Interlink Networks Services, LLC.

Software Version 8.3.1 Release Notes

May 2016

Interlink Networks Services, LLC.
2531 Jackson Road
Suite 306
Ann Arbor, MI 48103-3818
734-821-1200 (tel), 734-821-1235 (fax)
www.interlinknetworks.com

FILE CONTENTS

8.3.1 RELEASE NOTES

NEW FEATURES in 8.3.1:

1. Add support for a new "-syslogfacility <facility>" radiusd command-line parameter. If logging to syslog has been enabled and if “-syslogfacility” is specified, all syslog messages are directed to the specified facility. if not specified, then some syslog messages are directed to the auth facility and others are directed to the daemon facility.

   The supported facility values are:
   local0, local1, local2, local3, local4, local5, local6, local7, kern, user, mail, daemon, auth, syslog, lpr, news, uucp, cron

2. Add support for a new "-syslogtimestamp on/off" radiusd command-line parameter, which controls whether to add the logfile's timestamp+loglevel information to the syslog lines. Default=ON.

CHANGES in 8.3.1:

1. Add support for Arista (vendor number 30065) VSAs.

2. Add support for Ciena (vendor number 1271) VSAs.

3. Add support for Infoblox (vendor number 7779) VSAs.

4. Add support for Mikrotik (vendor number 14988) VSAs.

5. Add support for Nortel (vendor number 562) VSAs.

6. Changes were made to the advanced policy's LOG command output written to the logfile.

   a) LOG output is now outputted, as much as possible, onto a single logfile line. Individual AVPs are no longer displayed one-per-line.
   b) LOG command output line length only limited by global logfile line length limit.
   c) Removal of "decisionfile://<filename>(line <n>, character <n>): " header from LOG output line(s).
   d) Removal of comma following the “<log-message>” text of the LOG command:
      LOG “<log-level>” “<log-message>”, <attr-spec>, ... <attr-spec>

7. The Solaris RAD-Series server now uses the OpenSSL libraries compiled using the Solaris Studio 12.4 which allows using the hardware acceleration for encryption.

FIXES in 8.3.1:

- None

REMOVED FEATURES in 8.3.1:

- None

KNOWN ISSUES in 8.3.1:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /`opt/aaa` by default.
2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of `radiusd` does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The `iaaaAgent.conf` file in the configuration directory does not control the attempt to reconnect currently.

4. Server Manager issues:
   a) There is a problem with Java JRE 1.8.0_60 through 1.8.0_71 which causes Internet Explorer 11 to occasionally insert a delay of about 20 seconds when an applet is called. This issue was fixed with Java JRE 8u72. This does not affect the Firefox browser. This affects some Server Manager "Edit Configuration" screens.
   b) As of September 2015 the Chrome browser no longer supports Java plugins, and thus will not work with the Interlink Server Manager.
   c) The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

5. TLS stateless session resumption fails if doing EAP-TLS or EAP-TTLS and using the Windows 8 or 8.1 supplicant. TLS stateless session resumption is, by default, disabled.

8.3.0 RELEASE NOTES

NEW FEATURES in 8.3.0:

1. Support for WiMAX (vendor 24757) Vendor Specific Attributes (VSAs) has been greatly expanded in dictionary definition, user profile configuration, advanced policy manipulation and session accounting (Livingston CDR format) logging.
   a. The WiMAX VSA format including the flags field is fully supported.
   b. Fragmentation/reassembly of long WiMAX values is supported using the Continuation Flag of the WiMAX VSA header.
   c. Salt encrypted WiMAX attributes are supported.
   d. WiMAX attributes which can contain either an IPv4 or IPv6 address are supported.
   e. WiMAX TLVs and sub-attributes are supported.
   f. The WiMAX dictionary is based on "WiMAX Forum Network Architecture, Detailed Protocols and Procedures, Base Specification, WMF-T33-001-R022v02, WMF Approved, (2014-04-18)"
   g. Support for WiMAX TLVs, with no limitation on the sub-attribute nesting level other than the limitation imposed by the size of the length field (8 bits) of the parent TLV. TLVs and their sub-attributes can be configured in user's profiles, used in advanced policy files, logged in Livingston accounting records and packed/unpacked in RADIUS messages.
   h. Support for WiMAX attributes whose value is an 8-bit integer or a 16-bit integer.
   i. Fragmentation/reassembly of long (requiring two attributes) salt encrypted WiMAX attributes is supported.
   j. Response message occurrence rules are not supported on the sub-attribute level, nor the complex relationships between sub-attributes.
   k. Length-checking of WiMAX attributes which have a complex (structured) value is not supported.

2. Support has been added for new Attribute Value Pair (AVP) formats, as specified in IETF RFC 6929, in dictionary definitions, user profile configurations, advanced policy manipulations and session accounting (Livingston CDR format) logging.
   a. Extended-Type attributes are supported.
   b. Long-Extended-Type attributes are supported.
   c. Extended-Vendor-Specific attributes are supported.
   d. The TLV data type is supported. TLVs and their sub-attributes can be configured in user's profiles, used in advanced policy files, logged in Livingston accounting records and packed/unpacked in RADIUS messages.
   e. The 64-bit integer data type is supported.
   f. Handling of unknown attribute types and invalid attribute values is supported, i.e. such attributes can proxied and logged.
   g. Attributes defined in RFC6929 "RADIUS Protocol Extensions" have been configured in the dictionary.

3. Implemented support for new attribute types Some defined by WiMAX specification:
   a. A new IP address attribute which can hold either an IPv4 address or an IPv6 address, as indicated by its length, type **ip46addr**.
   b. A new attribute which holds a signed 32-bit integer value, type **signed32**.
   c. A new attribute which holds a signed 64-bit integer value, type **signed64**.
   d. A new attribute which holds an unsigned 64-bit integer value, type **unsigned64**.

4. Implemented generic support for fragmentable attributes, such as EAP-Message and Unisphere-DHCP-Options. Such attributes are concatenated into a single occurrence with a long value when received, and fragmented (if necessary) into multiple occurrences when transmitted. An attribute is marked as fragmentable (i.e. concatenate-on-receive/fragment-on-transmit) via a new **FRAGMENTABLE** flag in the dictionary. Standard RADIUS attributes (e.g. EAP-Message) as well as VSAs (e.g. Unisphere-DHCP-Options) can be fragmented/defragmented.
    - The **FRAGMENTABLE** flag is disallowed for attributes of type **tag-str**. If configured, the flag is discarded and the attribute is retained.
    - The **FRAGMENTABLE** flag is ignored, if configured for a TLV sub-attribute.

5. Implemented support for the new **errorlog** file, which facilitates recognition of significant events and exceptions in processing without the need of searching lengthy logfiles.
    a. There is a new command-line startup switch "`-errorlog [on|off]`" which defaults ON. If ON, the server will open a new errorlog file, named errorlog, in the same directory as the logfile resides.
    b. There is a new aaa.config parameter "`maximum-errorlog-file-size N`" which specifies the maximum size, in bytes, of the errorlog file. The min/default/max values are 64KB/2GB/2GB.
    c. The errorlog file rolls over when it gets close to (within 8KB of) the configured maximum-errorlog-file-size value. When rolled over, the existing full errorlog is renamed to errorlog.old, and a new errorlog is opened.
    d. The server constantly appends to errorlog. Server restarts leave any pre-existing errorlog intact (the server opens errorlog for "append").
    e. There are two ways that messages are selected for writing into the errorlog:
        - The server has an internally-defined set of informational messages that are written to both the logfile and errorlog. In addition, there is a new aaa.config parameter "`errorlog-level <level>`", which configures the log level(s) of messages which are logged to both the logfile and the errorlog.
        - The `<level>` parameter is a string indicating the log levels. The default value is "`ACEWN`", which means that, by default, (A) (C) (E) (W) and (N) messages are logged to the errorlog. The <level> parameter can specify any subset of the characters 'A' 'C' 'E' 'N' and 'W'. For example,
        `errorlog-level ACE  ## errorlog only (A) (C) and (E) level messages.`

6. Configured, in the dictionary, RADIUS attributes defined in the following RFCs:
    - *RFC 6519 "RADIUS Extensions for Dual-Stack Lite"*
    - *RFC 6911 "RADIUS Attributes for IPv6 Access Networks"*
    - *RFC 6930 "RADIUS Attribute for IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)"*
    - *RFC 4675 "RADIUS Attributes for Virtual LAN and Priority Support"*

7. Added support for the handling of received unknown (i.e. not in dictionary) attributes, as recommended in RFC6929. Received unknown attributes are preserved and can be duplicated, proxied, echoed, logged, etc... Previously the 8.2 version of the server discarded unknown received attributes and versions prior to that would preserve and proxy/echo/log unknown VSAs from requests, but would discard unknown RADIUS attributes from requests and would discard all unknown attributes received in responses. The pre8.2 server would also sometimes incorrectly double-encapsulate an unknown VSA when transmitting it.

8. Added support for a new "Roaming-Accounting on/off" aaa.config parameter. The default is OFF, which means the expiration of previous hops/subsessions of a roaming session is disabled, and the server's previous behavior does not change. This feature allows session tracked sessions to be cleaned up where users roam from one AP to another.

9. Added support for new parameters in the aatv.ProLDAP{} block:
    - `TCP-Keepalive,`

- `TCP-Keepalive-Idle`,
- `TCP-Keepalive-Interval`,
- `TCP-Keepalive-MaxCount`

The parameters `TCP-Keepalive-Idle`, `TCP-Keepalive-Interval`, and `TCP-Keepalive-MaxCount` are only configurable in Linux implementations. In Solaris implementations, the system-defined values of these parameters are used and are not overridable by the RAD-Series server.

10. Added a new logfile message, which identifies the OpenLDAP, OpenSSL and SNMP versions that the RAD-Server was built with. This message appears exactly once, at startup time. The message is also included in the errorlog file.

    For example:

```
(I): OpenLDAP version: "2.3.20". OpenSSL version: "OpenSSL 1.0.1q 3 Dec 2015".
     [proldap_init]
(I): NET-SNMP Version "5.4.3"  [agent_init]
```

11. Implemented a new built-in **RADLOG** plug-in, which can be inserted into the FSM for the purpose of generating a logfile line when invoked.

12. Implemented a new built-in **CANCEL_PROXY** plug-in which is callable from radius.fsm.

13. Changes to server's logging to logfile:
    a. Enhanced the logfile parsing error messages for bad attributes in the users and realm files. The message now indicates that the user's profile has been discarded.

    b. Adding the license start/end date to the logfile. This update changes the server's "Server started..." logfile line

       From:

```
(I): Server started, ready to process requests
```

       To: e.g.:

```
(I): Server started, ready to process requests. \
     License-End-Date(2016-01-10/01:13:03)
```

    c. Add support for a "Configuration Summary" in the logfile and in errorlog. This summary appears just before the "Server started" message at startup time, and also appears after each HUP. There is a summary line for each config file, giving the filename, number of errors, and a synopsis of the file, e.g.:

```
(I): Configuration Summary:
(I):   dictionary : 0 errors. 2088 attributes and 2437 values read from 38
dictionary files
...
```

    d. Changed server logging so that, following a HUP, the server reports the "`Server is listening` ..." and "`Server is proxying from`..." logfile messages for all ports, whether their configuration has changed or not.

e. Improvements to the server's "next state not found" message, clarifying the text, and indicating the server's action.

f. Add reporting of OS type in first line of logfile, e.g.:

(I): Version 8.3.0 (Solaris), licensed software

3. Changes to **radcheck**:

a. Added the license end date to radcheck output. Here is a sample line as might appear in a radcheck output:

```
Version 8.3.0 (Solaris), Debug-Level(0), errorlog(enabled), \
License-End-Date(2017-01-01/00:00:00)
```

b. Change to radcheck output, to indicate if error logging is enabled or disabled. See NEW FEATURES #6 for details about errorlog.

c. Add to the radcheck output a list of configuration files which had errors, e.g.:

```
Configuration errors: aaa.config(3) dictionary(2)
```

d. Change radcheck to report the OS type when reporting the server's version, e.g.:

```
Version 8.3.0 (Solaris), Debug-Level(0), errorlog(enabled) …
```

e. Change the default setting for the aaa.config "`radcheck <value>`" parameter, from "`radcheck +mf+queues`" to "`radcheck +queues`". This means, by default, the malloc/free counts will not be produced by the 8.3.0 server when generating radcheck output.

f. The radcheck output has been reorganized and labels changed to make the output easier to understand.

4. Changed the server's handling of the situation where an Accounting-Request is received which is missing the Acct-Status-Type attribute.

Previously, the server produced a "next state not found" logfile error, and discarded the accounting request.

Now the server will generate an "Acct-Status-Type=Unknown-Type" attribute, and append this attribute to the accounting authreq. The accounting message can be logged if desired. The new manufactured "Acct-Status-Type=Unknown-Type" attribute will not be proxied. The "Unknown-Type" is a new VALUE added to the dictionary. When an Accounting-Request is received, check for the presence of the Acct-Status-Type and Acct-Session-Id AVPs whether type=NAS or type=Proxy.

5. Add a list of the server's directories to logfile and errorlog, right after the "Server Startup command". Also add a line of information to the logfile and errorlog, which provides information on the OS/machine on which the RAD-Server is running, e.g.:

```
(I): OS information: sysname('SunOS') nodename('t33') release('5.10')
version('Generic_141444-09') machine('sun4v')
```

6. Enhance advanced policy code, to now support date constants, e.g.:

```
insert Event-Timestamp = "Dec 27 2015/00:00:00"
```

CHANGES in 8.3.0:

1. Upgraded OpenSSL to 1.0.1q. This incorporates the fix for the "Logjam" vulnerability as well as fixes for many other OpenSSL vulnerabilities.

2. Change the parsing of integer attribute values to check for overflow and invalid characters.

3. Renamed dictionary attribute type "**octet**" to "**unsigned8**". Renamed dictionary attribute type "**short**" to "**unsigned16**". This makes their names consistent with the new types: **signed32**, **signed64**, and **unsigned64**.

4. Added a check to prevent the same attribute name from being used by different vendors.
   - For duplicate attribute names, discard the duplicate and continue on.
   - For orphan named values, discard the value and continue on.

5. Change dictionary parsing so that attribute data types ("**integer**", "**string**", etc.) are now case-insensitive, leading whitespace is allowed and trailing comments following a # are allowed after dictionary entries.

6. Change the code which reads in the dictionary at startup time to set the ENCAPS/NOENCAPS based on an algorithm rather than based on the ENCAPS/NOENCAPS configuration flag in the dictionary.

   The dictionary's ENCAPS/NOENCAPS flags have been removed. If encountered these flags will be ignored.

   The algorithm is as follows:
   - All RADIUS attributes are NOENCAPS.
   - All VSAs that are not mapped are ENCAPS.
   - VSAs that are mapped are NOENCAPS.

7. Changed the dictionary parsing so that errors that were formerly fatal (server would terminate) are now handled by discarding the flawed dictionary entry, logging an error message which indicates the issue and the server's action, and continuing on. Also changed some errors that were silently ignored to be reported.

8. After reading in the vendors mapping table(s) and the dictionary, the server checks that all vendor mapped attributes are defined in the dictionary. If a vendor mapped attribute is not defined in the dictionary, a logfile message will be generated and the mapping of the undefined attribute will be removed from that vendor's mapping table.

9. Changed all the advanced policy code's avpair value buffers to be of size 4096, system-independent.

10. Changed the default maximum length of a logfile line, excluding the timestamp header, to be 4096 characters on both Solaris and Linux. The Solaris size changed from 1024 to 4096 and Linux is unchanged at 4096.

11. Changed the handling of escape sequences for string attributes: Non-printable characters can be configured using an escape sequence which consists of '\x' and two hex characters (e.g. "\x7F"). The server no longer allows a single hex character. If a users file or policy file entry contains an escaped x ("\x") which is not followed by two hex digits, this is considered a syntax error and the user entry or policy are discarded.

12. Changes to handling of **session.las** file:
    - Every time session.las is written, the previous session.las (if it exists) is renamed to session.las.old, rather than overwritten.
    - Remove the code which bypasses the writing of session.las if no changes have occurred since the last checkpoint.

13. Moved all of the pre-defined **%EVENT** definitions from the FSM files into the server.

14. Changed the "`%enable_ingress_egress_policy YES`" from an indication that the FSM runs the Ingress/Egress policies to an action statement that tells the server to run the polices or to not run them.

15. Removed inappropriate/misleading logfile message which was added in version 8.2, and which appears when proxying EAP-Authentications. The message is this:

    "`(N) eap_State_match(): No match for State ...`"

16. Change the logfile message which is generated when encountering an unknown attribute, to include the possibility of a non-fully-qualified sub-attribute name.

17. Disallow **tag-str** and **tag-int** attributes for vendor WiMAX. The attribute is discarded.

18. Changes to management of user-defined `%event` names/codes, to solve some known issues.
    a. The server no longer allows two event names to share the same event code. The mapping between event names and event codes is one-to-one.
    b. User-defined `%event`s must have event codes in the range 100-4000, whether specifically-assigned in the FSM, or assigned by the server. Previously the server allowed full 32-bit user-defined %event code values, even though the server only stored the names of the event codes which were < 4096.
    c. The logfile displays the event name and event code of all user-defined %events.
    d. The server has SDK-defined constants for these values:
        - 100 = The minimum (base) value for user-defined %event codes.
        - 4000 = The maximum value for user-defined %event codes.
        - 4096 = The maximum value for an event code, whether internally-defined or user-defined. This is the size of the table which saves the event names.

19. Change the maximum line length of configuration files, which previously had a mix of maximum line lengths. Now all configuration files except the ***users** files have a max line length of 4096. This is not configurable, and was not configurable before.

    The line length of the *users file is, as before, configurable. Prior to this update, the min/default/max was 1K/16K/2GB. With this update, the min/default/max has been changed to 1K/16K/16K.

7. Change to allow an empty userid, e.g. a User-Name of "`@outerrealm.com`". This satisfies RFC4372, which says: "*Some authentication methods, including EAP-PEAP, EAP-TTLS, EAP-SIM and EAP-AKA, can hide the true identity of the user from RADIUS servers outside of the user's home network. In these methods, the User-Name attribute contains an anonymous identity (e.g., @example.com) sufficient to route the RADIUS packets to the home network but otherwise insufficient to identify the user. ...*"

8. Changing the log-level of these two logfile messages, from (I) to (N).

```
(N): proldap_close: Closing connection to LDAP server '<...>' as '<...>'
(N): get_open_result: Connected to LDAP server '<...>' as '<...>', socket(<n>)
```

9. Changing the log-level of these logfile messages, which appear when OpenSSL generates a TLS alert (and hence failing the authentication), from (I) to (N):

```
(N): ProcessHandshake PEAP: AAA Server generated TLS alert: '<description>'
(N): ProcessHandshake TTLS: AAA Server generated TLS alert: '<description>'
(N): ProcessHandshake TLS: AAA Server generated TLS alert: '<description>'
```

10. If the server encounters an attribute which is too-long-to-pack, the response message is dropped. Previously, the offending attribute was ignored, and the response was sent (minus the offending attribute).

11. Changed the code which proxies a request message: If the server encounters an attribute which is too-long-to-pack, the request message is dropped. Previously, the offending attribute was ignored, and the request was sent (minus the offending attribute).

12. Also changed the handling of duplicate requests, so that if the response or proxied request was discarded due to an attribute which is too-long-to-pack, the authreq is left to quietly time out, after logging a retry.

13. Change the logfile line that is generated when a response is sent, from LOG_DAEMON to LOG_AUTH.

14. Changed the logging of the "Server directories" and "Configuration Summary" block and the logging of a message by routine file_init() from LOG_AUTH to LOG_DAEMON.

15. If an attribute name exceeds 64 characters (which is the current limit of the RAD-Series server's shared memory interface), it is reported in the logfile.

16. Changed the advanced policy command which logs all instances of an avp (e.g. "`log 'NOTICE', Reply-Message[*]`") to log each occurrence on a separate line, rather than multiple occurrences per logfile line. This makes for better readability.

17. Changed the dictionary parsing code so that an attribute name is allowed to contain only these characters: `A-Z, a-z, 0-9, -` (dash), and `_` (underscore), which is the set of characters that advanced policy allows in an attribute name.

18. Changed the dictionary parsing code so that a named attribute value whose name consists of only numeric digits is disallowed. Also updated the current dictionaries so that no named attribute value has a name consisting of only numeric digits.

19. Increased advanced policy parser's internal stack from 500 to 1000 commands.

20. Change expiration time for unresolvable clients so that the RAD-Server retries a DNS lookup after 20 minutes, rather than 15 minutes as before.

21. Changed the log level of the "`rad_init: No authentication listen ports were opened`" logfile message from (W) to (N) to be consistent with the

    "`(N): rad_acct_init: No accounting listen ports were opened`" message.

22. Changed log-level of "`(E) get_open_result: ERROR <n> connecting to LDAP server '<url>' as '<xxx>' - <msg>`", from (E) to (W).

23. Changed log-level of "`(I) IPv6 support is ENABLED/DISABLED`" logfile message from (I) to (N).

24. Changed the server to mask the value of an ipv6prefix attribute when the attribute is configured with "**MASK_VALUE**" in the dictionary.

25. Server Manager changes:
    a. Upgrade the Java VM installed with the Server Manager to 7 update80. The Server Manager now runs with Java 7u80, Tomcat 7.0.67, and Java servlet package version 3.0.
    b. In the **Administartion:Start** Options, added control for enabling/disabling the errorlog file.
    c. Changed the earliest year in the Server Manager's YEAR dropdown list to N-2, where N is the current year. This offers the ability to view logfiles for the current year and the two years prior to that. This YEAR dropdown appears in the **Maintenance->Logfile** and **Maintenance->Statistics** screen.
    d. Changed the start time / end time controls in the Maintenance screens, so that MONTH, DAY, and YEAR each have their own dropdown.
    e. Changes to the Server Manager's handling of LDAP realm configurations:
        - Change to discard unknown LDAP realm or unknown LDAP Directory parameters and preserve the remainder of the realm configuration. Prior to this update, an unknown parameter would cause the entire realm to be silently discarded i.e. the realm would no longer be present in the updated authfile.
        - Change to recognize and preserve the LDAP realm's "`Policy-Pointer`" parameter, if present. Prior this update, "`Policy-Pointer`" was treated as an unknown parameter.
        - Change to recognize and preserve the LDAP Directory's "`NumberOfBinds`" parameter, if present. Prior this update, "`NumberOfBinds`" was treated as an unknown parameter.

26. Changes to man pages:
    - Updated the man pages to reflect the 8.3 changes.

27. Utility program changes:
    - Changed the utility programs (radcheck, radpwtst, sesstab, etc.) so that the Interlink copyright lines are not included as part of the output of a normal run. The copyright lines are included in "-v" and "-h" output. The "-0" command-line parameter has been removed.
    - Expand the radcheck help display to include a description of each command-line parameter.

FIXES in 8.3.0:

1. Fixed an issue in the debug display of a received EAP-Message. If the complete received EAP-Message spans multiple EAP-Message attributes (such as when receiving a certificate), only the octets from the first EAP-Message attribute were displayed in radius.debug. Now the complete EAP-Message is displayed.

2. Fix rare buffer overflow bug in radpwtst, found during testing. radpwtst concatenates the values of all received Reply-Message attributes into a single buffer of size 4096 octets. If the received message contains more than 4096 bytes of Reply-Message data, the buffer can overflow resulting in a segmentation fault. The Reply-Message buffer is now increased from 4K to 32K bytes.

3. Change CHECK and DENY failures to both produce "Access not allowed" failures. Previously a DENY failure produced "Access not allowed" while a CHECK failure (incorrectly) produced "Authentication failure".

4. Fix an advanced policy problem where a decision file line is silently discarded if it contains an attribute whose value is very long e.g. "`insert <attr>=<value>`", where <value> is long. Now <value> will be allowed if <=4096 octets in length, and a logfile message is generated if the <value> is longer than 4096 octets.

5. Fixed an issue in the code that reads "`%event <name> <value>`" lines from the FSM. The code failed to check for overflow e.g. "%event DROP 99999999999999" was accepted and silently truncated to the value 2147483647 (=0x7FFFFFFF=the max signed 32-bit value). The code now detects the overflow, and generates an (E) logfile message and terminates the server.

6. Fixed an issue where the server incorrectly generates an error when parsing a named attribute value, from a check/deny/reply item list, if the VALUE's name begins with a decimal digit in the dictionary.

7. Fixed an issue where a dictionary integer VALUE starting with a plus sign (+) is treated as an error.

8. While adding support for the handling of received unknown (i.e. not in dictionary) attributes as recommended in RFC6929, fixed an issue in the pre8.2 server where it would sometimes double-encapsulate an unknown VSA when transmitting it.

9. Found an issue where, if the server is started with inetd/SMF and terminates due to a bind error ("Address already in use") for one of its configured ports, inetd will immediately try to start the server again upon receipt of the next message to the inetd port. This sequence would repeat over and over. Changed the server's behavior when it is started by inetd and experiences a bind error, the server will generate a logfile message, ignore the port with the bind error, and start up without that port.

10. Fix a problem where a HUP can cause the server to terminate with "`(E): sig_fatal: exit on signal (6)`", when trying to close an LDAP connection that is not fully up.

11. Fixed a problem found internally, which can core the server with a segmentation fault, when converting the value of an abinary attribute to text when the value is bad. The value is converted to text when:
    - displaying the value in the debug file when debug is running
    - logging the value in the accounting logfile if received in an Accounting-Request message
    - logging the value in the logfile via an advanced policy "LOG" command

12. Server Manager fixes:
    - Fixed a Server Manager problem where the "LocalRealms->New LDAP Directory" window would fail to pop up, and where the "Proxies->Add Realm" window would fail to pop up.
    - Fixed problem, when using the Firefox browser, where Firefox would sometimes open items in a new tab after the user opens the Server Properties items.
    - Fixed various Server Manager problems encountered when using the Internet Explorer 11 browser.

REMOVED FEATURES in 8.3.0:

1. Desupport the "`send_proxy_action`" **aaa.config** parameter. Remove dictionary definition and all internal support for the Merit `Proxy-Action` attribute.

2. Remove LDAP policy support, previously deprecated but still supported.

3. Removed configuration of "`Hold Accounting Requests`" and "`Hold Authorization Requests`" from the Server Manager. These parameters do not function as described, and were a diagnostic tool that should not be configurable.

KNOWN ISSUES in 8.3.0:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /`opt/aaa` by default.
2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent
    a. If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of `radiusd` does not help).
    b. If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
    c. The `iaaaAgent.conf` file in the config directory does not control the attempt to reconnect currently.

4. Server Manager issues:
    a. There is a problem with Java JRE 1.8.0_60 through 1.8.0_71 which causes Internet Explorer 11 to occasionally insert a delay of about 20 seconds when an applet is called. This issue was fixed with Java JRE 8u72. This does not affect the Firefox browser. This affects some Server Manager "Edit Configuration" screens.
    b. As of September 2015 the Chrome browser no longer supports Java plugins, and thus will not work with the Interlink Server Manager.
    c. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

5. TLS stateless session resumption fails if doing EAP-TLS or EAP-TTLS and using the Windows 8 or 8.1 supplicant. TLS stateless session resumption is, by default, disabled.

8.2.3 RELEASE NOTES

NEW FEATURES in 8.2.3:

CHANGES in 8.2.3:

1. Upgraded OpenSSL from version 0.9.8y to 0.9.8zc.

2. Add Broadband Forum VSAs (formerly ADSL) from RFC 4679.

3. Updated the 3GPP2 dictionary based on X.S0011-005-C_v3.0_061030.pdf.

4. Added logfile messages for every listen and proxy port that is opened.

5. Changed server logging so that, following a HUP, the server reports the "Server is listening ..." and "Server is proxying from..." logfile messages for all ports, whether their configuration has changed or not.

6. Added new '(E)' logfile message if, within a radius_socket{} block, an 'auth_udp_recv_buffer_size' is configured but no 'authport' is configured.

   Added new '(E)' logfile message if, within a radius_socket{} block, an 'acct_udp_recv_buffer_size' is configured but no 'acctport' is configured.

   These are not fatal errors, the server's action is to ignore the useless udp_recv_buffer_size parameter. Previously, in this situation, the udp_recv_buffer_size parameter was silently ignored.

7. Suppress the "(N) For socket(<i>), the requested UDP receive buffer size(<m>) differs from the actual UDP receive buffer size(<n>)" logfile message on Linux, when the actual buffer size is 2x the requested buffer size. The reason is that Linux systems double the requested buffer size value (within the kernel) when you set it [via setsockopt()], and returns the doubled value when you query it [via getsockopt()].

FIXES in 8.2.3:

1. Fixed an EAP-PEAP TLS Session Resumption issue when using the Windows 8 and 8.1 supplicants.

2. Fix two bugs in routine auth_vectortoa(), which converts a 16-octet RADIUS packet authenticator field into a printable string which could overflow its static buffer and trash memory on every 20th invocation.

3. Fixed an issue where, if started with inetd/SMF, the server ignored the udp_recv_buffer_size parameter configured in a radius_socket{} for a port which matches the inetd port.

4. Fixed a couple of logfile messages, where the wrong socket number or the wrong port type (authentication versus accounting) is reported for the inetd port (under unusual configurations).

5. Fixed an issue where a change is made after the server has been started that only affects a port's udp_recv_buffer_size, and then the server is subsequently HUPed, the new udp_recv_buffer_size was ignored and the port left as-is with the previous udp_recv_buffer_size.

6. Fixed an issue where a server starts as an AUTHENTICATION-ONLY server (i.e. with no accounting listen ports), is reconfigured as a non-AUTHENTICATION-ONLY server (i.e. an accounting port is configured), and then HUPed. The server will bind to the new port but will not process any received messages from the newly opened port's receive buffer.

7. Fixed the same issue where a server starts as an ACCOUNTING-ONLY server is reconfigured as a non-ACCOUNTING-ONLY server and then HUPed.

REMOVED FEATURES in 8.2.3

KNOWN ISSUES in 8.2.3:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:

a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

8.2.2 RELEASE NOTES

NEW FEATURES in 8.2.2:

CHANGES in 8.2.2:

1. Upgraded to OpenSSL from version 0.9.8l to 0.9.8y. This fixes an issue with accepting SHA256 and SHA512 message digests in certificates. This affects EAP-TLS/TTLS/PEAP, LDAP and SNMP. Neither the old nor the new OpenSSL version has the heartbleed bug.
2. Added Propel (vendor 14895) VSAs.
3. Added Starent (vendor 8164) VSAs.
4. Added definitions for additional WISPr VSAs.
5. Updated 3GPP dictionary.
6. Updated Altiga dictionary.

FIXES in 8.2.2:

1. Fixed an issue, which we have not seen happen in production, in routine avpair_vtoa(), where an internal buffer can overflow, trash memory and core the server or utility program.

NOTE: The bug could only happen if debug is enabled and there is a string/octets attribute with more than 800 octets of data.

REMOVED FEATURES in 8.2.2

KNOWN ISSUES in 8.2.2:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:

a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

8.2.1 RELEASE NOTES

NEW FEATURES in 8.2.1:

CHANGES in 8.2.1:

FIXES in 8.2.1:

1. Fix to RSA SecurID code to handle users configured in users file with "Authentication-Type=SecurID".

REMOVED FEATURES in 8.2.1

KNOWN ISSUES in 8.2.1:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

8.2.0 RELEASE NOTES

NEW FEATURES in 8.2.0:

1. Re-architected RSA SecurID Support including:
   a) Integration of the latest RSA 8.1.2 Authentication Libraries
   b) The RAD-Server subprocess for RSA SecurID authentication supports a configurable number of simultaneous authentication requests to the RSA Authentication Manager for higher performance. The default is 1024.
   c) New authentication statistics are collected and reported by the SecurID subprocess.
   d) The SecurID subprocess now logs messages to the RAD-Server's logfile and radius.debug file. There are a number of new logfile and debug messages generated by the subprocess.
   e) The SecurID subprocess is now configurable in aaa.config for the following new parameters
      1. Debug-Level (Default 0. Range 0-4)
      2. Log-Statistics-Interval (Default 0. Range 0 to 2147483647)
      3. Number-of-Authorization-Control-Blocks (Default 1024. Range 1-8192)
      4. RSA-Trace-Level (Default 0. Range 0-15)

2. Dictionary support for Vendor Specific Attributes has been greatly expanded.
   a) The dictionary now supports the flexible configuration of VSA formats, including VSAs with <type> of 1-4 bytes and <length> of 0-4 bytes. The <type> and <length> of a vendor is configurable in the vendors file, with two new optional "Type-Field-Size=n" and "Length-Field-Size=n" parameters. By default "Type-Field-Size=1" and "Length-Field-Size=1".
   b) An "%INCLUDE <filename>" directive has been added to the dictionary, so that the dictionary can be partitioned into multiple parts. This facilitates the isolation of custom modifications while allowing standard updates of other attributes.
   c) Greater flexibility in organizing and storing the dictionary files.
   d) Vendor specific dictionary files are provided for a number of vendors.
   e) A dictionary.custom file is provided for defining customer specific VSA definitions.

3. Added support for logging to both syslog and logfile concurrently.
   The RAD-Server supports these startup options:
   "-g logfile"
   "-g syslog"
   "-g stderr"
   "-g logfile+syslog" (either order)
   "-g stderr+syslog" (either order)

   This implementation is backwards-compatible with existing startup scripts which support "-g logfile", "-g syslog", and "-g stderr".

4. Updated Chargeable-User-Identity (CUI) Support
   a) A CUI AVP is generated only if
      1. The RAD-Server is acting as the home server.

2. A NUL CUI AVP is present in the Access-Request.
3. A non-NUL CUI AVP is not already present for the response.

b) Session tracking is no longer necessary to support CUI.
c) Added a new "CUI-Encryption-Secret = <cui_encryption_secret>" configuration parameter to aaa.config. The optional parameter allows a customer to configure the secret used for encryption of the server-generated CUI. If not configured, a default value is used. The CUI-Encryption-Secret can be up to 128 characters long, and is truncated if a value longer than that is configured.
d) A NUL CUI is proxied in a request, but a NUL CUI is never returned in an Access-Accept, whether the destination is of type=NAS or type=Proxy.
e) A configured CUI will be returned in the Access-Accept only if the NAS requested one (via a NUL CUI in the request).
f) Several logging improvements were made.

5. Definitions have been added to support RADIUS packet types 21-51.

CHANGES in 8.2.0:

1. Changed the Session Manager to save the NAS-IPv6-Address AVP, if present in the Access-Request, in the session entry in addition to saving the NAS-Identifier and NAS-IP-Address AVPs.

2. Updated the Java JRE used by the Server Manager to version 7u25, to take advantage of numerous security improvements.

3. Updated the Tomcat server used by the Server Manager to version 4.1.40, to take advantage of numerous security improvements including the configuration of only strong encryption methods.

4. TLS stateless session resumption is disabled by default.

FIXES in 8.2.0:

1. TACACS+ and BRIEF accounting was fixed to log correctly when multiple streams are configured.

2. Fixed parsing of the clients file to do DNS lookups of names consisting solely of hex digits without colon characters. These had been interpreted as invalid IPv6 addresses.

3. Fixed parsing of the clients file to support "type=" strings longer than 128 characters.

4. Fixed a minor bug with FORKREPLY plugins so that they are now compatible with EAP-PEAP and EAP-TTLS authentication.

5. Fixed bug which can core a RAD-Series server when doing PEAP or TTLS authentication, if the NAS sends an unsolicited Access-Request/EAP-Message when no Access-Challenge is pending, or if the NAS sends an Access-Request/EAP-Message where the EAP-Response's EAP-Identifier does not match the outstanding EAP-Request's EAP-Identifier. The core occurs if the illegitimate request arrives during a certain timing window. This bug was the cause of the server cores reported in [support #10214] and [support #10222].

6. Various fixes to handling VSA length errors and handling empty tagged string AVPs.

REMOVED FEATURES in 8.2.0

KNOWN ISSUES in 8.2.0:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

8.1.0 RELEASE NOTES

NEW FEATURES in 8.1.0:

1. Implemented the first phase of the next generation Session Manager including:
   a) Well defined session states, transitions, timeouts, and states applying to simultaneous session and licensing limits
   b) Greater configuration control for session management including
      "Session-Collision-Timeout"
      "Session-Dropped-Timeout"
      "Session-Finished-Timeout"
      "Session-MIA-Timeout"
      "Session-Pending-Timeout"
      "Session-Unconfirmed-Timeout"
      "Accounting-OnOff-Support"
      "Acct-Interim-Grace-Period"
      "Session-Table-Update-Interval"
      "Session-Table-Checkpoint-Interval"
      "Session-Checkpoint-File-Lifetime"
      "Minimum-Acct-Interim-Timeout"
      "Simultaneous-Use-States"
   c) Performance improvements
   d) Generic token pools
   e) Server-generated accounting records for the following cases, where no Acct-Stop message is received from the NAS:
      1) NAS sends an Accounting-On/Off
      2) A session expires while waiting for an Interim-Acct message in the MIA state
      3) A session expires in the COLLISION state
      4) A session expires in the UNCONFIRMED state
   f) Session-Id identification when there are multiple Class AVPs
   g) Quicker releasing of resources such as IP addresses, tokens, and session counts following failed authentications.
   h) Support for a new Interim-Accounting timeout algorithm, based upon measuring the intervals between Interim Accounting messages.
   i) Improved matching of Accounting-Requests to sessions through use of the internally generated Session-Id.
   j) Improved logging of error conditions such as the session table full.

2. Added support for Chargeable-User-Identity. If the server receives a <NUL> CUI AVP in the Access-Request, it will generate a CUI and return the CUI in the Access-Accept, save the value, and change the value approximately every week. If the server is doing session management, then the CUI will be added to the accounting record that the server logs. This is the mechanism (i.e. searching the accounting logs) by which the customer maps a CUI to the real user identity.

3. Added a logfile (I) message when a radcheck request is received. This marker message helps correlate the radcheck statistics with the logfile.

4. Added support for the RETRIEVE_DEFAULT event. A search of a users file returns a RETRIEVE_DEFAULT, rather than a RETRIEVE_ERROR, if the default user is retrieved.

5. Added support for Delegated-IPv6-Prefix, from RFC4818.

6. Added a new "-writesess" parameter to radcheck. If specified, radcheck will request the server to write out session.las when processing the radcheck request.

7. Expanded statistics reported by radcheck.

8. Added new aatv.Tunneling{} configuration parameters:
   a) Tagged-VSA-Hints Accept | Discard | Reject
   b) Tunnel-Password-Requires-Message-Authenticator YES | NO
   c) HINTS Accept | Discard

CHANGES in 8.1.0:

1. Added NOLOG to these Interlink attributes, to suppress their appearance in accounting logfiles: Date-Time, Time-Of-Day, Day-Of-Week, Interlink-Packet-Code, and Interlink-Proxy-Action. This more accurately reports what was actually received in the Accounting-Request. NOLOG can be removed from any of these attributes in the dictionary for customers wanting them logged.

2. Changed las.conf processing so that End-Realm is optional in realm configurations.

3. Expanded support for a User-Name or User-Id longer than 64 characters. Now up to 253 characters (max length of a RADIUS string attribute) is supported. The server, session table, accounting, clients, sesstab, radrecord, and LDAP searches all support the longer userids.

4. The proxy server was changed to ignore NO_APPEND when handling an Acct-Response, and to retain the original acct authreq's AVP list, to which the proxy appends any new AVPs from the response following the proxy's Proxy-State AVP.

5. Improved parsing of command-line -p/-q/-pp/-qq parameters to handle out of range values.

6. Changed sesstab to display both the outer and inner identities for tunneled sessions e.g. PEAP/MSCHAP. Previously sesstab displayed only the inner identity.

7. The value of the Proxy-State AVP is changed to guarantee its uniqueness. The Proxy-State value is "<fsm-state>-<counter>-<authreq-pointer>", where <counter> is a 32-bit counter which increments each time a Proxy-State AVP is

generated. The <counter> is initialized to a 32-bit random number at startup. The <authreq-pointer> is the address of the authreq. The <authreq-pointer> component is never dereferenced.

8. Changed all aaa.config keywords to be consistently case insensitive.

9. The server no longer runs the Session-Timeout, which caused potential conflicts with the NAS, which has responsibility for Session-Timeout.

10. Improved range checking of las.conf configuration parameters. Extended log.config and las.conf parsing error handling.

11. Changed the server to reject an Access-Request for a realm which isn't being session-tracked and if session-managed resources (tokens and/or IP addresses) are being requested.

12. Changed the tunneling support so that the aatv.Tunneling{} block is re-read upon a HUP signal. Previously the aatv.Tunneling{} block was only read at startup, and ignored thereafter.

13. The new session checkpoint interval takes place immediately after the HUP, rather than waiting for one expiration.

14. The Server Manager was updated to configure parameters which previously could only be added by directly editing the configuration files.

FIXES in 8.1.0:

1. The RADIUS server was fixed to fully support 32 bit values in the NAS-Port AVP. Formerly, this support was limited by special values internally indicating conditions such as the absence of the AVP.

2. Fixed the parsing of integer las.conf parameters to check for underflow and overflow conditions.

3. log.config parsing was fixed to correctly handle trailing spaces and parameters in quotes.

4. Fixed the handling of configured tagged AVPs, so that all configured tagged AVPs, not just a fixed set of RADIUS tagged AVPs, take part in the merging-of-tunneled-hints algorithm.

5. Fix for the cases during logfile rollover where some messages related to logfile compression were not being logged.

6. Changes were made to correct the output of misleading logfile messages which can appear when a response is received for which there is no matching proxied request. The server now logs "Received unexpected response whose id matches

no pending request", and no longer says either "Received response ... with bad authenticator" or "Received response for completed request".

7. Fixed bug which occurs when debug is on and a request is received with a zero-length password.

REMOVED FEATURES in 8.1.0

1. Removed support for obsolete Modem-Start, Modem-Stop, and Cancel Acct-Status-Types.

2. Removed configuration of obsolete and unsupported LAS services.

3. Removed support for the ACCT_DUP event.

KNOWN ISSUES in 8.1.0:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The 32 bit X window compatibility libraries must be present for the installer to run. On systems where they are missing, the installer gives a cryptic message about missing the JVM. The real problem is that the OS is missing the X window compatibility libraries. On Linux OSs the libraries can be installed with the equivalent of

```
# yum install libXp.i686
# yum install libXt.i686
# yum install libXtst.i686
```

5. The Server Manager may experience an out-of-memory exception if Maintenance->Statistics or Maintenance->Logfile has to parse a logfile with too many records in the given selection time span. The workaround is to select shorter time spans, say two 12-hour periods rather than one 24-hour period, to accumulate the desired information.

8.0.2 RELEASE NOTES

NEW FEATURES in 8.0.2:

1. Added support generic salt encryption of attributes. This allows for attributes to be identified as salt encrypted in the dictionary to support salt encrypted vendor specific attributes.

CHANGES in 8.0.2:

1. Changed server handling of an error condition: Now the server, when asked to create a tagged-int attribute with tag > 31 or with the value > 24 bits, will discard the attribute.

FIXES in 8.0.2:

1. Fix problem where a miss-configured abinary attribute value, with a keyword exceeding 80 characters, can overflow an internal buffer, possible causing the server to core.

2. Fixed an issue where the server could hang waiting for input on one of its ports following a HUP.

REMOVED FEATURES in 8.0.2

KNOWN ISSUES in 8.0.2:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:

a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

8.0.1 RELEASE NOTES

NEW FEATURES in 8.0.1:

1. Added support for a new "compress-logfile on|off|yes|no" aaa.config parameter. Default = on = yes, the server's current behavior.
   If set to OFF, the server will not compress the logfile upon logfile rollover, leaving it as logfile.yyyymmdd rather than compressing it to logfile.yyyymmdd.gz.

2. Added support for authport=zero and acctport=zero in the radius_socket{} block. Prior to this update, the exact non-zero port needed to be explicitly configured.
   Now, if authport is configured as zero, the server will execute a hierarchy of steps to determine the authentication listen port:
   [Step1] If "-p authport" is configured, use that value, else
   [Step2] If the environment variable RAD_AUTH_PORT is defined, use that, else
   [Step3] If getservbyname() returns a port, use that, else
   [Step4] Use 1812, as the default defined by the RADIUS RFC 2865.

   Now, if acctport is configured as zero, the server will execute a hierarchy of steps to determine the accounting listen port:
   [Step1] If "-q acctport" is configured, use that value, else
   [Step2] If the environment variable RAD_ACCT_PORT is defined, use that, else
   [Step3] If getservbyname() returns a port, use that, else
   [Step4] Use 1813, as the default defined by the RADIUS RFC 2866.

CHANGES in 8.0.1:

1. Implement fix so that an Accounting-On/Off message from a NAS will cause that NAS's sessions to be cleared, previously the server received and ACKed an Accounting-On/Off message, but did not clear any sessions. Also now we allow the Acct-Session-Id AVP to be absent in an Accounting-On/Off. [12317]

2. Change the processing of the clients file to allow and ignore comments (starting with '#') at the end of a line.

3. Added a '(N):' type logfile message when we disallow PEAP fast reconnect and force a new full authentication. Note: only the message is new.

4. A new logfile message now appears if there are no certificates configured in the aatv.ProLDAP{} block. It appears once each time the authfile is read and there is a ldaps:// URL. The message now reads:
   "(N): No certificates are configured in 'aatv.ProLDAP{}' for use by ldaps:// connections"
   Previously a '(E):' message occurred for each ldaps://... URL configured in the authfile

5. Changed the ipv6check.sh script to handle a different DNS response for ns0.ietf.org. Also improved the checking of returned IPv6 addresses.

6. Changed the server so that when it receives a response to a proxied request to accept, rather than discard, an empty (no AVPs) Accounting-Response even though it does not contain the RFC required Proxy-State attribute. Additional changes were made to correctly distinguish an empty response from a malformed response.

7. Changed the maximum number of states in the FSM from 256 to 1024. This change requires a new SDK to handle the larger number of states.

8. The server now logs "Received unexpected response whose id matches no pending request" instead of "Received response ... with bad authenticator" and "Received response for completed request".

9. Added protection for when radiusd acts as a proxy server and receives a response with an AVP that needs to be decrypted and re-encrypted and the AVP is corrupted. This is for these AVPs: MS-CHAP-MPPE-Keys, MS-MPPE-Recv-Key, MS-MPPE-Send-Key, Tunnel-Password and Cisco-Avpair("leap:session-key=xxx").

10. Changed the server acting as a proxy to ignore NO_APPEND when handling an Accounting-Response, and to retain the original acct authreq's AVP list, to which the proxy appends any new AVPs from the response following the proxy's Proxy-State. That is, the proxy server treats an Acct-Response as APPEND, whether the home server was configured as APPEND or NO_APPEND.

11. Added a new global variable, "int radiusd_pid", which holds the process-id of the radiusd main process. This is available to child processes who may want to check if their parent is still alive, i.e. the child process can periodically compare their current parent [as returned by getppid()] against radiusd's PID.

12. Changed DNS update child process to check, after every DNS lookup, if his parent process is still alive. If not, the DNS update process terminates.

13. Improved logfile messages regarding starting and ending of child processes:
    - When a DNS-Update or logfile-compression child is forked, a logfile message is generated, indicating that the child has been launched and indicating the child's process-id, e.g.:

      (I): Have forked DNS Update child process with pid(4589). [update_clients]

    - When the server receives a SIGCHLD signal for a child process, a logfile message indicating the PID of the terminating child is displayed, e.g.:

      (I): DNS update (pid:4589) finished. [child_end]
           or, if not a DNS update child:
      (I): Received SIGCHLD signal for child process (pid:4589). [child_end]

    - Changed the loglevel for a logfile message for a child process that

terminated abnormally, from (I) to (A).

FIXES in 8.0.1:

1. Fixed an issue in three dprintf() statements which can core on Solaris. This bug occurs only when debug is on and only when running with a non-default FSM where "%enable_ingress_egress_policy" is NOT set to "yes", and only when proxying requests.

2. Fixed an issue in a dprintf() statement which can core on Solaris. This bug occurs only when debug is on and a request is received with a zero-length password.

3. Fix bug in log_init() when processing log.config's "default-path" parameter and encountering an error, where the server can try to display a NULL string pointer.

4. Fixed the launching of a child process such as SecurID or Oracle, to close any inherited RADIUS listen or RADIUS proxy sockets. If the server was launched by (x)inetd, then this includes socket #0 opened by (x)inetd. Failure to close these sockets causes a problem when HUPing a server which has the SecurID sub-process running and is then not able to bind to its RADIUS listen ports.

5. Fixed an issue in the parsing of the "Session-Collision-Checking on/off" parameter value in las.conf.

6. Fixed an issue that can cause a core when running a very specific custom FSM. This bug allowed multiple entries in the proxied request index for the same authreq.

7. Correct the display of a Tagged string attribute whose tag value is > 31.

8. Fixed a segmentation-fault core which occurs when debug level is > 2 and when processing log.config and finding that the configured accounting log AATV doesn't exist.

9. Fixed the sesstab utility to correctly display a session in INIT state (value -2) rather than as a session in state 254 (unknown).

10. Fixed parsing of "type=xxx" field of clients file, so an invalid (i.e. unknown) flag, e.g. type=NAS+BLAH, is be detected and logged.

REMOVED FEATURES in 8.0.1

KNOWN ISSUES in 8.0.1:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

8.0.0 RELEASE NOTES

NEW FEATURES in 8.0.0:

1. Support for IPv6 addressing and communications has been added to the server. See the Admin Guide for more details.
   a) Support for the new IPv6 attribute types: ipv6addr, ipv6prefix and interfaceid have been added [12318].
   b) Support for the IPv6 socket and IPv6 DNS lookups has been added [12319].
   c) Support for the IPv6 connections in the utilities has been added [12319].
   d) IPv6 support for communications to a LDAP server has been added [12319].
   e) An IPv6 readiness tool has been added to the installation.

2. The SNMP RADIUS MIB support has been updated to the extended MIBs per RFCs 4668/4669/4770/4771, replacing support for the deprecated MIBs per RFCs 2619/2621. This includes new support for the RADIUS Client MIB [12319].

3. Significant performance improvements have been made in these areas:
   a) Duplicate request checking,
   b) Dictionary lookups of attributes and values,
   c) Lookup of users from the users file,
   d) Management of proxied requests,
   e) Management of work queue,
   f) Lookup of a client by name or IP address from the clients file, and
   g) Management of session table.

4. You can now configure how the date attributes in the accounting log files are displayed. In the log.config file, you can now define the strftime-format string used to display the dates. The implementation is fully backwards-compatible: if the new fields are not configured, then the server behaves as before. See the Administrator's Guide for details.

5. Added the ability to optionally configure HH:MM:SS when configuring check/deny/reply attributes and the ability to configure dates in either YYYY-MM-DD format or MMM-DD-YYYY format.
   Also added the ability to optional configure the time zone of "UTC" to override the default of local time for date check/deny/reply attribute values [12318]. See the Administrator's Guide for details.

6. Added support for the "<ipaddr>/nn" wild carded address format for both IPv4 and IPv6 addresses in the clients file. The IPv4 '*' syntax e.g. '192.168.*' continues to be supported. Added a check for duplicate wild carded clients, e.g. 192.168.* and 192.168.*.* and 192.168.0.0/16 are equivalent [12319].

7. Added support for the optional "srcip=<ipaddr>" parameter in the clients file. It can also have an optional source port, "srcip=[ipaddr]:port. Note the square brackets are mandatory if a port is entered [12319].

8. Added a global proxy_udp_recv_buffer_size parameter in aaa.config. Added auth_udp_recv_buffer_size and acct_udp_recv_buffer_size parameters that can be optionally configured within the new aaa.config radius_socket{} listen block.

   a) The range of valid values for the UDP recv buffer size parameters is 8192-to-8388608 (8KB-8MB). If the xxx_udp_recv_buffer_size is not configured, the socket is opened but the socket is left at whatever buffer size the system uses by default.

   b) When a socket is opened: if there is a configured xxx_udp_recv_buffer_size and that buffer size does not match the actual buffer size assigned by the OS, then a logfile (N) message is generated.

9. Added support for two new global aaa.config parameters: "default_source_ipv4_address" and "default_source_ipv6_address" [12319].

   a) The default values, if not configured, are:
      default_source_ipv4_address 0.0.0.0
      default_source_ipv6_address ::

   b) These parameters specify the default source IP address to use when proxying a RADIUS request, if the client record did not specify a "srcip=<ipaddr>" field.

   c) The default values indicate the ANY address, which lets the system pick the source IP address.

10. The server can now listen for requests on multiple authentication sockets and on multiple accounting sockets, and can proxy from multiple sockets. Each such socket is configured with an IP address (IPv4 or IPv6) and a port.

CHANGES in 8.0.0:

1. Two new messages were added for an aaa.config {}block which is configured but not processed. During the initial server startup, the message is:

   (N): config_cleanup: Block 'xxx{}' was configured but not processed.

   Following a HUP, the message is:

   (N): config_cleanup: Block 'xxx{}' was configured but not reprocessed following the HUP.

2. Added support for optional [ ] around IPv4 addresses in clients file.

3. Improved the error detection of clients file records.

4. Changes to the handling of the aatv.ProLDAP{} configuration:
   a) Continue processing the remainder of the config block,
      upon encountering a bad parameter or bad value, rather than
      bailing out and ignoring the remainder of the block.
   b) Added a new logfile (N) message upon encountering a
      TLS-xxx parameter with an empty string e.g. TLS-CertFile "".
   c) When processing the authfile for a PROLDAP auth type, generate a new
      logfile (E) message upon encountering a "URL ldaps:..." line, and
      finding that there are no certificates configured in the
      aatv.ProLDAP{} block.
   d) Added a new (E) logfile message if an "Enable-Default-Conf" parameter
      follows a "TLS-xxx" parameter, as processing of the TLS-xxx parameter
      requires the Enable-Default-Conf value to be first.

5. Changed the time the server waits before removing a session
   in the LAS-Stop state from the session table, from Session-Clear-Time
   (default: 30.25 minutes) to Session-Hold-Time (default: 45 seconds).

6. Changed the minimum acceptable time between HUPs from 1 second to
   2 seconds. Now a new HUP will follows the end of the previous HUP by fewer
   than two seconds will be ignored, and a logfile message generated.

7. Changed the decision file processing so that, when displaying a date
   attribute value, it will always display the date and time.

8. Changed all printing of a date attribute, outside of accounting logging,
   to print the full-precision date and time.

9. Upgraded to Net-SNMP version 5.4.3.

10. Added the NAS-Restart enumerated value for the LAS-Code attribute
    to the dictionary.

11. When parsing configured check/deny/reply items and encountering an invalid
    attribute value, the handling of invalid configured attribute values
    has changed to:

    a) A logfile error message is generated as before, identifying the
       invalid attribute and value.

    b) A logfile error message "(E) <function>: Parse error for user
       '<userid>' in file '<filename>' at line <#> (check/deny items)", or
       "...(reply items)", is generated.

    c) The user record is discarded. The authentication will fail.

    Note -- This also affects more than just configured check/deny/reply
       items. For example the initial processing of radius.fsm processes the
       xvalue and xstring optionally appended to the end of a FSM entry.
       If there is an error in <xvalue> or <xstring>, this error was
       previously ignored and the <xvalue> or <xstring> were discarded.

Now the server will report the error and fail to start due to a FSM syntax error.

12. Improved the logging of attribute validation failure messages. Every validation failure message is now a Warning-level (W) message, formerly most were Error-level (E). Every validation failure message now identifies the failed attribute by name, displays the value, and indicates that the server's action is to discard the offending attribute [12320,12323].

13. Changed loglevel of some "xxx exceeds range" messages from loglevel (A) to loglevel (E).

14. Improved logging of bad configured attribute values, to identify the specific badly-configured attribute:

    a) You now get an (E) message if the attribute name is not in the dictionary.

    b) You now get an (E) message if the attribute value is incompletely specified, e.g. "NAS-IP-Address=" or just "NAS-IP-Address". Previously such an incomplete specification would just cause the check/deny/reply item to be silently discarded.

    c) You now get an (E) message if you specify "!=" for a Reply-Item.

    d) In all cases of a bad attribute, you should now see a two line pair of error messages. The first line of the pair identifies the bad attribute and the second line identifies the userid.

15. For received attributes of type tagged-integer, the server now checks that the tag is in the range 0-31. If not, a logfile message is generated and the attribute is discarded. Also check that the data length of a tagged-integer attribute is exactly 6. If not, a logfile message is generated and the received attribute is discarded.

16. Allow optional square brackets around any IPv4 addresses configured as a proxy hosts in the authfile [12319].

17. Improved the logfile message which appears when the server receives an unexpected proxied response, i.e. the response is for an already-completed proxied request, or the response for a pending request has a bad authenticator, or the id of the response matches the id of no pending request.

18. Changed log level of "HUP signal received (HUP#<n>)" logfile message from (I) to (N).

19. Changed the loglevel of the logfile message which is generated, when a proxied request cannot be sent because the proxied host name is not yet DNS-resolved, from (E) to (N). Also reworded (clarified) the

message, which now says: "(N): radius_send: Proxying to server '<name>' failed, server name not yet resolved to an IP address.". The server does not respond to the NAS, the server awaits a later NAS retransmission whereupon the DNS resolution has likely been completed and the REDO action will cause the proxied request to be sent.

20. Changed the logfile message for a proxied request which cannot be sent when the proxied host (from the authfile) does not exist in the clients file. It now says "(E): radius_send: Proxying to server '<name>' failed, server not in 'clients' configuration file."

21. Changed the logfile message which appears when a request is received from a NAS and which is missing a required attribute, to additionally display the User-Name and NAS-Port.

22. When retransmitting a response, the server now logs an (N)-level message indicating that a response is being retransmitted. The log message identifies the message type and remote client.

23. Changes made to radpwtst are:
    a) Made the "-x" behave the same as "-X" behaves.
    b) Added a new switch and value, -ipv6 on/off, to allow it to send IPv6.
    c) Added a new switch and value, -secret secret, to allow communication with a server which is not in the clients file.
    d) Changed radpwtst to display the list of sent and received attributes if debug is on ("-x" or "-X").
    e) Improved error messages about parameter problems.
    f) Removed the "Password=7" and "Status-Server=12" help text and replaced it with "n = Packet code n, 0 <= n <= 255" under the "-c <code>".
    g) Revise all of the help text, describing all the parameters.

24. De-supported the "-:" and "-P" command line parameters in radcheck. Added "-0" to the help text. Reformatted the help text.

25. Changed the configurable maximum size of Max. Authentication Requests (global_auth_q.limit) and Max. Accounting Requests (global_acct_q.limit) from 65535 to 100000.

FIXES in 8.0.0:

1. Fixed an issue with printing the time zone of any "date" attribute on Solaris [12317].

2. Fixed an issue in the Merit style accounting logfile where any date attribute was incorrectly displayed as "yyyy-mm-dd/mm/yy", e.g. "2010-01-01/20/10" [12317].

3. Improved the error checking when processing configured "ipaddr" type attributes such as configured check/deny/reply attributes to make sure

they are in standard dotted-quad format [12322].

4. Fixed radpwtst to handle the "-i <clientid>" parameter correctly. Changed it to send either NAS-IP-Address or NAS-IPv6-Address in an authentication request, whichever is appropriate [12319].

5. Added a length check for received 32-bit attributes of type integer, date, and ipaddr [12320,12323].

6. Removed a misleading logfile message that occurred if the PROLDAP configuration has 'retrieve-only false' and the user password check fails. An extraneous Error-Level (E) message was written to the logfile [12317]. An example message is:

   ```
   (E): Providing ERROR event to FSM: Authentication: 168/152
     'fred@realm.com' from t25.abc.com port 2501
   ```

7. Fixed an issue where the year 2038 was treated as a valid year when converting a text date to an attribute.

8. Fixed an issue which allowed time values earlier than 1970 on some systems.

9. Fixed an issue where some configured local time zone Dec-31-1969 date/times were treated as invalid even when these local time zone date/times translated to a UTC date/time which was on or after Jan-1-1970 00:00:00 UTC.

10. Fixed an issue where ".1.2.3" was treated as a name rather than an invalid IPv4 address.

11. Fixed an issue where the final attribute of a received message is accepted if it "only" runs off the end of the packet by one or two bytes. Clarified the logfile error message when we run off the end of a received message.

12. Added a check for a received attribute value length of zero. The attribute is now discarded and a logfile (W) message is produced if the attribute is of type integer, ipaddr, date, octet, short, tag_str or tag_int. The attribute is accepted and a logfile (N) message is produced if the attribute is of type string, octets, or filter_binary.

13. Fixed a possible buffer overflow problem when displaying IP addresses.

14. Fixed an issue with replies sent by FREPLY AATVs being discarded by the server.

15. Fixed the case where a proxied request which could be sent returns a NAK and the ingress/egress policy is enabled; previously the server did not add an Interlink-Reply-Status attribute to the authreq, as required by the replyDispatch AATV.

16. Fixed radpwtst output, when attempting to output a logfile error message.

17. Fixed an issue where the radiusAuthServTotalPacketsDropped counter is sometimes incorrectly incremented if the server is doing PEAP tunneled authentication.

18. Fixed an issue where the server still counts all the existing sessions even though the las.conf is configured with Simultaneous-Use=-1. This does no harm other than the wasted overhead of counting sessions.

19. Fixed the aaa.config parameter avpair_checking, which was ignored in some cases. This had some performance implications.

20. Fixed an issue which can crash server when processing a Filter-Id attribute of length > 16 characters, and for a realm for which we do session-tracking.

REMOVED FEATURES in 8.0.0

1. De-supported the configuration of the "ourhostname" parameter in the aaa.config file. Use new aaa.config block radius_socket{} instead.

2. Remove support for paired <name1>/<name2> in clients file.

3. Remove support for obsolete US Robotics extensions.

4. Removed the support for the "Proxy Forwarding" message. The Proxy-Forwarding uses RADIUS message code of 216 and is Merit-specific, not a standard RADIUS message type.

5. The following switches, and all code conditionally-compiled under these switches, have been removed:

   MERIT_HUNTGROUP, MERIT_HUNTGROUP_DAC, MERIT_HUNTGROUP_SHP, Y2K, MERIT_TIMELEFT, MERIT_NASMAN, MERIT_HGAS, MERIT_OAS, MERIT_ORGANIZATION

6. Removed the support for the "year_2000 on/off" configuration parameter, which controlled whether the year was displayed as 4-digits or 2-digits. The rad_time() routine now always displays the year as 4-digits.

7. Removed -C (token caching) and -P (password changing) support.

KNOWN ISSUES in 8.0.0:

1. When installing the server over previous installations, the java runtime files are not correctly updated. The workaround is to rename the java directory before starting the install process. The java directory is in the binary directory, /opt/aaa by default.

2. When installing the server over previous configurations and you have hand edited your old configurations for an authentication plug-in, it is possible that your plug-in line from the configuration may be lost during the conversion of the authfile. Re-adding the line after the installation/conversion will be necessary.

3. There are some issues interfacing to the SNMP master agent [12307]:
   a) If the master agent is not running when radiusd starts it will never connect to master agent even after starting master agent (a HUP of radiusd does not help).
   b) If the connection to the master agent comes up but the master agent is stopped and restarted, the connection to radiusd is not reported as down and does not regain functionality (a HUP does not help).
   c) The iaaaAgent.conf file in the config directory does not control the attempt to reconnect currently.

4. The User-Name AVP in an Accounting-On request is not required and has no particular meaning since the Accounting-On applies to the entire NAS and not to a particular session or user. Currently the Accounting-On processing only clears sessions from the session table if the specified realm is configured in las.conf. The realm comes from the User-Name or is NULL realm if there is no User-Name AVP. It is possible that an administrator wants to do session tracking for some realms but not the NULL realm which would prevent the Accounting-On from doing the session clears that are needed [12303].