability.

This method just discussed of controlling radius state machine changes is dependent on looking at all aaamgr processes and finding one that triggers the condition of failed retries. This method is subject to some degree to some randomness of failures, and so may not be the ideal algorithm to detecting failures. But it is especially good at finding aaamgr(s) that are broken while all others are working fine.

Keepalive approach

Another method of detecting radius server reachability is using dummy keepalive test messages. This involves the constant sending of fake radius messages instead of monitoring live traffic. Another advantage of this method is that it is always active, vs. with the consecutive failures in a aaamgr approach, where there could be periods where no radius traffic is sent, and so there is no way to know if a problem exists during those times, resulting in delayed detection when attempts do start occurring. Also when a server is marked down, these keepalives continue to be sent so that the server can be marked up as soon as possible. The disadvantage to this approach is that it misses issues that are tied to specific aaamgr instances that may be experiencing issues because it uses the management aaaamgr instance for the test messages.

Here are the various configurables relevant to this approach:

The command “radius (accounting) detect-dead-server keepalive” turns on the keep-alive approach instead of the consecutive failures in a aaamgr approach. In the example above, the system sends a test message with username Test-Username and password Test-Username every 30 seconds, and retries every 3 seconds if no response is received, and retries up to 3 times, after which it marks the server down. Once it gets its first response, it marks it back up again.

Here is an example authentication request/response for the above settings:

The same SNMP traps are used to signify the unreachable/down and reachable/up radius states as with the consecutive failures in a aaamgr approach:

The “show radius counters all” has a section for keeping track of the keepalive requests for authentication and accounting as well – here are the authentication counters:

Troubleshooting commands/approaches

Now that the trigger for AAA Unreachable traps has been explained, the next step is to understand the various troubleshooting commands to use to determine impact and try to figure out root cause. Unreachability is a very wide term. It doesn't explain where the unreachability is - in the network, on the server, or on the ASR. For instance, is it known whether the requests were even sent in the first place? Did the server receive the requests? Did it respond to the requests. Did the responses make it back to the ASR and if so, were they processed or dropped on the internal path (i.e. flows). This section attempt to address how to answer these questions.

Radius configuration basics

There are first some basics that one needs to be familiar with with regards to the RADIUS configuration. Most of the configuration for RADIUS is in a specifically named group, and all contexts have a default group which can be configured as follows. Many times configurations will have just one group, the default group.

If specific named aaa groups are used, they are pointed to by the following statement configured in a subscriber profile or Application Point Name (APN) (depending on the call control technology), for example:

Note: The system first checks the specific aaa group assigned to the subscriber, and then checks the aaa group default for additional configurables not defined in the specific group.

Here are useful commands that summarize all the values assigned to all the configurables in the various aaa group configurations. This allows quick viewing of all the configurables including default values without having to examine the configuration manually, and possibly help to avoid making mistakes when assuming certain settings. These commands report across all contexts:

The most important configurable is of course the radius access and accounting servers

themselves. Here is an example:

Note the max-rate feature that limits the number of requests sent to the server per aaamgr per second

In addition, the NAS IP address is also required to be defined, which is the IP address on an interface in the context from which radius requests are sent and responses received. If not defined, requests are not sent and monitor subscriber traces may not post an obvious error (no radius requests sent and no indication why).

radius attribute nas-ip-address address 10.211.41.129

Note that because both authentication and accounting are often handled by the same server, a different port number is used to differentiate authentication vs. accounting traffic on the RADIUS server. For the ASR5K side, the UDP source port number is NOT specified and is chosen by the chassis on a aaamgr basis (more on this later).

Normally multiple access and accounting servers are specified for redundancy purposes. Either a round robin or prioritized order can be configured:

radius [accounting] algorithm {first-server | round-robin}

The first-server option results in ALL requests being sent to the server with the lowest-numbered priority. Only when retry failures occur, or worse, a server is marked down, is the server with the next priority tried. More on this below.

When a radius (accounting or access) request is sent, a reply is expected. When a reply is not received within the timeout period (seconds):

radius [accounting] timeout 3

The request is resent up to the number of times specified:

radius [accounting] max-retries 5

This means that a request can be sent a total of max-retries + 1 times until it gives up on the particular radius server being tried. At this point, it tries the same sequence to the next radius server in order. If each of the servers have been tried max-retries + 1 times without response, then the call is rejected, assuming there is no other reason for failure up to that point.

As a sidenote, there are configurables that allow for users to have access even if authentication and accounting fail due to timeouts to all servers, though a commercial deployment would not likely implement this:

radius allow [accounting] authentication-down

Also, there are configurables that can limit the absolute total number of transmissions of a particular request across all the configured servers, and these are disabled by default:

radius [accounting] max-transmissions 256

For example if this is set = 1, then even if there is a secondary server, it never is attempted

because only one attempt for a specific subscriber setup is ever attempted.

show task resources facility aaamgr all

Each aaamgr process is paired with and "works for" an associated sessmgr process (responsible for overall call handling) and is located on a different Packet Services Card (PSC) or Data Processing Card (DPC) but using the same instance ID. Also in this example output note the special aaamgr instance 231 running on the System Management Card (SMC) for ASR 5000 (or Management Input Output card for ASR 5500 (MIO)) which does NOT process subscriber requests but does get used for radius test commands (see later section for more detail on that) AND for operator CLI login processing.

In this snippet, aaamgr 107 located on PSC 13 is responsible for handling all RADIUS processing for the paired sessmgr 107 located on PSC 1. Reachability problems for aaamgr 107 affects calls on sessmgr 107.

In the following example, note that problems with aaamgr 92 are affecting the paired sessmgr as easily seen when compared to other sessmgrs with respect to session counts:

show radius counters { {all | server <server IP>} [instance <aaamgr #>] | summary}

The number one command to be familiar with is varieties of "show radius counters"

This command reports back many useful counters for troubleshooting radius issues. The "show radius counters all" command is very valuable in tracking success and failures on a server basis, and it is important to understand the meaning of the various counters that compose this command, as it may not be obvious. The command is context-sensitive and so must be run in the same context where the aaa group(s) are defined.

Important note: Over an un-monitored time period, it is difficult to draw any conclusions from the counter values or the relationships amongst counters. To make accurate conclusions, the best approach is to reset the counters and monitor them over a period of time when the issue being troubleshot is occurring.

In the following output, note "Access-Request Sent" = 1, while "Access-Request Retried" = 3. So, any given new request to a particular radius server is only counted once, and all the retries are counted separately. In this case, that is a total of 3 + 1 = 4 access requests sent. Note the counter "Access-Request Timeouts" = 1. A single timeout occurs only when ALL the retries fail, so in this case, 3 retries without a response result in 1 Timeout (not 4). This happens across all of the configured servers until there is success, or all attempts have failed. So pay attention to the counters that are tracked for each server separately. Here is an example of this, where:

Note also that timeouts are NOT counted as failures, the result being that the number of Access-Accept received and Access-Reject received will not add up to Access-Request Sent if there are any timeouts.

Analysis of these counters may not be completely straightforward. For example for Mobile IP (MIP) protocol, as the authentications are failing, there is no MIP Registration Reply (RRP) being sent, and the mobile may continue to initiate new MIP Registration Requests (RRQ) because it

has not received a MIP RRP. Each new MIP RRQ causes the PDSN to send a new Authentication request which itself can have its own series of retries. This can be seen in the Id field at the top of a packet trace – it is unique for each set of retries. The result is that the counters for Sent, Retried, and Timeout can be much higher than expected for the number of calls received. There is an option that can be enabled to minimize these extra re-tries, and it can be set in the Foreign Agent (FA) (but not on the Home Agent (HA)) service: “authentication mn-aaa <6 choices here> optimize-retries”

Some other useful counters:

"Access-Request Response Dropped" - occurs if the call fails to setup while waiting for responses to authentication requests.

"Access-Request Response Last Round Trip Time" - indicates any delays between the endpoints, though it obviously would not indicate where the delay might be.

"Access-Request Current Consecutive Failures in a mgr" relates to what was discussed in the first section on triggers for AAA Unreachable traps. It represents the aaamgr(s) with the highest count of consecutive timeouts.

"Current Access/Accounting-Request Queued" indicates requests that are not being responded to and remaining in the queue (accounting allows for a build-up of the queue indefinitely while authentication does not)

The most common scenario seen when AAA Unreachable is reported is that Access Timeouts and/or Response Drops are also occuring, while Access Responses are not keeping up with requests.

If access to priviledged technical support mode is available, then further investigation can be done at the aaamgr instance level to determine if one or more specific aaamgrs are the cause of the increase in overall "bad" counts. For example, look for aaamgrs that are located on a specific PSC/DPC having high counts or maybe a single aaamgr or random aaamgrs having issues - look for patterns. If all or most aaamgrs are having issues, then there is increased likelihood that the root cause is either external to the chassis OR manifesting large-scale on the chassis. General health checks should be done in that case.

Here is example output showing an issue with a specific aaamgr for accounting. (The issue turned out to be a bug in a firewall between the ASR5K and the RADIUS server that was blocking traffic from a specific aaamgr instance (114) port). Over a three week period, only 48 responses have been received, yet over 100,000 timeouts have occurred (and that doesn’t include re-transmits).

In conclusion, determine which counters are incrementing, for which servers, and at what speed.

show session subsystem facility {aaamgr | sessmgr} {all | instance <instance #>}

While it is beyond the scope of this article to examine all the superfluous output from this command, a couple examples are worth looking at. Like any other troubleshooting, comparing the output between what is believed to be good versus bad aaamgr instances often reveals obvious differences in the values reported. This could be reflected in the total number of requests, failure/success rate, auth cancelled, etc. As a reminder, be sure to clear the session subsystem (one instance cannot be cleared, they all must be cleared) so as to eliminate any history which could potentially provide a clouded picture of the current state.

Continuing with the same issue mentioned earlier with respect to a single aaamgr failing for accounting, here is output from a different node with that same issue except a different sessmr instance 36. Note all the interesting fields for the failing aaamgr and how those values increase over time with the two captures of the command. Meanwhile output from instance 37 is shown as

an example of a working aaamgr.

One should also run show task resources to check for any uneven session counts (used column) amongst all sessmgrs. If any are found, check the paired aaamgrs for those sessmrs with this command to see if there are any fields that are out of line - if the issue is due to RADIUS then there is a good chance to find something.

In the show task resources example in a previous section, there was a signficantly lower session count on sessmgr 92 which was paired to aaamgr 92. The output from show session subsystem shows significant increase in the total max-outstanding and aaa auth purged counters, and elevated Current max-outstanding counters. One can use the grep feature live on the chassis and/or Notepad++ or other powerful search editor to quickly analyze data. Run the command multiple times to see what values are increasing or remaining elevated:

ping

traceroute

An ICMP Ping tests basic connectivity to see if the AAA server can be reached or not. The ping may need to be sourced with the src keyword depending on the network and needs to be done from the AAA context to have value. If ping to the server fails, then try pinging intermediary elements including the next hop address in the context, confirming there is an ARP entry to the next-hop address if ping fails. Traceroute can also help with routing issues.

radius test instance x auth {radius group <group> | all | server <IP> port <port>} <username> <password>

radius test instance x accounting {radius group <group name> | all | server <IP> port <port>}

With access to the Tech Support Test commands, one can further test whether a specific aaamgr is able to reach any RADIUS server. For a basic RADIUS connectivity test, independent of any specific aaamgr instance, use the generic version of this command which doesn't specify any specific instance # but uses the management instance by default. If this fails, then it may point to a wider issue independent of specific instances.

This command sends a basic authentication request or accounting **start** and **stop** requests and waits for a response. For authentication, use any username and password, in which case a reject response would be expected, confirming that RADIUS is working as designed, or a known working username/password could be used, in which case an accept response should be received

Here is an example output from monitor protocol and running the authentication version of the command on a lab chassis: Here is an example from a live chassis:

Here is an example output from running the accounting version of the command. A password is not needed.

The following output is for the same aaamgr instance 36 just mentioned where connectivity to a specific RADIUS accounting server is broken:

```
[source]PDSN> radius test instance 36 accounting all test
```

```
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

show radius info [radius group <group name>] instance { X | all}

This command reports the Network Processor Unit (NPU) flow ID and UDP port used by the configured NAS IP address to connect to RADIUS servers. This is reported in the aaa group default section of the output. Certainly the port number can be useful if one needs to match RADIUS packets in a packet capture with a specific aaamgr instance #. (Note that NPU flows are complicated and not something discussed in this article but an entity that a support engineer would be able to investigate further.) It also tracks outstanding requests to the server. In the same example issue used throughout this article, only a specific RADIUS server <==> NAS IP / UDP port pair had failed as highlighted.

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014
```

```
RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

monitor subscriber

Monitor subscriber can be used to determine if authentication is at least attempted and whether a reply is being processed for the calls being monitored. Turn on option 'S' which stands for Sessmgr Sender Info - effectively reporting on the sessmgr or aaamgr instance # that is handling the messaging in question. Here is an example for a MIP call on an HA attaching to sessmgr / aaamgr instances 132.

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
```

```
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

There is a failure example at the end of this article as well.

Packet Capture

Sometimes there is not enough information on the ASR to determine why reachability issues are occuring, in which case a packet capture will be necessary. When troubleshooting individual subscriber issues, identifying the respective packets in a trace should be easy. Otherwise, knowing the UDP port being used at either end of a particular aaamgr instance # <==> RADIUS server pair could be helpful if the issue is tied to specific ports/aaamgr instances. Attempting capture at multiple places in the network may be necessary to determine where packets are getting dropped. In the issue being analyzed throughout this article, it was a packet capture in just the right place in the transport path between the ASR and the RADIUS server that was the break-through in solving the issue.

Remediations

This last section offers some ideas for remediating RADIUS connectivity issues. These are not presented in any particular order but rather simply a list to consider in the troubleshooting process.

If the RADIUS server is getting overloaded, the load could be decreased via the value (default 256) configured for “radius (accounting) max-outstanding”, which sets a limit on the number of

outstanding (unanswered) requests for any given aaamgr process. If the limit is reached, logs may indicate this: “Failed to assign message id for radius authentication server x.x.x.x:1812”.
Rate-limiting RADIUS messages to specific servers may also help reduce load via the rate-limit keyword for the respective server configuration lines.

Sometimes it is not a problem of connectivity but of increased accounting traffic, which is not a problem with RADIUS persay, but pointing to another area, such as increased ppp renegotiations which are causing more accounting starts and stops. So one may need to troubleshoot outside of RADIUS to find a cause or trigger for the symptoms being observed.

If during the troubleshooting process it has been decided to remove a radius authentication or accounting server from the list of live servers for whatever reason, there is a (non-config) command that will take a server out of service indefinitely until it is desired to put it back in service. This is a cleaner approach than having to remove it from the configuration manually:

```
{disable | enable} radius [accounting] server x.x.x.x
```

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
```

```
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

A PSC or DPC migration or a line card switchover can often clear problems due to the fact that the migration results in the restart of the processes on the card, including the npumgr which has been the cause of problems from time to time with regards to NPU flows.

But in an interesting twist with the aforementioned example of aaamgr 92, the AAA Unreachable failures actually STARTED when a PSC migration was done. This was triggered due to an NPU flow going missing when a PSC migration was done making PSC 11 standby. When it was made active an hour later, the actual impact of the missing flow started for aaamgr 92. Issues like this are very difficult to troubleshoot without assistance from Technical Support.

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms
```

```
RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

The issue was temporarily resolved with a port switchover which caused the PSC card which had a missing NPU flow for aaamgr 92 to no longer be connected to an active line card.

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

The last failure trap:

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
```

```
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

Similarly, restarting specific aaamgrs that get "stuck" may also resolve issues, though this is an activity that Technical Support should do since it involves restricted Tech Support commands. In the aaamgr 92 example introduced in the show task resources section earlier, this was attempted but did not help because the root cause was not aaamgr 92 but rather the missing NPU flow that aaamgr 92 needed (it was an NPU issue, not a aaamgr issue). Here is relevant output of the attempt. "show task table" is run in order to show the association of process id and task instance # 92.

```
[source]PDSN> radius test instance 36 accounting all test
Wednesday September 10 10:06:29 UTC 2014

RADIUS Start to accounting server 209.165.201.1, port 1646
Accounting Success: response received
Round-trip time for response was 51.2 ms

RADIUS Stop to accounting server 209.165.201.1, port 1646
Accounting Success: response received
```

```
Round-trip time for response was 46.2 ms

RADIUS Start to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 89.3 ms

RADIUS Stop to accounting server 209.165.201.2, port 1646
Accounting Success: response received
Round-trip time for response was 87.8 ms


RADIUS Start to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Stop to accounting server 209.165.201.3, port 1646
Communication Failure: no response received

RADIUS Start to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 81.6 ms

RADIUS Stop to accounting server 209.165.201.4, port 1646
Accounting Success: response received
Round-trip time for response was 77.1 ms

RADIUS Start to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Stop to accounting server 209.165.201.5, port 1646
Accounting Success: response received
Round-trip time for response was 46.7 ms

RADIUS Start to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 79.6 ms

RADIUS Stop to accounting server 209.165.201.6, port 1646
Accounting Success: response received
Round-trip time for response was 10113.0 ms
```

Final Example

Here is a final example of a real outage in a live network that pulls together many of the troubleshooting commands and approaches discussed in this article. Note that this node handles 3G MIP, and 4G Long Term Evolution (LTE) and evolved High Rate Packet Data (eHRPD) call types.

show snmp trap history

By the traps alone, it can be confirmed that the starting point matches with what the customer reported as 19:25 UTC. As an aside, note that **AAAAuthSvrUnreachable** traps for primary server 209.165.201.3 didnt start happening until hours later (not clear why, but good to note; but **accounting unreachable** to that server started right away)

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
```

```
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

show task resources

The output shows a much lower count of calls on DPC 8/1. Based on this alone, without any further analysis, one COULD suggest that there is an issue on DPC 8 and propose the option to migrate to the standby DPC. But it is important to acknowledge what the actual subscriber impact is - in these scenarios typically the subscribers will connect successfully on a subsequent attempt and therefore impact is not too significant for the subscriber and they likely will not report anything to the provider, assuming that there is no user-plane outage also going on (which is possible depending on what's broken).

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

monitor subscriber

A call setup was caught where there was no response to the authentication request to

primary 209.165.201.3 for sessmgr 242 on DPC 9/1 which happens to have its paired aaamgr residing on DPC 8/1, confirming 3G failures due to AAA unreachable on 8/1. It also confirms that even though there hadn't been any AAAAuthSrvUnreachable traps for 209.165.201.3 up to that point in time, it doesn't mean that there isn't a problem for handling responses for that server (as shown above, traps do start but hours later).

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

show sub [summary] smgr-instance X

What is interesting is that the session count for sessmgr 242 is similar to other working sessmgrs. Further investigation showed that 4G calls, also hosted on this chassis, were able to connect and so they made up for the lack of 3G Mobile IP calls being able to connect. It can be determined that going back as far as 8 hours which was after the outage has started, the are no MIP calls for this sessmgr 242, while going back 9 hours to before the outage started, there are connected calls:

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

```
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

LTE and eHRPD calls show a higher ratio to MIP calls when comparing sessmgrs that are connected to working and broken aaamgrs:

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

radius test instance X authentication server

All aaamgrs on 8/1 are dead – no radius test instance commands work for any of those aaamgrs but do work for aaamgrs on 8/0 and other cards:

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

```
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

show radius counters all

The flagship command for troubleshooting RADIUS shows lots of timeouts that are increasing quckly:

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```

Remediation

During the maintenance windows, a DPC migration 8 to 10 resolved the issue, the AAAAuthSrvUnreachable traps stopped, and DPC 8 was RMA'd and the root cause was determined to be a hardware failure on DPC 8 (details of that failure are not important to know for the purposes of this article).

```
Sun Dec 29 19:28:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 5 ip
address 209.165.201.3
Sun Dec 29 19:32:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:33:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 43 (AAAAccSvrReachable) server 5 ip address
209.165.201.3
Sun Dec 29 19:34:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 2 ip
address 209.165.201.3
Sun Dec 29 19:35:05 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 2 ip address
209.165.201.3
Sun Dec 29 19:38:13 2013 Internal trap notification 42 (AAAAccSvrUnreachable) server 6 ip
address 209.165.201.8
```

```
...
Sun Dec 29 23:12:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:13:03 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:54:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:54:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
Sun Dec 29 23:58:13 2013 Internal trap notification 39 (AAAAuthSvrUnreachable) server 4 ip
address 209.165.201.3
Sun Dec 29 23:58:14 2013 Internal trap notification 40 (AAAAuthSvrReachable) server 4 ip address
209.165.201.3
```