Part No. NN47250-700
November 2008

4655 Great America Parkway
Santa Clara, CA 95054

Nortel WLAN Security Switch 2300 Series Troubleshooting and Debug Guide

NORTEL

Copyright © 2008 Nortel Networks. All rights reserved.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Nortel.

The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of that license. The software license agreement is included in this document.

Trademarks

*Nortel, Nortel Networks, the Nortel logo, and the Globemark are trademarks of Nortel Networks.

All other trademarks and registered trademarks are the property of their respective owners.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Nortel Networks reserves the right to make changes to the products described in this document without notice.

Nortel Networks does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

Content

How to get help 5

General troubleshooting tips 7

Rules of troubleshooting 7

Useful CLI commands 7

WLAN 2300 security switch software troubleshooting scenario quick reference sheet 9

WSS software debug command descriptions 14

WMS troubleshooting areas 18

Troubleshooting scenarios 20

Client unable to connect to wireless network 20

Switch stability 21

WMS service database corruption 22

Troubleshooting auto-tune channel 23

Troubleshooting auto-tune power 24

Data Rate Enforcement 25

Mobility-Domain troubleshooting (seed and secondary-seed) 27

RF Analysis 27

RF Visualization 27

Voice Monitoring 28

RfLink 28

Scheduled Reports and E-mail 28

Unethered mesh AP unable to connect to portal AP 28

To verify the session is local-switched 29

Local switching enabled and the AP cannot boot 30

Session is disconnected after roaming 30

WLAN Location Engine 2340 troubleshooting areas 31

System availability 31

Administrative Web User Interface 33

Web User Interface 33

Sensor Connection and Communication 34

Tracking 35

Dashboard 36

Common Troubleshooting Techniques for WLAN Location Engine 2340 38
Remote Access to the WLE2340 Command Line Interface 38
The Dashboard Logs 39
Debug trace walkthroughs 40
Dot1x level 10 trace of WPA/TKIP with local PEAP-MSCHAPv2 40
Dot1x level 10 trace of dynamic WEP in pass-thru: 46
RADIUS level 10 trace of 802.1X pass-thru authentication 51
SM level 10 trace of client connecting 56
SM level 10 trace of client tear-down (idle disconnect) 64
Emergency Recovery Tree 71

How to get help

This section explains how to get help for Nortel products and services.

Getting Help from the Nortel Web site

The best way to get technical support for Nortel products is from the Nortel Technical Support Web site:

http://www.nortel.com/support

This site provides quick access to software, documentation, bulletins, and tools to address issues with Nortel products. More specifically, the site enables you to:

- download software, documentation, and product bulletins
- search the Technical Support Web site and the Nortel Knowledge Base for answers to technical issues
- sign up for automatic notification of new software and documentation for Nortel equipment
- open and manage technical support cases

Getting Help over the phone from a Nortel Solutions Center

If you need additional information to that available on the Nortel Technical Support Web site, and you have a Nortel support contract, you can get help over the phone from the Nortel Solutions Center.

In North America, call 1-800-4NORTEL (1-800-466-7835).

Outside North America, go to the following Web site to obtain the phone number for your region:

http://www.nortel.com/callus

Getting Help from a specialist by using an Express Routing Code

To access some Nortel Enterprise Technical Support Centers, you can use an Express Routing Code (ERC) to quickly route your call to a specialist in your Nortel product or service. To locate the ERC for your product or service, go to:

http://www.nortel.com/erc

Getting Help through a Nortel distributor or reseller

If you purchased a service contract for your Nortel product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller.

General troubleshooting tips

Rules of troubleshooting

1. **Read the Release Notes!**
   Release Notes contain important feature functionality information but they also include the "Best Practices" section which has recommendations for everything from Web-base AAA to 802.1X clients, as well as documentation on existing issues and known behaviors.
2. **Try to reproduce the problem.**
   If you can create a recipe for recreating the problem it will be fixed quickly. Try to write down what you were doing prior to the problem and then retracing your steps to make it happen again. *Reproducing the issue is vital to quick turnaround of fixes.*
3. **Try to isolate the location of the problem**.
   Obvious demarcation points are from the WSS to the network, and from the AP to the client. If this is a packet-related problem you can place sniffers at these points to verify that the packets are entering and leaving the Nortel gear correctly. This depends on being able to reproduce the problem of course. If possible, you should also load Ethereal onto the client and/or servers involved in the problem area (i.e. the DHCP server if client's aren't getting IP addresses). In 4.0 the remote sniffing capability will help greatly with client problems on DAPs.
4. **Characterize the problem as specifically as possible and look for patterns.**
   Is the problem occurring on all clients, or just some? Do the problem clients have any OS, NIC, or software similarities? Does the problem occur in the same portion of the building? Does the problem occur only on APs tuned to specific channels? How frequently does the problem occur? Is there any regularity to the frequency of occurrence?

Useful CLI commands

You can disable the CLI's press any key to continue prompt by setting the screen length to "0". You can re-enable it by setting it back to 24. This makes a text log from terminal programs MUCH easier to read.

set length 0

To transfer the tracefile's contents off of the switch via TFTP:

save trace tftp://10.1.1.107/trace.txt

Save a backup of the current configuration

save config <filename>

Copying just the configuration via tftp:

copy configuration tftp://10.1.1.107/config.txt

Transfers CLI configuration from WSS to tftp server.

copy tftp://10.1.1.107/config.txt config.txt

Transfers WSS CLI config from tftp server to WSS.

load configuration config.txt

You can restrict the number of entries displayed in the system and trace logs with some additional arguments:

show log buffer -10

(show last 10 entries from system log, newest entry first)

show log trace -10

(show last 10 entries from trace log, newest entry first)

show log trace /10

(show last 10 entries from trace log, newest entry last)

show log buffer matching AUTHORIZATION

(show all entries from system log containing the string AUTHORIZATION)

show log trace matching DOT1X

(show all entries from trace log containing the string DOT1X)

WLAN 2300 security switch software troubleshooting scenario quick reference sheet

Note. If "set trace …" commands are grouped together when listed, you should set them all at the same time. If they have a line between them they should be run separately.

Scenario	Sub-Scenario	Troubleshooting reference
User Authentication & Authorization Issues	General	Show sessions Show sessions network verbose Show sessions network session-id # Show dot1x clients Set trace sm level 7 mac-addr <mac-addr> Set trace dot1x level 5 mac-addr <mac-addr>
	802.1X/WPA	Show dot1x stats Show dot1x clients Show dot1x config Set trace dot1x level 8 mac-addr <mac-addr>
	Web Portal	Show crypto certificate web Set trace sm level 7 mac-addr <mac-addr> Set trace web level 10 Set trace dns level 10 Set trace httpd level 10

	RADIUS	Show aaa Ping <ip> Traceroute <ip> Set trace radius level 5 Check RADIUS server's logs
	Authorization Failures (Identity based Networking)	Set log buffer severity warning Set trace authorization level 10
Client Connectivity Issues	Intermittent Disconnects	Check Client driver version and settings and supplicant version. Set trace sm level 7 mac-addr <mac-addr> Set trace dot1x level 8 mac-addr <mac-addr>
	Sticky client	N/A – Check Client driver version and settings, supplicant, and over-the air sniffer tracing – Roaming decisions are made by the clients and the AP has no input into this decision. Verify coverage via site-survey
	Frequent roaming	N/A – Check Client driver version and settings, supplicant, and over-the air sniffer tracing – Roaming decisions are made by the clients and the AP has no input into this decision. Verify coverage via site-survey

	No DHCP for clients	Verify encryption/auth settings on clients. If static WEP double-check key Show sessions Show sessions network session-id Run Ethereal on the client to verify packets from the network. Run Ethereal on the DHCP server to verify receipt of packets from client Use “snoop” feature to verify DHCP packets entering/leaving the AP.
AP/DAP Issues	General	Show dap status terse Show dap status Show dap unconfigured Show dap counters Show dap etherstats Set trace dap
	DAP booting problems	Ethernet sniff as close to the DAP as possible Verify spanning tree disabled on port DAP is connected to Verify DAP has DHCP reservation Check DHCP server logs
	TAPA Tunnel	Set trace tapa

	Auto-RF	Set trace autorf level 10 Show auto-tune neighbors Show auto-tune attributes Set log buffer severity notice
	RF-Detect	Set trace rf_master level 10 Set trace rf_slave level 10 Set trace rf_client level 10 Show rfdetect counters
	Active-Scan	Upgrade to REL 4.0.20 or newer. Over-the-air tracing Disable Active-scan to see if the problem follows the state.
L2/L3 Issues	General	Show fdb Show arp Show ip route Show
	Access control list (ACL)	Show security acl info all Show security acl map <acl-name> Show security acl resource-usage Show security acl hits
	QOS Queuing on AP	show dap qos-stats
	WSS-to-WSS tunneling	Show roaming vlan Show tunnel Show vlan Show fdb

	Mobility Domain connectivity	Show mobility-domain status Ping <other WSSs> Traceroute <other WSSs>
WMS to WSS communications		Show ip https Show crypto certificate admin Set trace httpd Check to be sure enable password matches
NOS Stability	Memory Leaks	Show memory sum ? This should be run at regular intervals, and then rapidly ahead of an anticipated crash. For example, if the WSS crashes every 5 days, run this command once per day, and then once every hour or two on the 5th day.
	Crash file, extraction, review	Dir Copy core:<file> tftp://<ip>/<file> Capture the output of “show tech” Capture serial console output during crash if possible. This is vital if the corefile turns out to be unreadable. Contact Nortel NETS and provide information.
	Excessive CPU Load	Show load Show fsm statistics

WSS software debug command descriptions

Command	Description
Set log buffer severity warning	Sets the WSSs internal syslog buffer severity to “WARNING” which is slightly more verbose than the default of “ERROR”. This allows you to see more messages which can help diagnose issues.
Set trace authorization level 10	Sends information to the trace log on the mapping of RADIUS attributes to appropriate WSS functions. This includes the success or failure of these mappings, and is useful in diagnosing failures where RADIUS was successful but the client is still being rejected.
Set trace autorf level 10	Sends information to the trace log on the behavior of the auto-tune channel and power features. This includes current neighbor information and decisions made by the algorithm.
Set trace dap level	Displays activity related to the WSS code which manages DAPs. This includes moving DAPs
Set trace dns level 10	Displays activity for the internal DNS client. This includes information on how DNS is intercepted for Web Portal clients when they are initially bringing up the login page.
Set trace dot1x level 10 mac-addr <mac>	Primarily shows the client progressing through the 802.1X state machine, but also includes useful information on falling through to MAC, Web Portal, and Last-Resort authentication. Includes identifying information on packets sent and received, along with timeouts and retransmits. For WPA client clients this also includes transmit/receipt of 4-way and 2-way handshake packets.
Set trace httpd	Displays activity for the internal web server. This will display events every time WMS contacts the switch, as well as information related to Web-Portal and Web View.
Set trace radius level 5	Sends RADIUS packet decodes to the trace log. This is useful for verifying which RADIUS attributes are being sent by the RADIUS server.
Set trace rf_client level 10	Displays information about clients of Rogue APs. This includes rogue classification messages.

Set trace rf_master level 10	Displays debug information for rf-detect related activity on the seed switch.
Set trace rf_slave level 10	Displays debug information for rf-detect related activity on the member switches of a mobility domain.
Set trace sm level 7 mac-addr <mac-addr>	Sends information on client's state changes within the session manager state machine. This includes low-level 802.11 events like "Association", "Re-Association", and "Disassociation". Specifying the mac-addr parameter will restrict the entries to be relevant to only a single mac-address, and is strongly recommended to be used whenever doing SM tracing.
Set trace tapa	Summarizes the tapa traffic and gives specific details on image downloads and configuration packets sent to the AP.
Set trace web level 10	Sends information on web-portal authentications to the trace log.
Show aaa	Displays configuration information as well as the current timeout and up/down status on configured RADIUS servers.
Show auto-tune attributes	Displays a table of the current auto-tune values that the algorithm uses to measure channel quality.
Show auto-tune neighbors	Displays a list of all APs neighboring a given radio, including BSSID and RSSI values for each.
Show arp	Lists the ARP table internal to the switch.
Show crypto certificate {admin|web|eap}	Shows a decode of the certificate in the specified cert-store. Useful for verifying signature on cert, time/date validity, and common name on the cert.
Show dap counters [#]	Displays radio statistics on the daps for everything from noise-floor to per-packet data rates.
Show dap etherstats	Displays packet statistics for the DAPs Ethernet ports.
show dap qos-stats	Displays transmit packet counts for each queue on the AP.
Show dap status	Shows current operating parameters for DAPs as well as serial#, IP, state, SSIDs, BSSIDs, current channel/power and other useful information.

Show dap status terse	Abbreviated version of “show dap status” which is very useful for “at-a-glance” status on DAPs and APs.
Show dap unconfigured	Shows DAPs which are contacting the mobility domain but are not configured on any of the WSSs. Contains all the required information to configure a DAP (model, serial).
Show dot1x clients	Shows usernames, MAC address, dot1x state, and encryption of currently connected clients
Show dot1x config	Shows current dot1x related configuration parameters.
Show dot1x stats	Shows counters of various portions of the dot1x state machine.
Show fdb	Shows the forwarding database within the switch. This is useful for verifying L2 forwarding paths through the switch
Show fsm statistics	Shows the amount of time the CPU has been spending in various portions of NOS’s finite state machine. When high CPU load is observed running this command at regular intervals will help narrow down which portions of the code are consuming the most CPU time.
Show ip https	Displays IP addresses of clients that have connected to the HTTP server as well as the time since last connection. Useful for checking to see if multiple WMS servers are talking to one WSS.
Show ip route	Displays the routing table for packets sent from the WSS. The WSS does not route client packets, so this has no impact on client data at all.
Show load	Shows average CPU load since boot as well as the average CPU load since the command was last run (labeled “delta”).
Show memory sum?	Shows memory allocation (elements and bytes) for various portions of processes. Run this command regularly on a particular process to help find memory leaks.
Show mobility-domain status	Displays the current status (up/down) as well as IP addresses of each switch in the mobility domain. This is only from the perspective of the current switch, so you should compare outputs from separate switches when debugging mobility domain issues.

Show roaming vlan	Displays the vlans which are currently available for tunneling across the mobility domain, as well as which switches are advertising each one.
Show security acl hits	Displays the number of hits on each ACL configured on the switch. You must use the command “hit-rate-sample <#>” to enable counters, with <#> being equal to the number of seconds between each sample. Use larger sample rates on production networks to avoid impacting performance.
Show security acl info all	Displays all ACE entries and all ACLs.
Show security acl map <acl-name>	Displays what ACLs have been mapped to. This is particularly per-user ACLs.
Show security acl resource-usage	Displays general statistics and counters on ACL usage on the WSS
Show sessions	Lists all active sessions on the WSS. Includes username, IP address, VLAN, AP and radio#.
Show sessions network session-id #	Show information on a specific client session. This includes detailed information like packet stats (wireless only), authentication server, encryption type, etc…
Show sessions network verbose	Lists active sessions along with the last 5 APs the client was associated to and how long ago it was done.
Show tunnel	Shows tunnels which have been initiated to or from the WSS, including current status (active/dormant).
Show vlan	Displays the VLANs/ports/tags currently active on the WSS, including tunneled VLANs.
Traceroute <ip>	Same as Unix traceroute command, will initiate it from the system IP address of the WSS.

WMS troubleshooting areas

URL or Local File	Description
https://<ip addr>	Accesses the WMS services log. Note: By default you will only be able to access the log from the WMS server itself. You will need to "Allow Remote Access" in Tools->WMS Services Setup in order to access this URL across the network (not recommended for security reasons). This logfile contains useful information on what the WMS service is doing, and when it is doing it.
https://<ip addr>/memory	Gives information on the current memory usage of the WMS service. It also has a button which forces "Garbage Collection" in the Java Virtual Machine Repeated visits to this URL over time are useful for monitoring memory leaks.
\<install dir>\conf\services-conf.xml	This is the service configuration file. You can modify this file (not recommended) to change behavior of the service, including which TCP port it binds to on startup. This file also contains the WMS Service login information and configuration.
\<install dir>\log\	This directory contains the full logs for all aspects of WMS. The contents of this directory are important when reporting issues with WMS.
\<install dir>\services-db\	This directory contains the 30 day rolling history database of RF, user, and Rogue data. If the database becomes corrupt (status of various devices becomes blue within WMS explorer, but they are up and able to communicate) you can stop the WMS service, delete this directory, and they restart the service to recover.

Troubleshooting scenarios

Client unable to connect to wireless network

Typical symptoms:

complete inability of the client to connect to the wireless network

No user session in “show sessions” command output, or only user’s mac-address listed with no VLAN, IP, and username.

Troubleshooting steps:

Update the client.

This includes getting the latest drivers for the NIC as well as OS patches and updated supplicants.

Get the output from “show tech”
You can use the “show tech” (or equivalent in an OEM box) commands to output common information used to troubleshoot problems. It includes the configuration file as well as the output of “show ap status”, recent syslog entries, and lots of other good information. TAC will always request it, so you might as well start off by getting it.

Set the system log to “Warning” severity

```
set log buffer severity warning
```

This will allow you to see authorization failure messages indicating incorrect VLAN names and other common authorization failures in the system log buffer.

Turn on dot1x tracing level 10, restricted to one problem client’s MAC address

```
set trace dot1x level 10 mac 00:01:02:03:04:05
clear log trace
```

Always start with DOT1X tracing, regardless of whether or not the system is using 802.1X authentication. This will show you the order authentications are attempted in, and whether or not 802.1X, Mac-Auth, Web-based AAA or Last-resort are attempted. With 802.1X clients, pay attention to the username in the trace, and whether or not it matches any network access rules.

Attempt to authenticate from the problem client and then check the logs.
After attempting to login, check both the system log and the trace log for interesting messages.

If there are no dot1x messages, the client is failing at a very low level, and probably isn’t even attempting to associate to the AP. Performing an over-the-air trace will verify if this is occurring. Some devices may require the following settings (especially older devices) for connectivity:

Enable “long preamble” in the Radio Profile.

Disable “WMM” in the Radio Profile

Set the radio-type to be 802.11b instead of 802.11g

If you can see that we are sending packets and the radius server isn’t accepting them, see if the customer will install Ethereal on the radius server or hook up an Ethernet sniffer directly in front of the radius server. If the packets leave the WSS and don’t arrive at the radius server it’s some sort of routing issue (check ip route table on the WSS and have customer check intermediate routers). If the packets are arriving at the radius server and it’s not acknowledging them, have the customer check the “radius client” configuration and the shared secret (again). If the shared secret is incorrect, or the client is not defined, Microsoft IAS will silently discard the packet.

If you see a “Status:FAIL from AAA” message in the trace log, then it means that the client failed authentication and the certificate or username/password is invalid. Check the log files on the Radius server for more information, and the client configuration. If you don’t see anything in the log files on the Radius server, then double-check the shared secret configured for the radius server (both on the WSS and on the radius server). You can also turn on radius tracing to see a decode of the packets we are sending to radius.

If you see an authorization failure, one of the radius attributes is incorrect, not present, or the VLAN the user is configured for is not available. The system log message should indicate which attribute is present, and what it is configured for. Go through the configuration to find out if it’s configured. Pay close attention to the capitalization of the attribute because the system used to be case-sensitive and there may still be some areas which still are.

If you see “excessive retransmits, deleting client” then something is not configured properly in the client. This means that the client is not answering 802.1X queries at some point. Review that section of the trace log and determine what part of the authentication you are in. If this is at the very beginning (identity requests) then have the customer check the basic configuration on the client and look for 3rd party dot1x supplicants like AEGIS. These can be installed by default with the NIC’s management programs. Check the properties of the NIC where it lists protocols (like TCP/IP and “Client for Microsoft Networks”) and uncheck any unfamiliar looking items. Also check to be sure that the client has the appropriate CA certificate and that all certificates involved haven’t expired.

Switch stability

Typical symptoms:
All DAPs on a switch rebooting simultaneously
“Core” files other than command_audit.cur” showing in the output of “dir”
Sluggish CLI and occasional missed ping responses.

Troubleshooting Steps:

Check for core files

Do a “dir” command and check for the presence of core files. Use the command “`Copy core:<file> tftp://<ip>/<file>`” to transfer the core files to the tftp server. Then contact NETS and provide the output of the “show tech” command.

NOTE: The file “command_audit.cur” is not a core file from a crash, even though it has the “core:” prefix.

Check the frequency of the cores for patterns

If the switch is crashing with a fairly regular period it is probably a memory leak. Periodically log the output of the command “show mem summary proc netsys” (replace netsys with whichever process is named in the corefile) to get a sample of memory usage on the switch over time, and send the logs to Nortel NETS. If the cores are happening at a regular interval, increase the frequency you run the command on the day when the core would be expected. The memory leak could be in a process other than the one which cores, so it may be necessary to repeat this with other processes as well.

Capture serial console output during a crash

If possible, setup a laptop to log all output from the serial port and leave it running until the switch crashes again. This is especially important if the switch isn’t leaving core files, or if the corefiles aren’t revealing much information about the crash.

Investigate possible causes

Try undoing the most recent configuration changes to see if they are related to the crashing. Attempt to identify what event is causing the crash (this may not be possible on a production network). TFTP the “command_audi.cur” file from the switch and look for configuration changes prior to the first crash.

Check CPU Load

Run the command “show load” and then wait for a few minutes and run it again. The “delta” value the second time you run the command will indicate the average CPU load for the period between the commands being run. CPU loads higher than 50% over a 5 minute period are likely indicators of a problem. If the CPU is pegged at 100% there is definitely a problem, and you should run the command “show fsm statistics” every couple of minutes and provide the output to NETS. This command will display CPU activity used for specific portions of the code and allows Engineering to narrow down which portion of the code is causing the CPU load.

WMS service database corruption

Typical symptoms:

Status of WSS and APs showing as blue (unknown) or is not accurately reported even though devices are known to be up and operational.

Troubleshooting steps:

> Check the Operating System logs to determine if the OS has been shutdown improperly. Most database corruption issues in WSS Software 4.0.20+ can be attributed to improper shutdown.

Verify the system meets the minimum requirements for WMS. If you are running both the service and the client on the same machine you must add the memory requirements together and use at least highest CPU requirement.
If neither of the first two steps apply, copy the contents of the “services-db” and “logs” directory then contact NETS for analysis.
Stop the WMS service as appropriate for your host operating system.
delete the <install-dir>\service-db directory
Start the WMS service as appropriate for your host operating system. WMS should now show correct status for all equipment after the next polling cycle.

Troubleshooting auto-tune channel

Typical symptoms:
Intermittent client disconnects, frequent channel changes.

Troubleshooting steps:

Verify that active-scan is enabled.
Auto-tune channel depends on active-scan to build the neighbor lists. If active-scan is disabled, auto-tune channel must also be disabled. This will typically occur in VoIP environments where the handset providers typically require auto-tune to be disabled anyway.
Set the syslog severity to “Notice” to show auto-tune operations.
Look for correlations between disconnect complaints and auto-tune channel events to verify that disconnects are due to Auto-Tune Channel. Display filtered logs on the WSS using the following command:

```
Show log buffer matching “Changing channel”
```

Messages will appear in the logs similar to:

```
Tue Jan 31 20:02:06 2006: <133>Jan 31 20:01:26 172.17.11.1
AUTORF_NOTICE: Changing channel on radio dap 14/1, 11->6: Too many
neighboring APs on channel(32098/36000)
```

Each of these messages will indicate the reason for the change, and you can try to correlate the DAP and timedate stamp with user complaints. Collect the entire log (unfiltered) for analysis. If the system is continually changing channels it has not converged for some reason, and the logs will assist in determining this.

Collect the output of "`show auto-tune neighbors`"

This command displays all of the neighbors seen by each radio. This will give you an idea of how many other APs are visible from each radio and what their signal strength is.

Turn on autorf tracing level 10 and capture the output over an extended time period.

This will display more detailed information on what the radio sees and why it changes it is channels, and this will provide additional debugging information to assist Engineering in determining why the system is changing channels.

Disable auto-tune channel

If the disconnects are too disruptive for the customer, use the "Apply Auto-Tune Settings" option under the "Manage" menu in WMS to convert the dynamic settings to static configuration and disable the auto-tune feature.

As of WSS Software version 4.0.21, the auto-tune algorithm still does not take into consideration client connectivity when it decides to change channels. Most customers value connectivity more than dynamic adaption of channels, so Auto-Tune channel should be used to set the initial channel set and then it should be converted to a static configuration by using the WMS "Apply Auto-Tune Settings" option under the "Manage" menu.

Modifications to the Auto-tune channel feature are in progress, and the first set should be implemented in the MSS 4.0.22 maintenance release in early March.

Troubleshooting auto-tune power

Typical symptoms:

APs appear not to be tuning power; client signal strength appears to vary widely and rapidly.

Troubleshooting steps:

Verify that active-scan is enabled

Auto-tune channel depends on active-scan to build the neighbor lists. If active-scan is disabled, auto-tune channel must also be disabled. This will typically occur in VoIP environments where the handset providers typically require auto-tune to be disabled anyway.

Set the syslog severity to "Notice" to show auto-tune operations

Look for auto-tune power level change messages. The algorithm will turn the power up if it sees clients retransmitting packets at a rate exceeding “max-retransmissions” (configured on the radio), and this is frequently the reason. You may need to reset on of the APs and monitor the logs – if it’s already tuned to maximum power in response to client retransmissions it will not log further messages. Rebooting the AP will set it back to the baseline power and show modifications from there.
If a client is reporting rapid signal strength fluctuations be sure to check the logs around this time to see if the AP’s power is raising or lowering in response to the client. If it is, skip to step 4.

Collect the output of “show auto-tune neighbors”

This command displays all of the neighbors seen by each radio. This will give you an idea of how many other APs are visible from each radio, and how loud they are. The baseline power will be adjusted so that the radio will just barely be able to transmit to the Nth farthest AP. For 802.11bg, N=3. For 802.11a N=8. If the Nth AP has a low RSSI, the radios power will be relatively high.

Disable the “reach-out” functionality of Auto-Tune

The AP will attempt to increase power to improve a client’s connectivity. This behavior will tend to leave APs operating closer to maximum power. If you want to disable this functionality you will need to adjust the “Data Retransmission” value on each radio to be 100% instead of the default of 10%. This will for the APs to stay at the “initial” power setting as determined by the Nth farthest AP.

Disable Auto-Tune Power

If clients are still experiencing issues, use the “Apply Auto-Tune Settings” option under the “Manage” menu in WMS to convert the dynamic settings to static configuration and disable the auto-tune feature.

Data Rate Enforcement

If the data rate enforcement is having problem, collect the following CLI output:

show ap counters

The **show ap counters** command lists the number of times a client attempts to connect with a disabled data rate.

For example,

Syntax wss# sh ap counters

AP: 2		**radio: 2**	
LastPktXferRate	6	PktTxCount	428475
NumCntInPwrSave	0	MultiPktDrop	0
LastPktRxSigStrength	-57	MultiBytDrop	0
LastPktSigNoiseRatio	38	User Sessions	1

TKIP Pkt Transfer Ct	0	MIC Error Ct	0
TKIP Pkt Replays	0	TKIP Decrypt Err	0
CCMP Pkt Decrypt Err	0	CCMP Pkt Replays	0
CCMP Pkt Transfer Ct	0	RadioResets	0
Radio Recv Phy Err Ct	0	Transmit Retries	30469
Radio Adjusted Tx Pwr	18	Noise Floor	-96
802.3 Packet Tx Ct	0	802.3 Packet Rx Ct	0
No Receive Descriptor	0	Invalid Rates	395

	TxUniPkt	**TxMultiPkt**	**TxUniByte**	**TxMultiByte**	**RxPkt**	**RxByte**	**Undcrpt Pkt**	**UndcrptByte**	**PhyErr**
6.0:	95964	311251	18476331	64275631	16931866	0	0	0	110
9.0:	0	0	0	0	3	0	0	0	3
12.0:	1835	3925	195798	551573	2	0	0	0	1
18.0:	0	0	0	0	1	0	0	0	2
24.0:	0	28	0	4227	4	0	0	0	0
36.0:	0	0	0	0	2	36	0	0	0
48.0:	0	0	0	0	24	203	0	0	1
54.0:	1275	5835	131802	3238	59663	0	0	0	113
TOTL:	97849	316479	18677964	64963233	3443	91768	0	0	230

show service-profile sp1

In the service profile, check whether the data rate setting is correct or not.

show radio-profile rp1

In radio profile, check whether the data rate enforcement is enabled or not.

Mobility-Domain troubleshooting (seed and secondary-seed)

If the mobility domain is having problem then collect the CLI Output combinations from each of the following mobility domain members:

- show mobility-domain
- show mobility-domain data
- show mobility-domain config
- show cluster
- show tech-support

If any of the mobility domain members are not active then verify the configuration. Also, from the other cluster members issue a ping request to the member that is no longer active to determine if there is an active path to the mobility-domain member.

RF Analysis

If "coverage hole", "high utilization", "rf interference" performance alarms are not available in WMS alarms then open a trouble ticket. Before opening a trouble ticket, collect the ZIP file that includes WMS logs and a snapshot of error message that may occur or have occurred. To collect the ZIP file go to WMS menu and then to **Help-> Report Problem** and enter the stated information and save it.

The path information of the zip file is provided at the bottom of the WMS screen. Now, you can open a trouble ticket with the above compiled information.

In WMS under alarms, if there are "coverage hole", "high utilization", "rf interference" alarms are not available then open a trouble ticket. Before opening a trouble ticket, collect the ZIP file that includes WMS logs and a snapshot of error message that may occur or have occurred. To collect the ZIP file go to WMS menu and then to **Help-> Report Problem** and enter the stated information and save it.

The path information of the zip file is provided at the bottom of the WMS screen. Now, you can open a trouble ticket with the above compiled information.

For both instances check for the RF Threshold settings and provide the information. This can be accomplished in the following two ways:

- WMS-> Monitor-> Sites -> Floor View -> Change RF Threshold
- WMS-> Services-> Setup-> Monitoring Settings

RF Visualization

RF Visualization is an extension of RF analysis. Hence, if there are performance alarms generated for RF Analysis, highlight the alarm and then go to floor view. In the floor view, it has to indicate where the AP is located for the alarm generated. If you determine that the alarm condition is valid, and floor view is not available then open a trouble ticket. Before opening a trouble ticket, collect the ZIP file that includes WMS logs and a snapshot of error message that may occur or have occurred. To collect the ZIP file go to WMS menu and then to **Help-> Report Problem** and enter the stated information and save it.

The path information of the zip file is provided at the bottom of the WMS screen. Now, you can open a trouble ticket with the above compiled information.

Voice Monitoring

If the QOS level is not being incremented properly under the statistics then verify that all QOS settings and markings are set throughout the network between the AP and WSS.

Voice monitoring helps to determine if packets are marked appropriately in the transmit direction or not.

In WMS, to view client statistics for each queue perform the following steps:

WMS -> Monitor-> Equipment -> Choose WSS-> Client SSID Details -> Highlight a user session -> Session Details -> Statistics.

In CLI, to view client statistics for each queue perform the following steps:

1. Show sess net session-id <client session number>
2. Another CLI command that is not client specific is "show ap qos-stats". This provides an aggregate number and is not session specific.

RfLink

In WMS this feature is named as rflink and in CLI it is named as rfping. This feature provides information on client session health by reporting RSSI, SNR, Round Trip Time, retries, and rate. If WMS is reporting incorrectly compare to the CLI then collect the following information:

CLI command:

Syntax WSS2360-1# rfping session-id <session> verbose

or

Syntax WSS2360-1# rfping mac <mac address> verbose

Then compare with the WMS output that can be accessed through

WMS -> Monitor-> Equipment -> Choose WSS-> Client SSID Details -> Highlight a user session -> rflink.

If you still report issue with rflink, execute a wireless packet capture near the AP client which is connected to filter on the client session.

Scheduled Reports and E-mail

If there is problem with the reports being generated or the email being sent by WMS then open a trouble ticket. Before opening a trouble ticket, collect the ZIP file that includes WMS logs and a snapshot of error message that may occur or have occurred. To collect the ZIP file go to WMS menu and then to **Help-> Report Problem** and enter the stated information and save it.

The path information of the zip file is provided at the bottom of the WMS screen. Now, you can open a trouble ticket with the above compiled information.

Unethered mesh AP unable to connect to portal AP

If you are unable to connect unethered mesh AP to portal AP, then check for the following:

1 Check that the portal AP is enabled for mesh services

Command line: **show ap status terse**. Check that the command output shows a flag (p) indicating that the AP is enabled as a portal AP.

Sniffer: Sniff the air on the appropriate channel and verify that the portal AP is broadcasting the SSID.

2 Check the Tx power levels on the portal AP

Command line: **show ap status terse**. Verify that the radio is enabled and verify Tx power levels.

Tip: For optimal results, the Tx power should be 10dbm or higher.

3 Check if the unethered mesh AP is configured on the switch where the portal AP resides.

Command line: **show ap config**. Verify that the AP has been configured on the switch. Otherwise, use the Auto-AP feature.

4 Check if unethered mesh AP has the correct SSID and pre-shared key configured.

Command line: **show ap boot-configuration <ap-num>**. Verify that mesh is enabled. Make sure SSID and pre-shared key matches the "mesh" service profile properties configured on the switch.

Tip: If available, directly connect to 2360 for troubleshooting purposes.

5 Check if untethered mesh AP received IP and WSS information.

Sniffer: Verify that the DHCP server has issued an IP address and provided WSS IP information to the mesh AP.

To verify the session is local-switched

To check the session is local-switched and in right status, perform the following:

- Check which AP has local-switching enabled and which VLAN is configured

Syntax WSS# show vlan-profile

The command output shows the AP numbers for the APs with local switching enabled, and the VLANs configured on the APs

- Check whether a session is on a local switched VLAN after clients association

Syntax WSS# show session network

The command output shows a flag (L), indicating that the session is on a local switched VLAN, under VLAN Name.

- Check whether a VLAN is local switched

Syntax WSS# show ap vlan <ap number>

The command output shows the mode of the VLAN as either local or tunnel

- Check the FDB entry of a specific AP

Syntax WSS# show ap fdb <ap number>

The command output shows the fdb entries of the AP.

Local switching enabled and the AP cannot boot

After enabling local switching if AP cannot boot up, then check for the following:

- Check the boot configuration of one AP

Syntax WSS# show ap boot-configuration <ap number>

The command shows whether AP is configured to boot up from default VLAN or a specific VLAN

- Check whether the vlan-profile configuration complies with the boot configuration

Syntax WSS# show vlan-profile

- Update the vlan-profile

Syntax WSS# set vlan-profile <profile name> vlan <vlan name> [tag <tag number>]

If AP boots up from default VLAN, then make sure this entry: default none is present in the service profile.

If AP has boot configured on a specific VLAN, then make sure the same VLAN is present in the service-profile in the format <vlan name> <vlan tag>.

Session is disconnected after roaming

Check whether the overlay mode works. In order to make Inter-WSS roaming work for local-switched sessions, the tunnel between WSSs is required, therefore the VLAN needs to be configured to support overlay mode.

WLAN Location Engine 2340 troubleshooting areas

This chapter provides information on troubleshooting WLAN Location Engine 2340 (WLE2340) and this is explained in the following sections:

- System availability
- Administrative Web User Interface
- Web User Interface
- Sensor Connection and Communication
- Tracking
- Dashboard

System availability

If the user encounters any basic difficulty in getting access to the system, which also includes the user is denied access to the Web User Interface (Web UI) or the Web UI does not run, you need to verify the following:

- General availability
- Password lost for Standard Web User Interface
- Password lost for the WLE2340 Admin User

Examples I cannot log in to the system.

General availability

If the user describes the problem as "The WLE2340 does not come up" or "We browse to the Web UI for the WLE2340, and we get the Firefox message - 'Server not Found'", then check for the following:

- Verify IP Address for the WLE2340
- Verify system availability
- Unavailability of CLI
- Availability of CLI and unavailability of standard Web UI
- Availability of Disk Space

Verify IP Address for WLE2340

You always start connectivity with the IP address, hence verify that the WLE2340 has the IP address that is being used to access it. You can do this through the Administrative UI or the Command Line Interface (CLI).

Verify system availability

To verify the system availability, perform the following:

- In the Web browser, check if the Web UI shows port 443.

- Check the Administrative Web UI through SSL on port 8003.
- Check if the command line for the WLE2340 is available, if none of the Web UI is available. This can be done at the WLE2340 with a serial connection, or remotely if remote access is enabled.

Unavailability of CLI

If the CLI is unavailable remotely, it can be possible that the SSH is not running or there is a basic issue with the WLE2340. The user needs to get physical access to the UI and then attempt to log on to the command line.

If the CLI is unavailable from a direct serial connection, then attempt to restart. If the WLE2340 does not boot it can be defective.

Availability of CLI and unavailability of standard Web UI

In this case, it is most likely that the Controller has not started.

Enter the following:

Syntax **show system status**

and verify that there are two java processes started. If not, restart the WLE2340 or the WLE2340.

Availability of Disk Space

If disk space is unavailable, then there is issue with the df command. If the **/dev/sda1** portion shows 100% full, then this is due to the logs. To verify, perform the following:

1. rm /opt/platform/server/controller/*log*
2. rm /opt/platform/server/agent/*log*
3. Edit /opt/platform/server/controller/*log* with vi.
4. Restart the WLE2340

Password lost for Standard Web UI

To check the lost password, perform the following:

- Login as another Admin User
- Verify Admin User availability

Login as another Admin User

If it is possible, login as another admin user then delete and re-create the user.

Verify Admin User availability

In this case, return the system to the initial state (having no users), so that the first access of the Web UI gives you the form to create the first admin user and declare the password for that user.

Password lost for the WLE2340 Admin User

To verify the lost password for the WLE2340 Admin user, perform the following:

Reset the Admin Password to the Factory Default

This requires physical access to the WLE2340. Connect through the serial cable connection. There is a 5-second delay before the prompt is shown. During this delay, hit the Escape key on the keyboard. The WLE2340 prompts whether to return to the default factory password ("password").

Administrative Web User Interface

The administrative Web User Interface is the series of Web pages available when connecting to port 8003 of the WLE2340 through SSL. It includes issues related to database access, system updates, and network and time configuration.

Examples "We uploaded a system update but the old version number still shows."

Network configuration

There is a general availability issue if the user is describing the problem as "The WLE2340 does not come up" or "We browse to the Web UI for the WLE2340, and get the Firefox message - 'Server not Found'".

Verify system availability for Administrative Web User Interface

To verify the system availability, do the following:

- In the Web browser, check if the Web UI shows up on port 443.
- Check the Administrative Web UI through SSL on port 8003.
- Check if the command line for the WLE2340 is available, if neither Web UI is available. This can be done at the WLE2340 with a serial connection.

Web User Interface

This includes issues related to the performance and functionality of the Web User Interface. This is for the Web pages available when connecting through SSL to the standard SSL port 443 of the WLE2340.

General Availability

There is a general availability issue if the user is describing the problem as "The WLE2340 does not come up" or "We browse to the Web UI for the WLE2340, and get the Firefox message - 'Server not Found'".

Verify system availability for Web User Interface

To verify the system availability, do the following:

- In the Web browser, check if the Web UI shows up on port 443.
- Check the Administrative Web UI through SSL on port 8003.

- Check if the command line for the WLE2340 is available, especially if neither Web UI is available. This can be done at the WLE2340 with a serial connection

Sensor Connection and Communication

It comprises of questions about getting sensors configured initially, and then maintaining them for use by the WLE2340. There are subsections for each sensor type:

- WLE2340 Locale Points
- Auto Discovered TZSP sensors provided by Trapeze, Nortel, and 3Com

Examples "The sensors all show as red in the Web UI"

Examples "I configured a sensor but it does not show at all in the Web UI"

All Sensors appear Down

The WLE2340 contains two running java processes:

- WLE2340 Agent, which communicates with the sensors and tracks devices.
- WLE2340 Controller, which provides administrative logic, handles UI requests, SOAP requests, and so on. The controller can run while the Agent is down, which shows that the WLE2340 is up and running. The WLE2340 cannot track anything without the Agent, and sensors may appear down.

Check the Agent Status

To check the agent status, browse to the **Configuration > Agents page of the Web UI**. If the Agent status is red or disabled, or white and inactive, then restart the WLE2340.

Check the Sensor IP Addresses

If the IP addresses of the sensors have changed, then proceed to the next sub-section on changed IP addresses.

Auto Discovered TZSP Sensors

APs from Trapeze, Nortel, and 3Com are capable of sending information to the WLE2340 and declaring themselves sensors. Unlike the Cisco APs, they may be in service providing coverage also, and the WLE2340 is not responsible for their configuration. Instead, when the WLE2340 sees these devices reporting data, it treats them as auto-discovered and adds them to the list of sensors.

If there is a problem with these devices acting as sensors, follow these steps to troubleshoot:

- Verify the Firmware on the Controller
- Verify the Snoop Configuration
- Check the Auto Discovery in the WLE2340
- Check the Sensor Statistics
- Firewall Settings
- Check the Agent Logs

Verify the Firmware on the Controller

The firmware on the controller must be of at least version 6.0 for official support of the AP as a sensor. One symptom that the firmware is not compliant is that the log for the Agent will fill with Array Index out of Bounds errors, or messages that the AP is reporting an illegal or unknown channel.

Verify the Snoop Configuration

On the controller, use the command **show snoop stat** to verify that the APs are supposed to act as sensors and that they are reporting data.

Also check the snoop map to make sure that the listener IP address matches the IP address for the WLE2340, and that the APs and radios that are acting as sensors are using this snoop map. The **show snoop info** command can help to get this information.

Finally, verify that the snoop mode is enabled for the APs and radios that should be acting as sensors.

Check the Auto Discovery in the WLE2340

Browse in the standard Web UI to the menu item **Configuration > Sensors**. Verify that the APs acting in snoop mode have been discovered and that they are listed with the correct IP address. If they are not, then disable them. Disable snoop mode from the controller. Then re-enable snoop mode.

Check the Sensor Statistics

You can look at statistics for the sensors with the show sensors command, from the command line interface. This gives the name, IP address, status (operational, a Boolean field), the number of devices tracked by the sensor and a packet count. The packet count should be raised for all active sensors that are reporting monitored wireless traffic. If the devices are not operational or the packet count is static, it is likely that the APs are not in snoop mode, or somehow unable to communicate with the WLE2340.

Firewall Settings

The APs acting in snoop mode sends TZSP formatted information through UDP on port 37008 to the WLE2340. If there is a firewall or VLAN configuration with port restrictions between the sensor APs and the WLE2340, this traffic may be blocked. Verify that the communication is clear between the two.

Check the Agent Logs

Problems with AP Communication are logged in the Agent log. Check those logs for problems that may not be discussed here.

Tracking

This covers questions related to the accuracy and latency of the tracking information provided by the WLE2340.

Examples "Why are certain devices tracking differently when in the same place?"

Stationary Devices (APs) move or Tracked Incorrectly

The WLE2340 monitors and tracks all devices producing wireless traffic, including APs and even APs doing dual duty as sensors. The WLE2340 tracks by matching a device to Fingerprints. Sometimes, an AP can be tracked to a nearby Fingerprint that offers the best match, but physically does not track location for the APs. This can be disconcerting to clients who know for sure where the AP is and expect it to track there.

The AP is possibly 10 feet off the floor in the middle of the hallway, whereas all Fingerprints were taken in rooms off of the hallway, at desk level. The nearest Fingerprint match may not be ideal.

To resolve this, the user can create a new Fingerprint specifically for the stationary device, and calibrate it using the MAC address of the stationary device. This pins the stationary device to its own Fingerprint, ensuring accuracy. Remember to add the Fingerprint to Dashboard Layouts also for accurate visualization, also.

Dashboard

The Dashboard client is used to connect to the WLE2340 and provide a graphical representation of tracking. This class comprises issues and questions regarding its use.

Connectivity

The most common issue with Dashboard is the inability to connect to the WLE2340, despite having valid credentials. The solution is to ping WLE2340 host name from the PC running the Dashboard (to ping use windows command prompt) and IP address at Windows hosts file (C:\WINDOWS\system32\drivers\etc) on the PC in which Dashboard is installed.

In this situation it is the case that there is a problem with forward or reverse lookup of the WLE2340 hostname in the local DNS. To connect to dashboard, do the following:

1. Find the hostname for the WLE2340. This is available in the Administrative UI in the **Configuration > Networking** section.
2. Check if this resolves to the correct IP address.

Make sure that there is forward and reverse DNS entries for the WLE2340 hostname.

Device display

Devices tracks and shows the Device List of the WLE2340 Web UI, but does not display correctly in Views of the Dashboard interface. To view the device list of the Applicance Web UI, to the following:

Verify Device List accuracy

Check the information that the Dashboard is getting from the WLE2340 to see if it matches up with the Device List. To view the information, perform the following:

1. From the Server Connection list in the left hand pane, right click on Locales.
2. Choose the "View Devices" option to get a full Device List.

If this does not match with the Device List in the Web UI, look for error messages in the lower left hand corner of the Dashboard, and also check the Dashboard logs for errors.

Check Properties of Layout Palette Elements

View the layouts to verify that the Locales where devices are tracked are listed and bound. Do not trust the graphical display that shows the name of the locale. Open the Layout Palette and then use the **Select** tool to select individual Locales. Once you select a Locale, check the **Layout Properties** panel of the Layout Palette and verify that the name is selected in the drop-down list.

If the name of the Locale is not selected in the drop down list, then it means the locales have become unbound. For instance, this can happen if the WLE2340 changes IP addresses.

The solution is to re-bind all Locales, Fingerprints, and Sensors by highlighting them and selecting the appropriate label in the Properties panel of the Layout Palette.

Common Troubleshooting Techniques for WLAN Location Engine 2340

Remote Access to the WLE2340 Command Line Interface

Allowing remote Access to the CLI

By default, the WLE2340 does not allow remote connectivity to the command line interface (CLI). However, it is possible to enable this by logging to the CLI. Use the **enable** command for access to privileged commands and enter the admin password for the WLE2340. To enable the sshd process and allow remote access to the system, use **set ssh** command.

The following CLI commands are available for troubleshooting:

Command	Description
show system	Takes one argument and returns information about the run time state of the WLE2340.
show system uptime	Reports the current system time, and how long the system has been up since last reboot. This information also shows on the landing page for the Administrative Web UI on port 8003.
show system version	Reports the version number of the system. This information also shows on the first page for the Administrative Web UI on port 8003.
show logs status	Shows memory and processor information, including average load and a process list.
show sensors	Lists all sensors registered in the system by name and IP address, then indicates current status, the number of devices seen by the sensor, and the packet count for the sensor.
show logs	Used to dump the log contents to standard output. Show logs takes one or more arguments to indicate what logs to dump.
show logs appliance	Shows the appliance log including system errors, remote session logins, database access, and so on.
show logs system controller	Shows the log for the Controller process. Useful for debugging availability issues and Web UI issues.
show logs system agent	Show the log for the Agent process. Useful for debugging sensor connectivity and tracking issues.

show interface eth0	Shows network information for the appliance.
show serial-number	Displays the appliance serial number.

The Dashboard Logs

There are logs for the Dashboard available on the client machine. In a Windows install, these files can be found in the following path:

C:\Documents and Settings\<username>\.dashboard\dashboard\var\log

Debug trace walkthroughs

Dot1x level 10 trace of WPA/TKIP with local PEAP-MSCHAPv2

DOT1X Apr 11 20:45:37.685261 DEBUG DOT1X-CLIENT: new wireless client from 00:0d:54:98:99:6d on port 16, radio 2

DOT1X Apr 11 20:45:37.685308 DEBUG DOT1X-STATE: 00:0d:54:98:99:6d transition from NOTHING to CONNECTING

You will see this sort of message frequently. It informs you of the client's changes in the 802.1X state machine.

DOT1X Apr 11 20:45:37.685341 DEBUG DOT1X-STATS: 00:0d:54:98:99:6d, enters connecting --> 139

DOT1X Apr 11 20:45:37.685389 DEBUG DOT1X-CLIENT: 00:0d:54:98:99:6d associated with a WPA IE

The client is configured for WPA.

DOT1X Apr 11 20:45:37.685410 DEBUG DOT1X-CLIENT: TKIP cipher in IE

Using Tkip…

DOT1X Apr 11 20:45:37.685427 DEBUG DOT1X-CLIENT: 802.1X authentication in IE

And WPA is configured for 802.1X instead of PSK.

DOT1X Apr 11 20:45:37.685447 DEBUG 00:0d:54:98:99:6d didn't send a PMKID in her RSNIE

The client is not attempting to do an 802.11i fast-roam by sending a PMK ID in the association request. This message is completely normal for WPA clients. WPA2 clients should (but don't have to) send a PMK ID when they associate.

DOT1X Apr 11 20:45:37.685475 DEBUG DOT1X-PACKET: setting id to networkid=slipshod-tkip,nasid=nos-3.0,portid=16 in request

After a client associates we always send an EAP Identity request if 802.1X is configured for that SSID. This message indicates what the contents of the ID request will be.

DOT1X Apr 11 20:45:37.685503 DEBUG DOT1X-PACKET: EAPoL EAP packet of 57 bytes w/id 1 (with retransmit set) sent to 00:0d:54:98:99:6d

This packet indicates that we sent the ID request with an EAP id value of 1. The EAP id values are used to match responses with requests.

DOT1X Apr 11 20:45:37.685536 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 5 seconds

Setting a retransmit timer (this is the dot1x tx-period)

DOT1X Apr 11 20:45:37.696820 DEBUG DOT1X-PACKET:EAPoL START packet received from 00:0d:54:98:99:6d

We received an EAPoL START packet from the client. Clients MAY initiate 802.1X by sending this packet, and Microsoft clients tend to always do this regardless of whether or not you've sent them an EAP Identity request. EAPOL Start packets do not have an EAP id value. They are intended to "kickstart" the authenticator (WSS) so it sends an EAP id request.

DOT1X Apr 11 20:45:37.696850 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d

Canceling the previous retransmit timer. You'll see a lot of these.

DOT1X Apr 11 20:45:37.696879 DEBUG DOT1X-STATE: 00:0d:54:98:99:6d transition back to CONNECTING

We're resetting the "CONNECTING" state because the client sent an EAPOL Start

DOT1X Apr 11 20:45:37.697012 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 5 seconds

DOT1X Apr 11 20:45:37.697034 DEBUG DOT1X-CLIENT: retransmit packet to 00:0d:54:98:99:6d

We’re retransmitting the previous packet (the EAP Identity Request).

DOT1X Apr 11 20:45:37.746255 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 1, len 17, received from 00:0d:54:98:99:6d

We’ve received a response to the EAP request with id 1 (in this case that’s the EAP Identity request we just sent).

DOT1X Apr 11 20:45:37.746285 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d

DOT1X Apr 11 20:45:37.746360 DEBUG DOT1X-CLIENT: glob '**' matches 'NORTEL\tash', ssid 'slipshod-tkip' matches 'slipshod-tkip': eap_type=25

At this point the WSS knows the outer username of the client, and begins to compare this username to the userwildcards on the “set authentication dot1x…” rules. This configuration is very simple so it matches the first one. If there were additional authentication rules in front of this one, they would be displayed in order you would see “does not match” instead of “matches”. The eap_type field is an internal number indicating which EAP type is configured on network access rule. Eap_type 25 is PEAP-MSCHAPv2, 254 is “pass-through”, ??? is EAP-TLS.

DOT1X Apr 11 20:45:37.746385 DEBUG DOT1X-CLIENT: EAP-ID resp for NORTEL\tash at 00:0d:54:98:99:6d doing PEAP

AAA has decided to do PEAP for this user based on the network access rule.

DOT1X Apr 11 20:45:37.746682 DEBUG DOT1X-STATE: 00:0d:54:98:99:6d transition from CONNECTING to AUTHENTICATING

DOT1X Apr 11 20:45:37.746705 DEBUG DOT1X-STATS: 00:0d:54:98:99:6d enters authenticating --> 11

DOT1X Apr 11 20:45:37.746788 DEBUG DOT1X-PACKET: EAPoL EAP packet of 10 bytes w/id 2 (with retransmit set) sent to 00:0d:54:98:99:6d

Sending the next EAP packet (which is the EAP-type negotiation).

DOT1X Apr 11 20:45:37.746820 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds

DOT1X Apr 11 20:45:37.747105 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 1, len 17, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.747136 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d

Here we see the client sending back a response for id 1 again. This happens frequently with Microsoft clients because both sides are initiating the 802.1X conversation.

DOT1X Apr 11 20:45:37.747182 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 5 seconds

DOT1X Apr 11 20:45:37.782314 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 2, len 80, received from 00:0d:54:98:99:6d

The client has finally caught up and sends back a response to the PEAP request.

DOT1X Apr 11 20:45:37.782339 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d

DOT1X Apr 11 20:45:37.783715 DEBUG DOT1X-PACKET: EAPoL EAP packet of 1414 bytes w/id 3 (with retransmit set) sent to 00:0d:54:98:99:6d

This is the beginning of the transmission of the server certificate used for the outer encryption tunnel in PEAP. From here the next several packets are the outer encryption processing. Incidently, if you look at the packets with a wireless sniffer you'll be able to see the comments in the x.509 certificate.

DOT1X Apr 11 20:45:37.783764 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds

DOT1X Apr 11 20:45:37.811835 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 3, len 6, received from 00:0d:54:98:99:6
DOT1X Apr 11 20:45:37.811875 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.811964 DEBUG DOT1X-PACKET: EAPoL EAP packet of 975 bytes w/id 4 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.811991 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:45:37.909013 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 4, len 128, received from 00:0d:54:98:99:6d

DOT1X Apr 11 20:45:37.909044 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.925427 DEBUG DOT1X-PACKET: EAPoL EAP packet of 57 bytes w/id 5 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.925464 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:45:37.962307 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 5, len 6, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.962336 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.962565 DEBUG DOT1X-PACKET: EAPoL EAP packet of 84 bytes w/id 6 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.962596 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:45:37.963605 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 6, len 40, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.963633 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.963702 DEBUG DOT1X-CLIENT: glob ** matches NORTEL\tash eap_type=25

The first phase of PEAP has completed, and now the inner MSCHAPv2 exchange is starting. This is the inner username. In Microsoft clients the inner and outer names are always the same. In other clients they can be different and the outer name is frequently “anonymous” or some variation therein.

DOT1X Apr 11 20:45:37.963797 DEBUG DOT1X: asked to change name NORTEL\tash at 00:0d:54:98:99:6d to NORTEL\tash

DOT1X Apr 11 20:45:37.963865 DEBUG DOT1X-PACKET: EAPoL EAP packet of 105 bytes w/id 7 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.963895 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:45:37.981434 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 7, len 94, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.981464 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.982306 DEBUG DOT1X-PACKET: EAPoL EAP packet of 82 bytes w/id 8 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.982343 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:45:37.983318 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 8, len 29, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.983348 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.983460 DEBUG DOT1X-PACKET: EAPoL EAP packet of 42 bytes w/id 9 (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.983490 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds

DOT1X Apr 11 20:45:37.984333 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 9, len 38, received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.984361 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.984709 DEBUG DOT1X-PACKET: EAPoL EAP packet of 8 bytes w/id 10 (without retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.984828 DEBUG DOT1X-STATE: 00:0d:54:98:99:6d transition from AUTHENTICATING to AUTHENTICATED

Since this example is being processed entirely on the WSS (local), there is no "PASS from AAA" statement, instead it jumps right to the "AUTHENTICATED" state.

DOT1X Apr 11 20:45:37.984957 DEBUG EAPOL-STATE: request authorization for NORTEL\tash at 00:0d:54:98:99:6d

Authorization is beginning. This is a very common area for configuration mistakes that prevent clients from connecting.

DOT1X Apr 11 20:45:37.985771 DEBUG DOT1X-STATE: NORTEL\tash at 00:0d:54:98:99:6d is authorized

No error message here, everything was processed successfully. If you wanted to see the authorization process you could turn on "set trace authorization". Generally you won't need to because warnings will be displayed in the syslog and trace log when a client fails due to authorization.

DOT1X Apr 11 20:45:37.986004 DEBUG DOT1X: begin a WPA 4way handshake with 00:0d:54:98:99:6d

Because this is WPA, we have a 4-way handshake for the unicast session key. The handshake follows:

DOT1X Apr 11 20:45:37.986030 DEBUG DOT1X: Sending message 1 of the 4way Handshake

DOT1X Apr 11 20:45:37.986055 DEBUG DOT1X-PACKET: EAPoL packet of 99 bytes (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:37.986082 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 5 seconds
DOT1X Apr 11 20:45:37.987021 DEBUG DOT1X-STATE: TX RSC is 0 for client NORTEL\tash at 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.007289 DEBUG DOT1X-PACKET: EAPoL KEY packet received from 00:0d:54:98:99:6d

DOT1X Apr 11 20:45:38.007315 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.007354 DEBUG DOT1X: Received message 2 of 4way handshake from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.007558 DEBUG DOT1X: Sending message 3 of the 4way Handshake
DOT1X Apr 11 20:45:38.007586 DEBUG DOT1X-PACKET: EAPoL packet of 125 bytes (with retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.007613 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 5 seconds
DOT1X Apr 11 20:45:38.010168 DEBUG DOT1X-PACKET: EAPoL KEY packet received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.010195 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.010226 DEBUG DOT1X: Received message 4 of 4way handshake from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.010268 DEBUG DOT1X-PACKET: sending 32 byte multicast key with index 1 to AP
DOT1X Apr 11 20:45:38.010376 DEBUG DOT1X-PACKET: sending 32 byte unicast key with index 0 to AP

Once the exchange is done we send the resulting keys down to the AP.

DOT1X Apr 11 20:45:38.032664 DEBUG DOT1X: Sending message 1 of the Group Key Handshake

DOT1X Apr 11 20:45:38.032698 DEBUG DOT1X-PACKET: EAPoL packet of 131 bytes (without retransmit set) sent to 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.044877 DEBUG DOT1X-PACKET: EAPoL KEY packet received from 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.044903 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:0d:54:98:99:6d
DOT1X Apr 11 20:45:38.044933 DEBUG DOT1X: Received message 2 of group key handshake from 00:0d:54:98:99:6d

We then do the 2-way handshake to send the multicast group-key to the client.

Dot1x level 10 trace of dynamic WEP in pass-thru:

DOT1X Apr 11 20:33:04.695773 DEBUG DOT1X-CLIENT: new wireless client from 00:05:5d:88:d1:63 on port 2, radio 2

DOT1X Apr 11 20:33:04.699969 DEBUG DOT1X-STATE: 00:05:5d:88:d1:63 transition from NOTHING to CONNECTING
DOT1X Apr 11 20:33:04.703742 DEBUG DOT1X-STATS: 00:05:5d:88:d1:63, enters connecting --> 4371

DOT1X Apr 11 20:33:04.707674 DEBUG DOT1X-PACKET: setting id to networkid=nortelwlan,nasid=nos-3.0,portid=2 in request
DOT1X Apr 11 20:33:04.711374 DEBUG DOT1X-PACKET: EAPoL EAP packet of 54 bytes w/id 1 (with retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.715237 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 1 seconds
DOT1X Apr 11 20:33:04.783819 DEBUG DOT1X-PACKET:EAPoL START packet received from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.787403 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.791069 DEBUG DOT1X-STATE: 00:05:5d:88:d1:63 transition back to CONNECTING
DOT1X Apr 11 20:33:04.795066 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 1 seconds
DOT1X Apr 11 20:33:04.798553 DEBUG DOT1X-CLIENT: retransmit packet to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.817116 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 1, len 18, received from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.820757 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.824340 DEBUG DOT1X-CLIENT: glob 'NORTEL*' matches 'NORTEL\jtran', ssid 'nortelwlan' matches 'nortelwlan': eap_type=254

Like the previous trace, this is a listing of the network access rules which don't match (not shown in this example) or match.

DOT1X Apr 11 20:33:04.828032 DEBUG DOT1X-CLIENT: EAP-ID resp for NORTEL\jtran at 00:05:5d:88:d1:63 doing PASSTHRU

DOT1X Apr 11 20:33:04.833653 DEBUG DOT1X-CLIENT: eapol_aaa_login (sess=0x1ceef94) 00:05:5d:88:d1:63 -> AAA

These two messages indicate that the AAA subsystem is being invoked to authenticate the user. The subsequent log messages interleave the RADIUS conversation with the EAP conversation because in pass-through mode the WSS is pretty much just a translator between clients who speak EAP and servers who speak RADIUS.

DOT1X Apr 11 20:33:04.840747 DEBUG DOT1X-STATE: 00:05:5d:88:d1:63 transition from CONNECTING to AUTHENTICATING

DOT1X Apr 11 20:33:04.844308 DEBUG DOT1X-STATS: 00:05:5d:88:d1:63 enters authenticating --> 342
DOT1X Apr 11 20:33:04.848419 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 1, len 18, received from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.852028 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63

```
DOT1X Apr 11 20:33:04.855502 DEBUG DOT1X-TIMEOUT: Cancelling unset retrans timer
DOT1X Apr 11 20:33:04.859089 DEBUG 00:05:5d:88:d1:63 in AUTHENTICATING state, already
received identity
DOT1X Apr 11 20:33:04.878354 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status
STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:04.882083 DEBUG DOT1X-PACKET: EAPoL EAP packet of 10 bytes w/id 2 (with
retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.885976 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:33:04.913966 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 2, len 112, received
from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.917577 DEBUG DOT1X-PACKET: Cancelling retrans timer for
00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.922600 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94)
00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:04.938630 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status
STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:04.942345 DEBUG DOT1X-PACKET: EAPoL EAP packet of 136 bytes w/id 3
(with retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.946275 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:33:04.961459 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 3, len 53, received
from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.965135 DEBUG DOT1X-PACKET: Cancelling retrans timer for
00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.970242 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94)
00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:04.987167 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status
STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:04.990919 DEBUG DOT1X-PACKET: EAPoL EAP packet of 32 bytes w/id 5 (with
retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:04.994810 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:33:05.016260 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 5, len 41, received
from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.020140 DEBUG DOT1X-PACKET: Cancelling retrans timer for
00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.025113 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94)
00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:05.042391 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status
STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:05.046266 DEBUG DOT1X-PACKET: EAPoL EAP packet of 62 bytes w/id 6 (with
retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.050173 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 6 seconds
DOT1X Apr 11 20:33:05.059548 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 6, len 95, received
from 00:05:5d:88:d1:63
```

DOT1X Apr 11 20:33:05.063185 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.068243 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94) 00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:05.087828 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:05.091529 DEBUG DOT1X-PACKET: EAPoL EAP packet of 78 bytes w/id 7 (with retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.095414 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 6 seconds
DOT1X Apr 11 20:33:05.119408 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 7, len 29, received from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.123004 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.128006 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94) 00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:05.141861 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status STATUS:GETDATA from AAA
DOT1X Apr 11 20:33:05.145584 DEBUG DOT1X-PACKET: EAPoL EAP packet of 42 bytes w/id 8 (with retransmit set) sent to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.149491 DEBUG DOT1X-TIMEOUT: set when_retrans timer for 30 seconds
DOT1X Apr 11 20:33:05.158916 DEBUG DOT1X-PACKET: EAPoL EAP packet, id 8, len 38, received from 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.162580 DEBUG DOT1X-PACKET: Cancelling retrans timer for 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.167624 DEBUG DOT1X-CLIENT: eapol_aaa_continue (sess=0x1ceef94) 00:05:5d:88:d1:63 forwarded to AAA
DOT1X Apr 11 20:33:05.182130 DEBUG DOT1X-CLIENT: 00:05:5d:88:d1:63 status STATUS:PASS from AAA

This message indicates success from the Radius server. If you get a “FAIL from AAA” you should check the timestamp between this message and the previous one. If several seconds have elapsed either there is a connectivity problem to the radius server or the shared secret is wrong. If there is no real elapsed time, then the user was rejected by radius and you should check the Radius server logs.

DOT1X Apr 11 20:33:05.185751 DEBUG DOT1X-PACKET: EAPoL EAP packet of 8 bytes w/id 10 (without retransmit set) sent to 00:05:5d:88:d1:63

DOT1X Apr 11 20:33:05.189549 DEBUG DOT1X-STATE: 00:05:5d:88:d1:63 transition from AUTHENTICATING to AUTHENTICATED

DOT1X Apr 11 20:33:05.193894 DEBUG DOT1X: asked to change name NORTEL\jtran at 00:05:5d:88:d1:63 to NORTEL\jtran

This message is printed when the WSS updates the initial username (outer) with the final inner username. This is relevant to TTLS clients primarily.

DOT1X Apr 11 20:33:05.205114 DEBUG DOT1X-STATE: NORTEL\jtran at 00:05:5d:88:d1:63 is authorized

Passed authorization successfully.

DOT1X Apr 11 20:33:05.208927 DEBUG DOT1X-STATE: sending keys to 00:05:5d:88:d1:63

DOT1X Apr 11 20:33:05.212506 DEBUG DOT1X-STATE: Putting NORTEL\jtran at 00:05:5d:88:d1:63 in vlan vlan-eng (130)

Placing the client on the proper VLAN.

DOT1X Apr 11 20:33:05.216127 DEBUG DOT1X-STATE: NORTEL\jtran --> tag 3 for vlan 130, cipher 4, bssid 00:0b:0e:00:d5:83

This is information regarding the WSS to AP connection used for this user.

DOT1X Apr 11 20:33:05.220068 DEBUG setting (nth) client NORTEL\jtran rekey period to 9

The rekey period refers to broadcast key rolling. As each client is added, this value is set to match the next switch-wide rollover period.

DOT1X Apr 11 20:33:05.223596 DEBUG DOT1X-PACKET: sending 13 byte multicast key with index 3 to AP

DOT1X Apr 11 20:33:05.227310 DEBUG DOT1X-PACKET: sending 13 byte unicast key with index 0 to AP

We send the keys down to the AP.

DOT1X Apr 11 20:33:05.235460 DEBUG DOT1X-PACKET: sending group key to 00:05:5d:88:d1:63

DOT1X Apr 11 20:33:05.239054 DEBUG DOT1X-PACKET: EAPoL packet of 61 bytes (without retransmit set) sent to 00:05:5d:88:d1:63

DOT1X Apr 11 20:33:05.243420 DEBUG DOT1X-PACKET: sending empty eapol keymsg to 00:05:5d:88:d1:63
DOT1X Apr 11 20:33:05.247019 DEBUG DOT1X-PACKET: EAPoL packet of 48 bytes (without retransmit set) sent to 00:05:5d:88:d1:63

We send key packets to the client.

DOT1X Apr 11 20:33:05.251025 DEBUG DOT1X: Session timeout for 00:05:5d:88:d1:63 set to 3600

DOT1X Apr 11 20:33:05.252763 DEBUG DOT1X-TIMEOUT: set when_reauth timer for 3600 seconds

And we set the re-authentication timer for this user (because he's WEP we need to reauthenticate in order to cycle the key.

RADIUS level 10 trace of 802.1X pass-thru authentication

Radius tracing will displays decodes of all packets sent to or from a RADIUS server, as well as debug information from the RADIUS process within NOS. Most of the information in the trace is very repetitive as all RADIUS attributes are sent back-and-forth for each RADIUS packet in an 802.1X exchange. The most frequently useful portion of the exchange is the very end where the RADIUS server send an "ACCEPT" or "REJECT" packet. The "ACCEPT" includes all of the RADIUS attributes configured for that user, many of which are used by MSS to authorize the user.

AAA Jan 31 22:44:46.696276 DEBUG (1872) RADIUS: Set srv to sg1/W2k3 (192.168.3.4/1812/1813)

Select the RADIUS server to use with this authentication. In this case it is "W2k3" and is part of the server group "sg1".

AAA Jan 31 22:44:46.696357 DEBUG (1872) RADIUS: set_rad_ident ident=196 local port=20003

Setup an internal identifier and open a local high-port for transmission of RADIUS packets.

AAA Jan 31 22:44:46.696419 DEBUG (1872) RADIUS: session EAP_LOGIN

AAA Jan 31 22:44:46.696479 DEBUG (1872) RADIUS: AAA_SESS_TYPE_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.696536 DEBUG (1872) RADIUS: AAA_STATUS_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.696594 DEBUG (1872) RADIUS: AAA_SENDER_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.696651 DEBUG (1872) RADIUS: AAA_AUTHEN_METHOD_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.696708 DEBUG (1872) RADIUS: AAA_NAS_PORT_ID: len=4
AAA Jan 31 22:44:46.696811 DEBUG (1872) RADIUS: Added IETF 87 RAD_NAS_PORT_ID vlen=3 1/2
AAA Jan 31 22:44:46.696881 DEBUG (1872) RADIUS: AAA_CALLING_STATION_ID: len=18
AAA Jan 31 22:44:46.696966 DEBUG (1872) RADIUS: Added IETF 31 RAD_CALLING_STATION_ID vlen=17 00-0B-7D-1F-FB-F5
AAA Jan 31 22:44:46.697032 DEBUG (1872) RADIUS: AAA_CALLED_STATION_ID: call rad_enc_called_station_id

AAA Jan 31 22:44:46.697165 DEBUG (1872) RADIUS: Added IETF 30 RAD_CALLED_STATION_ID vlen=29 00-0B-0E-14-E9-80:nortelwlan
AAA Jan 31 22:44:46.697238 DEBUG (1872) RADIUS: AAA_SVC_ATTR: len=4
AAA Jan 31 22:44:46.697319 DEBUG (1872) RADIUS: Added IETF 6 RAD_ATTR_SERVICE vlen=4 2
AAA Jan 31 22:44:46.697385 DEBUG (1872) RADIUS: AAA_SSID_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.697444 DEBUG (1872) RADIUS: AAA_MACADDR_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.697499 DEBUG (1872) RADIUS: AAA_COOKIE_ATTR: ignored (invalid sess type)
AAA Jan 31 22:44:46.697555 DEBUG (1872) RADIUS: AAA_EAP_MSG_ATTR: call rad_enc_eap_msg
AAA Jan 31 22:44:46.697635 DEBUG (1872) RADIUS: Added IETF 79 RAD_ATTR_EAP_MSG vlen=18 0x2010012....
AAA Jan 31 22:44:46.697698 DEBUG (1872) RADIUS: AAA_USERNAME_ATTR: call rad_enc_username
AAA Jan 31 22:44:46.697807 DEBUG (1872) RADIUS: Added IETF 1 RAD_ATTR_USER vlen=13 NORTEL\tash
AAA Jan 31 22:44:46.697902 DEBUG (1872) RADIUS: Added IETF 61 RAD_ATTR_NAS_PORT_TYPE vlen=4 19
AAA Jan 31 22:44:46.697994 DEBUG (1872) RADIUS: Added IETF 32 RAD_ATTR_NAS_IDENTIFIER vlen=7 Nortel
AAA Jan 31 22:44:46.698114 DEBUG (1872) RADIUS: Added IETF 4 RAD_ATTR_NAS_IP_ADDRESS vlen=4 192.168.12.7
AAA Jan 31 22:44:46.698257 DEBUG (1872) RADIUS: Added IETF 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0x0....

Determine which RADIUS attributes are appropriate for this type of connection and build RADIUS Request packet.

AAA Jan 31 22:44:46.698413 DEBUG (1872) RADIUS: W2k3 XMIT <196,20003,192.168.3.4>:1812, ACCESS_REQUEST, len 155

Transmit the RADIUS Access Request packet to the server. The packet is 155 bytes in length.

AAA Jan 31 22:44:46.698585 DEBUG (0) RADIUS: Authenticator=0x4d 41 a1 ee 10 5c a6 8f 53 cc ad 1c 0a 8c 6d 25

AAA Jan 31 22:44:46.698670 DEBUG (1872) RADIUS: 87 RAD_NAS_PORT_ID vlen=3 1/2
AAA Jan 31 22:44:46.698752 DEBUG (1872) RADIUS: 31 RAD_CALLING_STATION_ID vlen=17 00-0B-7D-1F-FB-F5
AAA Jan 31 22:44:46.698857 DEBUG (1872) RADIUS: 30 RAD_CALLED_STATION_ID vlen=29 00-0B-0E-14-E9-80:nortelwlan
AAA Jan 31 22:44:46.698934 DEBUG (1872) RADIUS: 6 RAD_ATTR_SERVICE vlen=4 2

AAA Jan 31 22:44:46.699007 DEBUG (1872) RADIUS: 79 RAD_ATTR_EAP_MSG vlen=18 0x2010012....
AAA Jan 31 22:44:46.699084 DEBUG (1872) RADIUS: 1 RAD_ATTR_USER vlen=13 NORTEL\tash
AAA Jan 31 22:44:46.699157 DEBUG (1872) RADIUS: 61 RAD_ATTR_NAS_PORT_TYPE vlen=4 19
AAA Jan 31 22:44:46.699230 DEBUG (1872) RADIUS: 32 RAD_ATTR_NAS_IDENTIFIER vlen=7 Nortel
AAA Jan 31 22:44:46.699311 DEBUG (1872) RADIUS: 4 RAD_ATTR_NAS_IP_ADDRESS vlen=4 192.168.12.7
AAA Jan 31 22:44:46.699386 DEBUG (1872) RADIUS: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0x9887a2e8....

Decode of packet.

AAA Jan 31 22:44:46.699518 DEBUG (1872) RADIUS: Set timer handle 1208220736 duration 5

Set RADIUS timeout for this packet – 5 seconds in this case.

AAA Jan 31 22:44:46.699629 DEBUG (1872) RADIUS: local ip addr is 192.168.12.7

IP interface packet is sent from.

AAA Jan 31 22:44:46.708717 DEBUG (1872) RADIUS: REPLY <196,1812,192.168.3.4>:20003, ACCESS_CHALLENGE, len 76

WSS receives RADIUS Challenge packet from RADIUS server. The packet is 76 bytes in length.

AAA Jan 31 22:44:46.708919 DEBUG (0) RADIUS: Authenticator=0xb5 61 ad 8d 69 54 7b c4 6b c3 6b 18 89 68 f9 b1

AAA Jan 31 22:44:46.709004 DEBUG (1872) RADIUS: 27 RAD_ATTR_SESSION_TIMEOUT vlen=4 30
AAA Jan 31 22:44:46.709080 DEBUG (1872) RADIUS: 79 RAD_ATTR_EAP_MSG vlen=6 0x1020006....
AAA Jan 31 22:44:46.709157 DEBUG (1872) RADIUS: 24 RAD_ATTR_STATE vlen=22 0x16d2034a....
AAA Jan 31 22:44:46.709234 DEBUG (1872) RADIUS: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0xf16a543b....
AAA Jan 31 22:44:46.709413 DEBUG (1872) RADIUS: Input: 27 RAD_ATTR_SESSION_TIMEOUT vlen=4 30
AAA Jan 31 22:44:46.709503 DEBUG (1872) RADIUS: Update AAA_SESSION_TIMEOUT_ATTR len=4 val=0x1e

AAA Jan 31 22:44:46.709581 DEBUG (1872) RADIUS: Input: 79 RAD_ATTR_EAP_MSG vlen=6 0x1020006....
AAA Jan 31 22:44:46.709640 DEBUG (1872) RADIUS: Call decode fn rad_dec_no_op
AAA Jan 31 22:44:46.709694 DEBUG (1872) RADIUS: rad_dec_no_op RAD_ATTR_EAP_MSG ignored
AAA Jan 31 22:44:46.709768 DEBUG (1872) RADIUS: Input: 24 RAD_ATTR_STATE vlen=22 0x16d2034a....
AAA Jan 31 22:44:46.709875 DEBUG (1872) RADIUS: Update binary AAA_RAD_STATE_ATTR len=22
AAA Jan 31 22:44:46.709962 DEBUG (1872) RADIUS: Input: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0xf16a543b....
AAA Jan 31 22:44:46.710021 DEBUG (1872) RADIUS: Call decode fn rad_dec_no_op
AAA Jan 31 22:44:46.710074 DEBUG (1872) RADIUS: rad_dec_no_op RAD_ATTR_EAP_MSG_AUTHENTICATOR ignored

Decode packet and call appropriate functions to relay EAP payload back to dot1x.

AAA Jan 31 22:44:46.717308 DEBUG (1872) RADIUS: Force preferred dot1x srv sg1/W2k3 (192.168.3.4/1812/1813)

Force the WSS to use the same RADIUS server for all packets subsequent to the first.

AAA Jan 31 22:44:46.717403 DEBUG (1872) RADIUS: Set srv to sg1/W2k3 (192.168.3.4/1812/1813)

AAA Jan 31 22:44:46.717464 DEBUG (1872) RADIUS: set_rad_ident ident=197 local port=20003

AAA Jan 31 22:44:46.717522 DEBUG (1872) RADIUS: session EAP_LOGIN
AAA Jan 31 22:44:46.717579 DEBUG (1872) RADIUS: AAA_SESS_TYPE_ATTR: ignored (invalid sess type)

…<deleted for brevity – several pages of similar decodes for each packet in an EAP exchange>…

AAA Jan 31 22:44:46.865712 DEBUG (1872) RADIUS: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0xb17f1471....

AAA Jan 31 22:44:46.865845 DEBUG (1872) RADIUS: Set timer handle 1208220736 duration 5
AAA Jan 31 22:44:46.865957 DEBUG (1872) RADIUS: local ip addr is 192.168.12.7

Repeat the same process for all packets in the EAP exchange.

AAA Jan 31 22:44:46.868037 DEBUG (1872) RADIUS: REPLY <203,1812,192.168.3.4>:20003, ACCESS_ACCEPT, len 268

Receive an "ACCESS ACCEPT" packet from the RADIUS server. This is where it gets interesting.

AAA Jan 31 22:44:46.868218 DEBUG (0) RADIUS: Authenticator=0x33 b6 52 70 93 ec 63 67 6b 78 13 c6 48 c5 d7 e8

AAA Jan 31 22:44:46.868299 DEBUG (1872) RADIUS: 79 RAD_ATTR_EAP_MSG vlen=4 0x30b0004....
AAA Jan 31 22:44:46.868379 DEBUG (1872) RADIUS: 81 RAD_TUNNEL_PRIVATE_GROUP_ID vlen=8 vlan-eng
AAA Jan 31 22:44:46.868457 DEBUG (1872) RADIUS: VSA=10, len=15, MICROSOFT RAD_ATTR_UNKNOWN vlen 7 0x1545241....
AAA Jan 31 22:44:46.868562 DEBUG (1872) RADIUS: VSA=26, len=51, MICROSOFT MS-CHAP-V2-Success vlen 43 0x1533d35....
AAA Jan 31 22:44:46.868624 DEBUG (1872) RADIUS: VSA=16, len=58, MICROSOFT MS-MPPE-Send-Key vlen 52
AAA Jan 31 22:44:46.868682 DEBUG (1872) RADIUS: VSA=17, len=58, MICROSOFT MS-MPPE-Recv-Key vlen 52
AAA Jan 31 22:44:46.868760 DEBUG (1872) RADIUS: 25 RAD_ATTR_CLASS vlen=30 0x450504eb....
AAA Jan 31 22:44:46.868837 DEBUG (1872) RADIUS: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0x22620286....

Decode of packet received.

AAA Jan 31 22:44:46.869027 DEBUG (1872) RADIUS: Input: 79 RAD_ATTR_EAP_MSG vlen=4 0x30b0004....

AAA Jan 31 22:44:46.869102 DEBUG (1872) RADIUS: Call decode fn rad_dec_no_op
AAA Jan 31 22:44:46.869156 DEBUG (1872) RADIUS: rad_dec_no_op RAD_ATTR_EAP_MSG ignored
AAA Jan 31 22:44:46.869231 DEBUG (1872) RADIUS: Input: 81 RAD_TUNNEL_PRIVATE_GROUP_ID vlen=8 vlan-eng
AAA Jan 31 22:44:46.869288 DEBUG (1872) RADIUS: Call decode fn rad_dec_tunnel_private_group_id
AAA Jan 31 22:44:46.869364 DEBUG (1872) RADIUS: (rad_dec_tunnel_private_group_id) RAD_TUNNEL_PRIVATE_GROUP_ID vlan-eng

Decode the "Tunnel-Private-Group-ID" value and pass it to authorization for use as the user's VLAN. In this case the VLAN is "vlan-eng". If you have configured other attributes on the RADIUS server they will appear in this list. All Nortel authorization attributes are of type "string", so you will be able to read the contents clearly in the trace. This is very useful when troubleshooting "authorization errors" on the WSS as you can see what the WSS thinks the RADIUS server is sending. Keep in mind that this is from the perspective of the WSS, and so is not a complete replacement for an Ethernet sniffer trace… But it's a very good starting point.

AAA Jan 31 22:44:46.869455 DEBUG (1872) RADIUS: Input: VSA=10, len=15, MICROSOFT RAD_ATTR_UNKNOWN vlen 7 0x1545241....

AAA Jan 31 22:44:46.869568 DEBUG (1872) RADIUS: No mapping for VSA=10, len=15, MICROSOFT RAD_ATTR_UNKNOWN vlen 7 0x1545241....
AAA Jan 31 22:44:46.869653 DEBUG (1872) RADIUS: Input: VSA=26, len=51, MICROSOFT MS-CHAP-V2-Success vlen 43 0x1533d35....
AAA Jan 31 22:44:46.869742 DEBUG (1872) RADIUS: Update binary AAA_MSCHAP_V2_SUCCESS_ATTR len=43
AAA Jan 31 22:44:46.869809 DEBUG (1872) RADIUS: Input: VSA=16, len=58, MICROSOFT MS-MPPE-Send-Key vlen 52
AAA Jan 31 22:44:46.869867 DEBUG (1872) RADIUS: Call rad_dec_mppe_key
AAA Jan 31 22:44:46.869995 DEBUG (1872) RADIUS: Update AAA_MS_MPPE_SENDKEY_ATTR len=32
AAA Jan 31 22:44:46.870066 DEBUG (1872) RADIUS: Input: VSA=17, len=58, MICROSOFT MS-MPPE-Recv-Key vlen 52
AAA Jan 31 22:44:46.870126 DEBUG (1872) RADIUS: Call rad_dec_mppe_key
AAA Jan 31 22:44:46.870247 DEBUG (1872) RADIUS: Update AAA_MS_MPPE_RECVKEY_ATTR len=32

The MS_MPPE Send and Receive key are sent to the dot1x process for use as keying material for encryption.

AAA Jan 31 22:44:46.870341 DEBUG (1872) RADIUS: Input: 25 RAD_ATTR_CLASS vlen=30 0x450504eb....

AAA Jan 31 22:44:46.870398 DEBUG (1872) RADIUS: Call decode fn rad_dec_class
AAA Jan 31 22:44:46.870472 DEBUG (1872) RADIUS: (1872) rad_dec_class RAD_ATTR_CLASS set into smdb
AAA Jan 31 22:44:46.870586 DEBUG (1872) RADIUS: Input: 80 RAD_ATTR_EAP_MSG_AUTHENTICATOR vlen=16 0x22620286....
AAA Jan 31 22:44:46.870647 DEBUG (1872) RADIUS: Call decode fn rad_dec_no_op
AAA Jan 31 22:44:46.870699 DEBUG (1872) RADIUS: rad_dec_no_op RAD_ATTR_EAP_MSG_AUTHENTICATOR ignored

SM level 10 trace of client connecting

SM Jan 4 02:52:41.183936 DEBUG SM-DOT11: assoc req from 00:10:c6:5d:ae:ae on port 2049

System has received an 802.11 association request from a wireless client with the MAC address “00:10:c6:5d:ae:ae” from DAP 1 (DAP num = port num – 2048 if port num is greater than 2048)

SM Jan 4 02:52:41.184020 DEBUG SM-DOT11: 00:10:c6:5d:ae:ae requests association to [slipshod]

Client connected to the SSID “slipshod”

SM Jan 4 02:52:41.184124 DEBUG SM-DOT11: sending associate response 0 to 00:0b:0e:2f:6d:00 for client 00:10:c6:5d:ae:ae

Responding to association request…

SM Jan 4 02:52:41.184448 DEBUG 20 04 00 01 00 01 00 02 00 2c 00 00 02 01 00 0b

SM Jan 4 02:52:41.184532 DEBUG 0e 2f 6d 02 00 10 c6 5d ae ae 00 0b 0e 2f 6d 02

SM Jan 4 02:52:41.184582 DEBUG 00 00 02 01 00 0b 0e 2f 6d 02 00 10 c6 5d ae ae
SM Jan 4 02:52:41.184632 DEBUG 00 0b 0e 2f 6d 02 d0 1d 11 00 01 00 00 08 73 6c

Hex dump of association response.

SM Jan 4 02body:52:41.184700 DEBUG SM-DOT11: this client is new to us

Setting up new session manager entry for this client

SM Jan 4 02:52:41.186274 DEBUG SMDB: (2) setting radio device id=2, slot=1

The number in parenthesis is the local session-id which is displayed in the output of "show sessions network". This request also came from the 802.11g radio "slot=1" means radio 1, which is the 802.11g radio in this AP. Pay close attention to the session-id when reading traces as it can be confusing when a client roams and there is an older session ID being torn down while a newer session-id is coming up.

SM Jan 4 02:52:41.186337 DEBUG SM-TRACE: (re)associate request from device 2

SM Jan 4 02:52:41.186454 DEBUG SM: (2) inserting IP 0.0.0.0

Inserting default value for snooped IP address into table.

SM Jan 4 02:52:41.186561 DEBUG SM-TRACE: state for 00:10:c6:5d:ae:ae --> INITIALIZING

SM Jan 4 02:52:41.186639 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 0h, to change state unk state 0 -> INITIALIZING, by i_smdb_create

Entering the "INITIALIZING" state of the SM state machine.

SM Jan 4 02:52:41.186767 DEBUG SM-DOT11: (2) client 00:10:c6:5d:ae:ae associated to crypto ssid, slipshod

SM Jan 4 02:52:41.186922 DEBUG SMDB: (2) i_smdb_set_service_prof: setting service prof "slipshod"
SM Jan 4 02:52:41.187017 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 0h, to change state INITIALIZING -> AUTH,ASSOC REQ, by sm_dot11_handle_associate

Transitioning states within SM.

SM Jan 4 02:52:41.187131 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 0h, to change state AUTH,ASSOC REQ -> AUTH AND ASSOC, by sm_dot11_handle_associate

SM Jan 4 02:52:41.187205 DEBUG SM-EVENT: (2) sending net/dot1x/eapol/associate to fsm net/dot1x/eapol

Informing the 802.1X state machine of the new client.

SM Jan 4 02:52:41.187291 DEBUG SM-TRACE: (2) added proc hist @484c13ac (3 by sm_dot11_handle_associate); 1 total

SM Jan 4 02:52:41.187353 DEBUG SM-STATE: (2) sm_dot11_handle_associate bumps kill lock vector to 2h

Adding a process hook and setting a lock.

SM Jan 4 02:52:41.187398 DEBUG SM-EVENT: (2) incrementing loadbal session on port 2049

Make a note that we have another session on DAP 1 for use with AP load balancing.

SM Jan 4 02:52:41.188453 DEBUG SM-TRACE: (2) added proc hist @484c132c (3 by wifi_association); 2 total

SM Jan 4 02:52:41.188529 DEBUG SM-STATE: (2) wifi_association bumps kill lock vector to ah
SM Jan 4 02:52:41.188571 DEBUG SM-ROAM: (2) wifi_association bumps roam refcount to 1

More process hooks and locks.

SM Jan 4 02:52:43.203223 DEBUG SM: (2) 00:10:c6:5d:ae:ae i_smdb_set_ingress_filter NULL by set_smdb_from_author_attrs

Checking to see if there is a User-based ACL (filter-id radius attribute) defined for this client.

SM Jan 4 02:52:43.203320 DEBUG SM: (2) 00:10:c6:5d:ae:ae i_smdb_set_vlan_name=NULL by set_smdb_from_author_attrs

Checking to see if there is a VLAN associated with this client.

SM Jan 4 02:52:43.203711 DEBUG smdb_insert_vlan: store ("default"+cruft, tot 18): 0

The client belongs on VLAN “default”. Insert this into the session entry.

SM Jan 4 02:52:43.203781 DEBUG SM: (2) 00:10:c6:5d:ae:ae i_smdb_set_vlan_name=default by set_smdb_from_author_attrs

Client session is configured for vlan default.

SM Jan 4 02:52:43.204003 DEBUG SM-TRACE: (2) added proc hist @484c11ac (3 by do_vlan); 3 total

SM Jan 4 02:52:43.204068 DEBUG SM-STATE: (2) do_vlan bumps kill lock vector to 1ah

More process hooks and locks.

SM Jan 4 02:52:43.204420 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 20020ch, to change state AUTH AND ASSOC -> AUTHORIZING, by aaa_dot1x_process_author_data

Another state change.

SM Jan 4 02:52:43.204512 DEBUG SM-EVENT: (2) sending net/dot1x/eapol/authorizing to fsm net/dot1x/eapol

Informing 802.1X state machine that the client's authorization is being processed.

SM Jan 4 02:52:43.206177 DEBUG SM-STATE: (2) setting tag to 1

Since this was the first client using this VLAN on the AP we need to extend the VLAN the client is configured for to the AP. We use 802.1q tags to indicate each radio/vlan combination, and tunnel the tagged packets inside of TAPA. These tags are created dynamically as needed, and can be re-used if additional clients are on the same radio and VLAN. You can see the tags created in the output of the "show vlans" command from the CLI by looking for tags on AP ports, or on DAP ports.

SM Jan 4 02:52:43.206774 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 20022ch, to change state AUTHORIZING -> AUTHORIZED, by eapol_set_authorized

Moving through the state machine…

SM Jan 4 02:52:43.206872 DEBUG SM-EVENT: (2) sending sm/authorized to fsm sm central fsm

Notifying the SM state machine that the client is authorized.

SM Jan 4 02:52:43.206975 DEBUG i_smdb_publish_ident_by_mac: (2) publishing; login type unknown aaa_printable_sess_type 1211039744

SM Jan 4 02:52:43.207142 DEBUG i_smdb_publish_ident_by_mac: (2) cluster_store returned 0

Populating the client's information in the cluster database. This is information is pushed out all of the WSSs in the mobility domain to assist inter-WSS roaming and preservation of accounting data across roams.

SM Jan 4 02:52:43.208429 DEBUG SM-NOTIFY: vlan_if_find_by_name("default") -> vlanp w/ name="vlan:1", num=1
SM Jan 4 02:52:43.208778 DEBUG SM-STATE: (2) setting mapping tag to 1
SM Jan 4 02:52:43.208840 DEBUG SM-EVENT: (2) added vport, i/f "2049", vlan "default", tag 1(1)

Populating the FDB with the client's entry on the proper VLAN, and finalizing the VLAN tunnel to the AP.

SM Jan 4 02:52:43.208987 DEBUG SM-EVENT: (2) enabled forwarding for 00:10:c6:5d:ae:ae, going ACTIVE

All of the low-level FDB work is done, and the switch is now forwarding traffic to/from the client.

SM Jan 4 02:52:43.209079 DEBUG SM-STATE: (2) mac 00:10:c6:5d:ae:ae, flags 28832ch, to change state AUTHORIZED -> ACTIVE, by sm_handle_authorized

SM Jan 4 02:52:43.209151 DEBUG SM-EVENT: (2) sending sm/active to fsm sm central fsm

Changing state again, and more notifications to the SM state machine.

SM Jan 4 02:52:43.209199 DEBUG SM-EVENT: (2) sending aaa/sm/notify to fsm net/igmp

Notify IGMP that there is a new client.

SM Jan 4 02:52:44.208859 DEBUG SM-ROAM: got RE_ASSO_CONF for localid 2, mac 00:10:c6:5d:ae:ae, status=CLUSTER

SM Jan 4 02:52:44.208968 DEBUG SM-TRACE: (2) added proc hist @484c10ac (4 by sm_handle_move_conf); 4 total
SM Jan 4 02:52:44.209029 DEBUG SM-STATE: (2) sm_handle_move_conf clearing kill lock, lock vector now =ah
SM Jan 4 02:52:44.209085 DEBUG SM-STATE: (2) clear lock 4 for sm_handle_move_conf (now ah) but state =ACTIVE, not KILLING
SM Jan 4 02:52:44.209137 DEBUG SM-ROAM: (2) got conf for sess in state ACTIVE
SM Jan 4 02:52:44.209176 DEBUG SM-EVENT: (2) send SM_AAA_SESS_START to AAA

SM Jan 4 02:52:44.209672 DEBUG SM-TRACE: (2) added proc hist @4855ac2c (3 by AAA new session); 5 total

SM Jan 4 02:52:44.209740 DEBUG SM-STATE: (2) AAA new session bumps kill lock vector to eh

Developer debug messages… Probably receiving confirmation back from the cluster database on the request to update with the client's identity/location.

SM Jan 4 02:52:49.160478 DEBUG SM-DOT11: assoc req from 00:0b:7d:26:9d:d7 on port 2049

Here comes another client… The only comments on this one will be with different messages.

SM Jan 4 02:52:49.160561 DEBUG SM-DOT11: 00:0b:7d:26:9d:d7 requests association to [slipshod]

SM Jan 4 02:52:49.160617 DEBUG SM-DOT11: Association from 00:0b:7d:26:9d:d7 found a WPA-Elem of 6 bytes
SM Jan 4 02:52:49.160672 DEBUG SM-DOT11: Association from 00:0b:7d:26:9d:d7 found a WPA-Elem of 6 bytes
SM Jan 4 02:52:49.160726 DEBUG SM-DOT11: Association from 00:0b:7d:26:9d:d7 found a WPA-Elem of 24 bytes
SM Jan 4 02:52:49.160780 DEBUG SM-DOT11: Association from 00:0b:7d:26:9d:d7 with valid WPA-Elem of 24 bytes

This client is configured for WPA, and we have to parse the WPA information elements from the association request. For WPA2 clients this would be used for 802.11i fast roaming.

SM Jan 4 02:52:49.160915 DEBUG SM-DOT11: sending associate response 0 to 00:0b:0e:2f:6d:00 for client 00:0b:7d:26:9d:d7

SM Jan 4 02:52:49.161256 DEBUG 20 04 00 01 00 02 00 03 00 5a 00 00 3c 00 00 0b
SM Jan 4 02:52:49.161340 DEBUG 0e 2f 6d 03 00 0b 7d 26 9d d7 00 0b 0e 2f 6d 03
SM Jan 4 02:52:49.161390 DEBUG 00 00 3c 00 00 0b 0e 2f 6d 03 00 0b 7d 26 9d d7
SM Jan 4 02:52:49.161440 DEBUG 00 0b 0e 2f 6d 03 20 03 11 00 0a 00 00 08 73 6c
SM Jan 4 02:52:49.161508 DEBUG SM-DOT11: this client is new to us
SM Jan 4 02:52:49.163086 DEBUG SMDB: (3) setting radio device id=3, slot=2
SM Jan 4 02:52:49.163150 DEBUG SM-TRACE: (re)associate request from device 3
SM Jan 4 02:52:49.163267 DEBUG SM: (3) inserting IP 0.0.0.0
SM Jan 4 02:52:49.163370 DEBUG SM-TRACE: state for 00:0b:7d:26:9d:d7 --> INITIALIZING
SM Jan 4 02:52:49.163448 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 0h, to change state unk state 0 -> INITIALIZING, by i_smdb_create
SM Jan 4 02:52:49.163574 DEBUG SM-DOT11: (3) client 00:0b:7d:26:9d:d7 associated to crypto ssid, slipshod

SM Jan 4 02:52:49.163646 DEBUG SM-DOT11: (3) i_smdb_set_rsn_ie: here's the hex:
SM Jan 4 02:52:49.163688 DEBUG dd 18 00 50 f2 01 01 00 00 50 f2 05 01 00 00 50
SM Jan 4 02:52:49.163743 DEBUG f2 02 01 00 00 50 f2 01 00 00 ee ee ee ee ee ee

Decodes from some of the ealier information elements in the association request.

SM Jan 4 02:52:49.163868 DEBUG SMDB: (3) i_smdb_set_service_prof: setting service prof "slipshod"

SM Jan 4 02:52:49.164057 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 0h, to change state INITIALIZING -> AUTH,ASSOC REQ, by sm_dot11_handle_associate
SM Jan 4 02:52:49.164173 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 0h, to change state AUTH,ASSOC REQ -> AUTH AND ASSOC, by sm_dot11_handle_associate
SM Jan 4 02:52:49.164245 DEBUG SM-EVENT: (3) sending net/dot1x/eapol/associate to fsm net/dot1x/eapol

SM Jan 4 02:52:49.164330 DEBUG SM-TRACE: (3) added proc hist @4855ad2c (3 by sm_dot11_handle_associate); 1 total

SM Jan 4 02:52:49.164390 DEBUG SM-STATE: (3) sm_dot11_handle_associate bumps kill lock vector to 2h

SM Jan 4 02:52:49.164435 DEBUG SM-EVENT: (3) incrementing loadbal session on port 2049

SM Jan 4 02:52:49.164663 DEBUG SM-TRACE: (3) added proc hist @4855abac (3 by wifi_association); 2 total

SM Jan 4 02:52:49.164727 DEBUG SM-STATE: (3) wifi_association bumps kill lock vector to ah

SM Jan 4 02:52:49.164770 DEBUG SM-ROAM: (3) wifi_association bumps roam refcount to 1

SM Jan 4 02:52:49.469541 DEBUG SM_STATE: localid 3, setting recv key of 32 bytes

SM Jan 4 02:52:49.469618 DEBUG SM_STATE: localid 3, setting send key of 32 bytes

Generating the encryption keys. This did not occur with the previous authentication because this one is WPA/TKIP while the previous one was static WEP.

SM Jan 4 02:52:49.470703 DEBUG SM: (3) 00:0b:7d:26:9d:d7 i_smdb_set_ingress_filter NULL by set_smdb_from_author_attrs

SM Jan 4 02:52:49.470799 DEBUG SM: (3) 00:0b:7d:26:9d:d7 i_smdb_set_vlan_name=NULL by set_smdb_from_author_attrs
SM Jan 4 02:52:49.471237 DEBUG smdb_insert_vlan: store ("default"+cruft, tot 18): 0

SM Jan 4 02:52:49.471311 DEBUG SM: (3) 00:0b:7d:26:9d:d7 i_smdb_set_vlan_name=default by set_smdb_from_author_attrs

SM Jan 4 02:52:49.471515 DEBUG SM-TRACE: (3) added proc hist @4855a8ac (3 by do_vlan); 3 total

SM Jan 4 02:52:49.471577 DEBUG SM-STATE: (3) do_vlan bumps kill lock vector to 1ah

SM Jan 4 02:52:49.471928 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 20000ch, to change state AUTH AND ASSOC -> AUTHORIZING, by aaa_dot1x_process_author_data

SM Jan 4 02:52:49.472017 DEBUG SM-EVENT: (3) sending net/dot1x/eapol/authorizing to fsm net/dot1x/eapol

SM Jan 4 02:52:49.473685 DEBUG SM-STATE: (3) setting tag to 2

Notice how tag 2 is now being used, even though both devices are on the same vlan. This is because the VLAN needs to be tunneled through TAPA separately for each radio. Tunnels are limited to only the radios that require them, preventing extraneous broadcast traffic over the air.

SM Jan 4 02:52:49.526158 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 20002ch, to change state AUTHORIZING -> AUTHORIZED, by eapol_set_authorized

SM Jan 4 02:52:49.526269 DEBUG SM-EVENT: (3) sending sm/authorized to fsm sm central fsm

SM Jan 4 02:52:49.526352 DEBUG i_smdb_publish_ident_by_mac: (3) publishing; login type unknown aaa_printable_sess_type 1211039744

SM Jan 4 02:52:49.526511 DEBUG i_smdb_publish_ident_by_mac: (3) cluster_store returned 0

SM Jan 4 02:52:49.527754 DEBUG SM-NOTIFY: vlan_if_find_by_name("default") -> vlanp w/ name="vlan:1", num=1

SM Jan 4 02:52:49.528091 DEBUG SM-STATE: (3) setting mapping tag to 2

SM Jan 4 02:52:49.528150 DEBUG SM-EVENT: (3) added vport, i/f "2049", vlan "default", tag 2(2)

SM Jan 4 02:52:49.528297 DEBUG SM-EVENT: (3) enabled forwarding for 00:0b:7d:26:9d:d7, going ACTIVE

SM Jan 4 02:52:49.528389 DEBUG SM-STATE: (3) mac 00:0b:7d:26:9d:d7, flags 28812ch, to change state AUTHORIZED -> ACTIVE, by sm_handle_authorized

SM Jan 4 02:52:49.528460 DEBUG SM-EVENT: (3) sending sm/active to fsm sm central fsm

SM Jan 4 02:52:49.528508 DEBUG SM-EVENT: (3) sending aaa/sm/notify to fsm net/igmp

SM Jan 4 02:52:49.603848 DEBUG SM-EVENT: (3) rssi -74, rate 108, idle 0 secs

SM Jan 4 02:52:49.603937 DEBUG SM-EVENT: (3) idle timer 0 left, reset to 180000 ms

These two messages are related to the idle-timeout function built-in to the AP. If a client has not transmitted data recently, the AP will send a null-data packet to the client and wait for an 802.11 ACK. If an ACK is received the timer is refreshed. If no ACK is received we continue checking for the client and will timeout the session if we do not receive a response.

SM Jan 4 02:52:49.604972 DEBUG SM-EVENT: (2) rssi -56, rate 22, idle 0 secs

SM Jan 4 02:52:49.605063 DEBUG SM-EVENT: (2) idle timer 0 left, reset to 180000 ms

We're checking the previous authentication now – session-id 2.

SM Jan 4 02:52:50.483075 DEBUG SM-ROAM: got RE_ASSO_CONF for localid 3, mac 00:0b:7d:26:9d:d7, status=CLUSTER

SM Jan 4 02:52:50.483184 DEBUG SM-TRACE: (3) added proc hist @4855a82c (4 by sm_handle_move_conf); 4 total

SM Jan 4 02:52:50.483244 DEBUG SM-STATE: (3) sm_handle_move_conf clearing kill lock, lock vector now =ah

SM Jan 4 02:52:50.483298 DEBUG SM-STATE: (3) clear lock 4 for sm_handle_move_conf (now ah) but state =ACTIVE, not KILLING

SM Jan 4 02:52:50.483350 DEBUG SM-ROAM: (3) got conf for sess in state ACTIVE

SM Jan 4 02:52:50.483390 DEBUG SM-EVENT: (3) send SM_AAA_SESS_START to AAA

SM Jan 4 02:52:50.483893 DEBUG SM-TRACE: (3) added proc hist @4855a42c (3 by AAA new session); 5 total

SM Jan 4 02:52:50.483959 DEBUG SM-STATE: (3) AAA new session bumps kill lock vector to eh
SM Jan 4 02:52:50.765171 DEBUG SM: (3) removing IP 0.0.0.0

SM Jan 4 02:52:50.765297 DEBUG SM: (3) inserting IP 10.30.25.109

This client has transmitted a broadcast packet that we can snoop (ARP request) to find it's IP address, so SM is noting the IP address. This IP will show up in the output of "show sessions network".

SM Jan 4 02:53:04.810810 DEBUG SM-EVENT: (3) rssi -71, rate 96, idle 0 secs

SM Jan 4 02:53:04.810903 DEBUG SM-EVENT: (3) idle timer 164896 left, reset to 180000 ms

SM Jan 4 02:53:04.811107 DEBUG SM-EVENT: (2) rssi -56, rate 22, idle 7 secs

SM Jan 4 02:53:04.811189 DEBUG SM-EVENT: (2) idle timer 164897 left, reset to 173000 ms

SM level 10 trace of client tear-down (idle disconnect)

SM Feb 02 01:00:19.205631 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 103 secs

SM Feb 02 01:00:19.205793 DEBUG SM-EVENT: (13) idle timer is tracking (77333 to go)

SM Feb 02 01:00:34.205601 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 118 secs

SM Feb 02 01:00:34.205678 DEBUG SM-EVENT: (13) idle timer is tracking (62298 to go)

SM Feb 02 01:00:49.206284 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 133 secs

SM Feb 02 01:00:49.206363 DEBUG SM-EVENT: (13) idle timer is tracking (47398 to go)

SM Feb 02 01:01:04.206675 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 148 secs

SM Feb 02 01:01:04.206784 DEBUG SM-EVENT: (13) idle timer is tracking (32384 to go)

SM Feb 02 01:01:19.207044 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 163 secs

SM Feb 02 01:01:19.207122 DEBUG SM-EVENT: (13) idle timer is tracking (17480 to go)

SM Feb 02 01:01:34.207524 DEBUG SM-EVENT: (13) rssi -68, rate 108, idle 178 secs

SM Feb 02 01:01:34.207601 DEBUG SM-EVENT: (13) idle timer is tracking (2494 to go)

SM Feb 02 01:01:36.677033 DEBUG SM-EVENT: (13): wireless idle timer fired; killing

Client has been unresponsive for 180 seconds, so the idle timer fires and SM begins to remove the session.

SM Feb 02 01:01:36.677129 DEBUG SM-STATE: (13) mac 00:0b:7d:26:9d:d7, flags 28812fh, to change state ACTIVE -> KILLING, by sm_handle_idle_timeout

State transition from “Active” to “Killing” initiated by SM.

SM Feb 02 01:01:36.677199 DEBUG SM-EVENT: (13) sending aaa/sm/notify to fsm AAA author

SM Feb 02 01:01:36.677248 DEBUG SM-EVENT: (13) sending sm/killing to fsm sm central fsm

SM Feb 02 01:01:36.677293 DEBUG SM-EVENT: (13) sending aaa/sm/notify to fsm net/igmp

SM Feb 02 01:01:36.677336 DEBUG SM-EVENT: (13) sending net/dot1x/eapol/dying to fsm net/dot1x/eapol

Let other processes in NOS know that the client’s session is being torn down.

SM Feb 02 01:01:36.677491 DEBUG SM-TRACE: (13) added proc hist @485a9cac (6 by aaa_sm_notification: not roam-out, and SL_AAA set); 6 total

SM Feb 02 01:01:36.678567 DEBUG SM-STATE: (13) remove from metering, curr KILLING, prev ACTIVE

Heard back from other cleanup process.

SM Feb 02 01:01:36.678744 DEBUG SM-EVENT: (13) disabled forwarding for 00:0b:7d:26:9d:d7 on "2050" (tag 1 mapped to 1)

Remove the FDB entry for the client.

SM Feb 02 01:01:36.679018 DEBUG SM-EVENT: (13) decrementing loadbal session on port 2050

Decriment the loadbalance counter on the AP the client was attached to.

SM Feb 02 01:01:36.679116 DEBUG (13) sm_do_client_boot: 00:0b:7d:26:9d:d7 will be removed from AP w/o deauth frame

SM Feb 02 01:01:36.679193 DEBUG SM-EVENT: forcing disassociation (but NOT de-auth) of client at 00:0b:7d:26:9d:d7, port 2050

SM initiated cleanup of the AP by sending de-associate message to AP.

SM Feb 02 01:01:36.680056 DEBUG SM: (13) removing IP 10.30.25.109

SM Feb 02 01:01:36.680175 DEBUG SM: (13) inserting IP 0.0.0.0

Replace client's IP address in session with default setting.

SM Feb 02 01:01:36.680281 DEBUG SM-TRACE: (13) added proc hist @485a9c2c (4 by sm_handle_killing_session); 7 total

SM Feb 02 01:01:36.680341 DEBUG SM-STATE: (13) sm_handle_killing_session clearing kill lock, lock vector now =ch

Heard back from another cleanup process.

SM Feb 02 01:01:36.680393 DEBUG SM-STATE: (13) not deleting from smdb because lock=000ch

Still waiting for all cleanup processes to return.

SM Feb 02 01:01:36.680603 DEBUG SM-TRACE: (13) added proc hist @485a9bac (4 by eapol_kill_client); 8 total

SM Feb 02 01:01:36.680670 DEBUG SM-STATE: (13) eapol_kill_client clearing kill lock, lock vector now =4h

Heard back from another cleanup process.

SM Feb 02 01:01:36.680721 DEBUG SM-STATE: (13) not deleting from smdb because lock=0004h

Still waiting for all cleanup processes to return.

SM Feb 02 01:01:36.686212 DEBUG (13) Final stats packet for 00:0b:7d:26:9d:d7: (tapa @48451287):

SM Feb 02 01:01:36.686279 DEBUG 20 04 00 0e 00 0d 00 1c 10 01 00 00 00 00 00 00

SM Feb 02 01:01:36.686335 DEBUG 00 48 10 02 00 00 00 00 00 00 00 00 01 10 03 00 00

SM Feb 02 01:01:36.686384 DEBUG (@484512a7):

SM Feb 02 01:01:36.686414 DEBUG 00 00 00 00 00 01 10 04 00 00 00 00 00 00 01 1c

SM Feb 02 01:01:36.686462 DEBUG 10 05 00 00 00 00 00 00 e5 b0 10 06 00 00 00 00

SM Feb 02 01:01:36.686567 DEBUG SM-EVENT: 13 send FINAL_STATS_READY to AAA

Built the final session statistics packet, passing to AAA. If RADIUS accounting is enabled, this would be sent out as a RADIUS "Stop" accounting packet.

SM Feb 02 01:01:36.687155 DEBUG SM-TRACE: (13) added proc hist @485a662c (4 by AAA do_kill_processing - final stats); 9 total

SM Feb 02 01:01:36.687229 DEBUG SM-STATE: (13) AAA do_kill_processing - final stats clearing kill lock, lock vector now =0h

Heard back from the last cleanup process, proceeding to kill the session.

SM Feb 02 01:01:36.687301 DEBUG SM-STATE: (13) delete 00:0b:7d:26:9d:d7 from the smdb

Delete the client from the session manager database.

SM Feb 02 01:01:36.687556 DEBUG (13) deleting; ref count history:

1138870896 release_killing_lock by AAA do_kill_processing - final stats

1138870896 release_killing_lock by eapol_kill_client

1138870896 release_killing_lock by sm_handle_killing_session

1138870896 dont_clear_killing_lock by aaa_sm_notification: not roam-out, and SL_AAA set

1138870669 set_killing_lock by AAA new session

1138870669 release_killing_lock by sm_handle_move_conf

1138870668 set_killing_lock by do_vlan

1138870667 set_killing_lock by wifi_association

1138870667 set_killing_lock by sm_dot11_handle_associate

Poorly formatted… Releasing all locks from other processes.

SM Feb 02 01:01:36.687672 DEBUG i_smdb_unpublish_ident_by_mac: (13) unpublishing ident by mac

Removing SMDB entry from internal tables.

SM Feb 02 01:01:36.687867 DEBUG SM: (13) removing IP 0.0.0.0

Remove IP address from session entry.

SM Feb 02 01:01:36.688024 DEBUG sm_sys_free: sys_freeing a ""sm/aaa_blob_internal"" of 56 bytes @4859692c

SM Feb 02 01:01:36.688092 DEBUG sm_sys_free: sys_freeing a ""sm/aaa_blob_internal"" of 56 bytes @4855cc2c

SM Feb 02 01:01:36.688140 DEBUG sm_sys_free: sys_freeing a ""sm/aaa_blob_internal"" of 40 bytes @484f11ac

SM Feb 02 01:01:36.688189 DEBUG sm_sys_free: sys_freeing a ""sm/aaa_blob_copy"" of 50 bytes @4857b6ac

SM Feb 02 01:01:36.688236 DEBUG sm_sys_free: sys_freeing a ""util/string"" of 26 bytes @485963ac

SM Feb 02 01:01:36.688283 DEBUG sm_sys_free: sys_freeing a ""util/string"" of 30 bytes @4859632c

Free memory which had been allocated for session.

SM Feb 02 01:01:36.688331 DEBUG SMDB: (13) set roaming peer (none)

SM Feb 02 01:01:36.688370 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a662c

SM Feb 02 01:01:36.688418 DEBUG SM-TRACE: (13) freed proc history @485a662c

SM Feb 02 01:01:36.688455 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a9bac

SM Feb 02 01:01:36.688502 DEBUG SM-TRACE: (13) freed proc history @485a9bac

SM Feb 02 01:01:36.688538 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a9c2c

SM Feb 02 01:01:36.688585 DEBUG SM-TRACE: (13) freed proc history @485a9c2c

SM Feb 02 01:01:36.688621 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a9cac

SM Feb 02 01:01:36.688667 DEBUG SM-TRACE: (13) freed proc history @485a9cac

SM Feb 02 01:01:36.688703 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a69ac

SM Feb 02 01:01:36.688749 DEBUG SM-TRACE: (13) freed proc history @485a69ac

SM Feb 02 01:01:36.688785 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a612c

SM Feb 02 01:01:36.688832 DEBUG SM-TRACE: (13) freed proc history @485a612c

SM Feb 02 01:01:36.688868 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @484ab32c

SM Feb 02 01:01:36.688950 DEBUG SM-TRACE: (13) freed proc history @484ab32c

SM Feb 02 01:01:36.688994 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @485a68ac

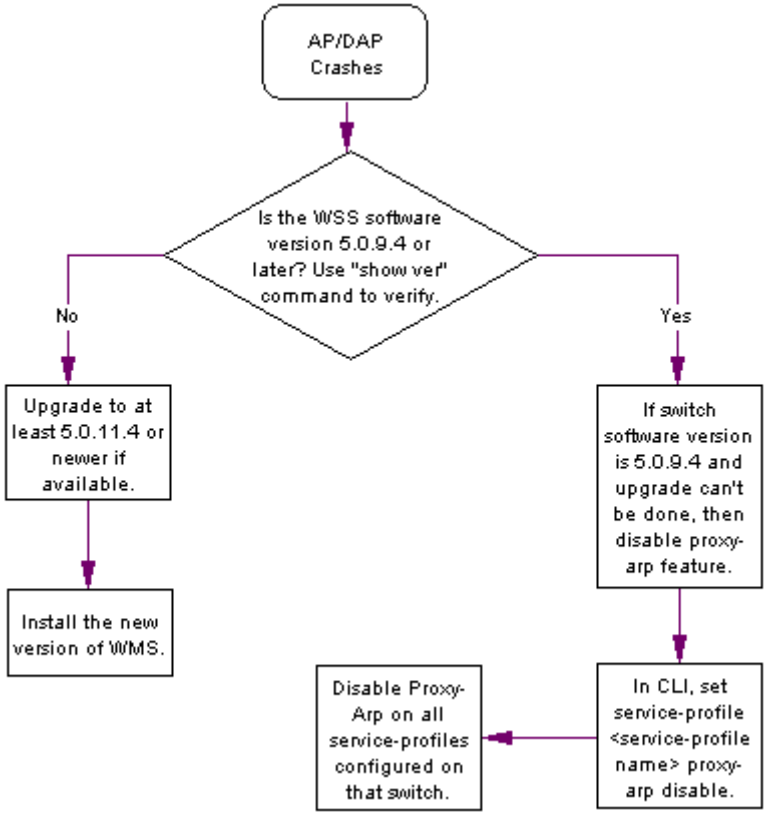
SM Feb 02 01:01:36.689040 DEBUG SM-TRACE: (13) freed proc history @485a68ac

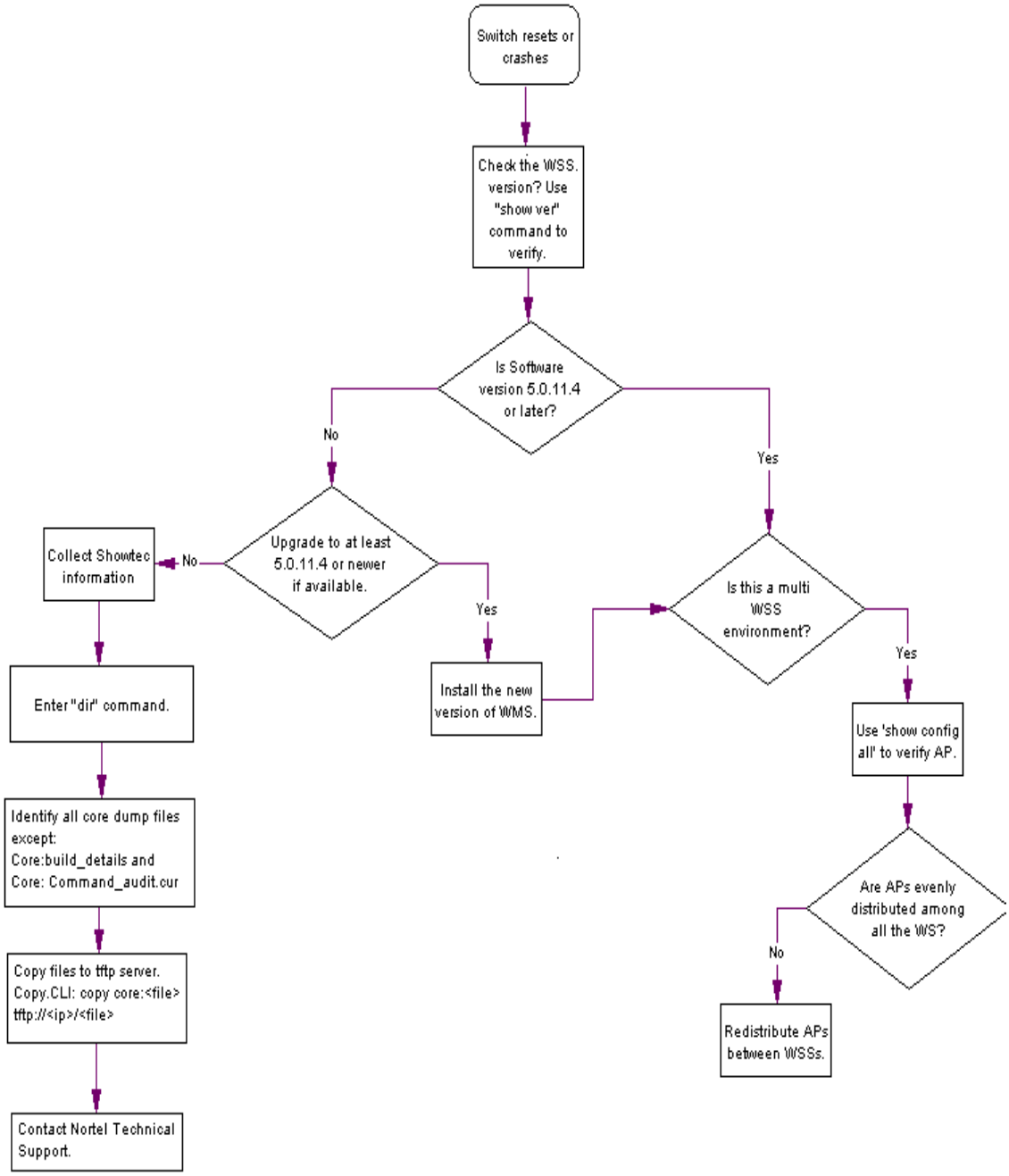
SM Feb 02 01:01:36.689077 DEBUG sm_sys_free: sys_freeing a ""sm/sm_process_history"" of 112 bytes @484ab42c

SM Feb 02 01:01:36.689124 DEBUG SM-TRACE: (13) freed proc history @484ab42c

Release all processes involved in session.

Emergency Recovery Tree




Switch resets or crashes
Check the WSS. version? Use "show ver" command to verify.
Is Software version 5.0.11.4 or later?
No
Yes
Upgrade to at least 5.0.11.4 or newer if available.
No
Yes
Collect Showtec information
Enter "dir" command.
Identify all core dump files except: Core:build_details and Core: Command_audit.cur
Copy files to tftp server. Copy.CLI: copy core:<file> tftp://<ip>/<file>
Contact Nortel Technical Support.
Install the new version of WMS.
Is this a multi WSS environment?
Yes
Use 'show config all' to verify AP.
Are APs evenly distributed among all the WS?
No
Redistribute APs between WSSs.

Nortel WLAN Security Switch 2300 Series Troubleshooting and Debug Guide

Nortel WLAN—Security Switch 2300 Series
Release 7.0

Sourced in Canada, the United States of America, and India
Document Number: **NN47250-700**
Document Status: **Standard**
Document Version: **03.01**
Release Date: **November 2008**

Copyright © Nortel Networks Limited 2008 All Rights Reserved
The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Nortel Networks.
*Nortel, Nortel (Logo), the Globemark, and This is the way, This is Nortel (Design mark) are trademarks of Nortel Networks.
*Microsoft, MS, MS-DOS, Windows, and Windows NT are registered trademarks of Microsoft Corporation.
All other trademarks and registered trademarks are the property of their respective owners.
To provide feedback, or to report a problem in this document, go to www.nortel.com/documentfeedback.