Remote Debugging

HOW TO GUIDE

February 2012

Revision 1

Table of Contents

Table of Contents 3
1. Remote File Retrieval 4
  1.1 Crash Information Files 4
  1.2 Tech Support Files 8
2. Live Packet Captures 10
  2.1 CLI Command Structure 10
  2.2 Use Cases / Examples 19
3. Wireless Debugging 26
  3.1 CLI Command Structure 26
  3.2 Use Cases / Examples 31

1. Remote File Retrieval

1.1 Crash Information Files

Crash files are created when the WiNG 5.x software an Access Point or Controller encounters a critical error or malfunction. Crash information is useful for identifying and troubleshooting hardware and software issues on a WiNG 5.x device and are used by Zebra Technologies Technical Support to troubleshoot and identify root causes of technical issues.

1.1.1 Identifying Devices with Crash Files

When a crash file is generated on a Controller or Access Point it is placed into a directory on the devices internal flash storage named ***crashinfo***. The WiNG 5.x software allows administrators to quickly identify devices that have generated crash files using the command line interface (CLI) or Web User Interface (Web-UI).

1.1.1.1 Command Line Prompt

When a Controller or Access Point has generated one or more crash files, the devices CLI prompt will display an asterisk (*). You can identify individual devices with crash files by directly connecting to a WiNG 5.x device using its serial console port or remotely connecting to a WiNG 5.x device using SSHv2 or Telnet.

Prompt Example:

```
rfs7000-2*>
```

1.1.1.2 Command Line Interface

In centrally managed deployments, Access Points which have generated crash files can be identified using the centralized Controllers CLI by issuing the ***show wireless ap on <rf-domain-name>*** command against each RF Domain in the network. Access Points with crash files are displayed with an asterisk (*) in front of their hostname:

CLI Example:

```
RFSX0001# show wireless ap on store-100
```

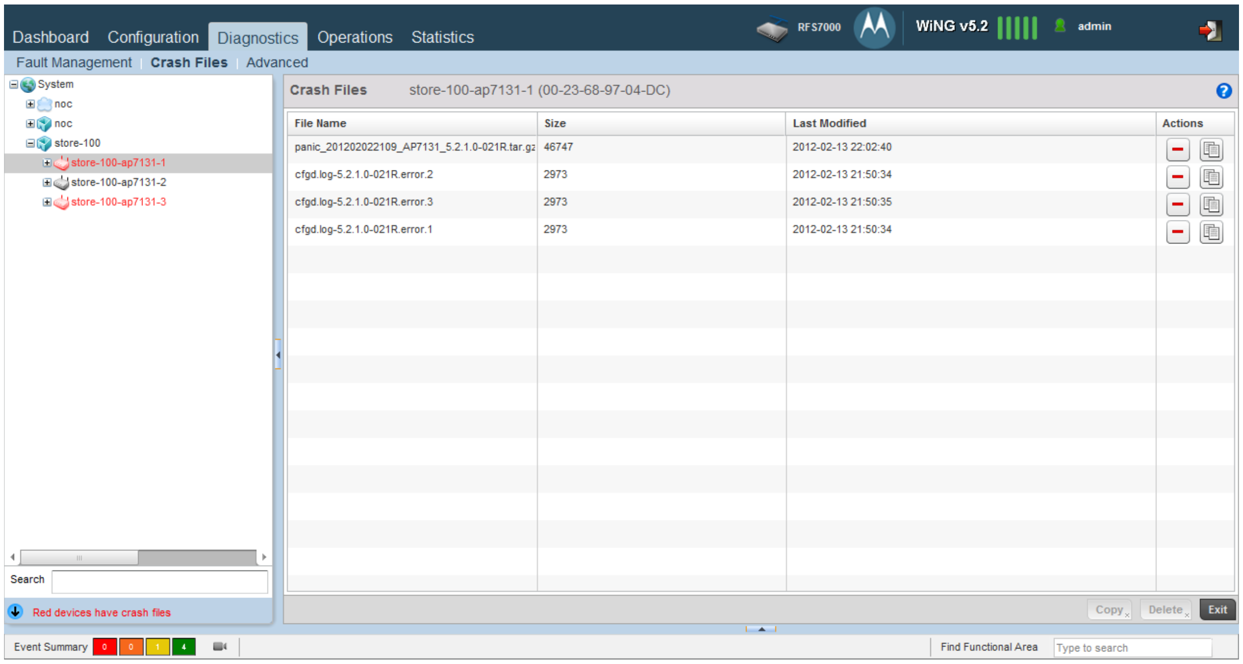
```
--------------------------------------------------------------------------------------------------------------
  MODE        : radio modes - W = WLAN, S=Sensor, ' ' (Space) = radio not present
--------------------------------------------------------------------------------------------------------------
  AP-NAME            AP-LOCATION     RF-DOMAIN        AP-MAC            #RADIOS MODE  #CLIENT          IP
---------------------------------------------------------------------------------------------------------------
* store-100-ap7131-1 JohnsonCityTN   store-100        00-23-68-97-04-DC       2 W-W         0  192.168.21.125
* store-100-ap7131-3 JohnsonCityTN   store-100        00-23-68-99-B9-30       2 W-W         0  192.168.21.124
  store-100-ap7131-2 JohnsonCityTN   store-100        00-23-68-99-B6-7C       2 W-W         0  192.168.21.126
---------------------------------------------------------------------------------------------------------------
```

1.1.1.3 Web User Interface:

WiNG 5.x devices which have generated crash files can be identified using the centralized Controllers Web-UI by selecting ***Diagnostics* → *Crash Files*** and expanding each ***RF Domain***. WiNG 5.x devices that have generated crash files are displayed in red in the tree providing easy identification. You can also view the crash files on the individual devices by selecting the device in the tree:

Web-UI Example:

Diagnostics → Crash Files → <rf-domain-name>



Crash Files store-100-ap7131-1 (00-23-68-97-04-DC)

File Name	Size	Last Modified	Actions
panic_201202022109_AP7131_5.2.1.0-021R.tar.gz	46747	2012-02-13 22:02:40	
cfgd.log-5.2.1.0-021R.error.2	2973	2012-02-13 21:50:34	
cfgd.log-5.2.1.0-021R.error.3	2973	2012-02-13 21:50:35	
cfgd.log-5.2.1.0-021R.error.1	2973	2012-02-13 21:50:34	

1.1.2 Retrieving Crash Information

Once a WiNG 5.x device with crash information has been identified, the crash files can be archived and copied to a centralized location by issuing the ***remote-debug copy-crashinfo*** command on the centralized Controller. This command allows administrators to archive and remotely copy crash files from one or more individual WiNG 5.x devices or all WiNG 5.x devices within a specified RF Domain to a centralized location without having to directly connect to each individual device.

When the ***remote-debug copy-crashinfo*** command is issued against a device, the WiNG 5.x device will archive all the crash files within its ***crashinfo*** directory into a single TAR file and copy the archive to the specified destination URL. The crash file archives can be copied to Compact Flash or USB storage installed on the centralized Controller or a centralized FTP or TFTP server.

1.1.2.1 Command Syntax

The following provides the command syntax for the ***remote-debug copy-crashinfo*** command:

Syntax:

```
remote-debug copy-crashinfo hosts <host> <host> | rf-domain <rf-domain-name> write <url>

        <host> - Name of the remote system (multiple names separate by spaces)

        <rf-domain> - RF Domain Name

        <url> - Destination / Path:
                tftp://<hostname|IP>[:port]/path/
                ftp://<user>:<passwd>@<hostname|IP>[:port]/path/
                usb1:/path
                usb2:/path
                cf:/path
```

1.1.2.2 Examples

The following CLI example demonstrates how to copy remote crash information from an individual WiNG 5.x device to USB storage installed within the centralized Controller:

Example:

```
RFSX000# remote-debug copy-crashinfo hosts store-100-ap7131-1 write usb1:/
```

The following CLI example demonstrates how to copy remote crash information from multiple individual WiNG 5.x devices to a centralized FTP server:

Example:

```
RFSX000# remote-debug copy-crashinfo hosts store-100-ap7131-1 store-102-ap7131-8 store-216-
ap6532-21 write ftp://user:password@192.168.10.10/crashinfo/
```

The following CLI example demonstrates how to copy remote crash information from all WiNG 5.x devices within a specified RF Domain to a centralized TFTP server:

Example:

```
RFSX000# remote-debug copy-crashinfo rf-domain store-100 write tftp://192.168.10.10/crashinfo/
```

Note – The retrieval of crash information files can be stopped by issuing the ***remote-debug end-session copy-crashinfo*** *command.*

Note – Retrieving crash information from multiple remote Access Points within an RF Domain can potentially generate a significant amount of traffic over the WAN. If WAN bandwidth is constrained, it is recommended that the remote crash files be retrieved during off peak hours.

1.1.3 Clearing Crash Information

Once the crash information has been retrieved from devices they can be cleared by issuing the ***remote-debug clear-crashinfo*** command on the centralized Controller. Crash files can be cleared from individual devices as well as all devices within a specified RF Domain. Once this command is issued all crash information will be lost.

1.1.3.1 Command Syntax

The following provides the command syntax for the ***remote-debug clear-crashinfo*** command:

Syntax:

```
remote-debug clear-crashinfo hosts <host> <host> | rf-domain <rf-domain-name>
```

1.1.3.2 Examples

The following CLI example demonstrates how to clear crash information on an individual WiNG 5.x device:

Example:

```
RFSX000# remote-debug clear-crashinfo hosts store-100-ap7131-1
```

The following CLI example demonstrates how to clear crash information on multiple WiNG 5.x devices:

Example:

```
RFSX000# remote-debug clear-crashinfo hosts store-100-ap7131-1 store-100-ap7131-3
```

The following CLI example demonstrates how to clear crash information from all WiNG 5.x devices within a specified RF Domain:

Example:

```
RFSX000# remote-debug clear-crashinfo rf-domain store-100
```

1.2 Tech Support Files

When troubleshooting an issue on a WiNG 5.x device or remote site, Zebra Technologies Technical Support may ask for one or more tech support files. Each tech support file is an archive that contains a detailed snapshot of a WiNG 5.x devices configuration, active processes, utilization, log files in addition to the outputs of various show commands. The tech support files are used by Zebra Technologies to identify hardware, software or deployment issues.

Tech support files can be archived and copied to a centralized location by issuing the ***remote-debug copy-techsupport*** command on the centralized Controller. This command allows administrators to archive and remotely copy tech support files from one or more individual WiNG 5.x devices or all WiNG 5.x devices within a specified RF Domain to a centralized location without having to directly connect to each individual device.

1.2.1.1 CLI Command Syntax

The following provides the command syntax for the ***remote-debug copy-techsupport*** command:

Command Syntax:

```
remote-debug copy-techsupport hosts <host> <host> | rf-domain <rf-domain-name> write <url>
```

```
        <host> - Name of the remote system (multiple names separate by spaces)

        <rf-domain-name> - RF Domain Name

        <url> - Destination / Path:
               tftp://<hostname|IP>[:port]/path/
               ftp://<user>:<passwd>@<hostname|IP>[:port]/path/
               usb1:/path
               usb2:/path
               cf:/path
```

1.2.1.2 Examples

The following CLI example demonstrates how to copy tech support information from an individual WiNG 5.x device to USB storage installed within the centralized Controller:

Example:

```
RFSX000# remote-debug copy-techsupport hosts store-100-ap7131-1 write usb1:/
```

The following CLI example demonstrates how to copy tech support information from multiple individual WiNG 5.x devices to a centralized FTP server:

Example:

```
RFSX000# remote-debug copy-techsupport hosts store-100-ap7131-1 store-102-ap7131-8 store-216-
ap6532-21 write ftp://user:password@192.168.10.10/techsupport/
```

The following CLI example demonstrates how to copy tech support information from all WiNG 5.x devices within a specified RF Domain to a centralized TFTP server:

Example:

```
RFSX000# remote-debug copy-techsupport rf-domain store-100 write
tftp://192.168.10.10/techsupport/
```

*Note – The retrieving of tech support files can be stopped by issuing the **remote-debug end-session copy-techsupport** command.*

Note – Retrieving tech support information from multiple remote Access Points within an RF Domain can potentially generate a significant amount of traffic over the WAN. If WAN bandwidth is constrained, it is recommended that the tech support information be retrieved during off peak hours.

Note – Some TFTP servers will time-out while the tech-support information is being archived by the WiNG 5.x device. To eliminate technical issues during the archive a file transfer it is recommended that you use a FTP server whenever possible.

2. Live Packet Captures

Each WiNG 5.x device includes a sophisticated integrated packet capture facility that can capture wired and wireless traffic at any point within the WiNG 5.x device. The live packet capture feature allows administrators to initiate packet captures on one or more remote WiNG 5.x devices or RF Domains and centrally view the packet captures in real-time.

The main benefit of the live packets capture feature is that it provides administrators with a fully integrated distributed sniffer system that has full visibility into wired and wireless traffic at a remote site eliminating the need for deploying standalone distributed sniffers to remotely troubleshoot connectivity, wireless client or application issues.

2.1 CLI Command Structure

The following section provides the command structure and parameters for the ***remote-debug live-pktcap*** command:

Command Structure:

```
remote-debug live-pktcap [Scope] [Presentation] [Capture Point] [Count] [Filters]
```

2.1.1 Scope

Live packet captures can be enabled on one or more WiNG 5.x hosts or all WiNG 5.x devices within a specified RF Domain.

When a live packet capture is enabled one or more WiNG 5.x hosts, traffic will only be captured by the specified WiNG 5.x hosts you define. It is possible that capturing traffic on a limited scope of devices may miss the interesting traffic as wireless hosts roam to Access Points not defined within the scope.

Command Syntax:

```
remote-debug live-pktcap hosts <host1> .. [Presentation] [Capture Point] [Count] [Filters]
```

When an RF Domain is specified, traffic can be captured by all the devices within the specified RF Domain. This allows traffic to be captured from all Access Points at a remote site providing full visibility in to the interesting traffic as a wireless hosts roam within the site.

Command Syntax:

```
remote-debug live-pktcap rf-domain <rf-domain-name> [Presentation] [Capture Point] [Count]
[Filters]
```

2.1.2 Presentation

Live packet capture files can be viewed in real-time on the centralized Controllers console, streamed in real-time to Wireshark using the Tazman Sniffer Protocol (TZSP) or saved to an external FTP server for offline viewing.

2.1.2.1 Controller Console

By default when no external FTP server is specified or TZSP is enabled, a summary of a live packet capture will be displayed in real-time on the centralized Controllers console where the live packet capture is initiated. It's important to note that a full packet decode will not be available in the Controllers console. If a full packet decode is required, the live packet capture should be saved to an external FTP or TFTP server or streamed to host running Wireshark using the TZSP protocol.

By default the console will display a summary of the captured packets which includes direction, source / destination, ports, QoS, length and flag information. The following example shows an example packet capture taken from an 802.11N radio:

Example:

```
1 14:42:01.761263 O  DATA Src:C4-64-13-A0-79-0D Dst:01-00-0C-CC-CC-CD Bss:5C-0E-8B-1D-C0-52 (from DS)
ENCRYPTED
2 14:42:02.682772 I TCP: 192.168.12.117 > 192.168.10.10 ports 50277 > 445, data length 39, PA, DF, DSCP 0
3 14:42:02.683769 O TCP: 192.168.10.10 > 192.168.12.117 ports 445 > 50277, data length 39, PA, DF, DSCP 0
4 14:42:02.684047 I TCP: 192.168.12.117 > 192.168.10.10 ports 50277 > 445, data length 43, PA, DF, DSCP 0
```

2.1.2.2 Tazman Sniffer Protocol (TZSP)

The Tazman Sniffer Protocol (TZSP) is designed to encapsulate other protocols over UDP. Live packet captures running on WiNG 5.x devices can be configured to stream the live packet capture data to a centralized host running Wireshark allowing the live packet capture data to be displayed in real-time.

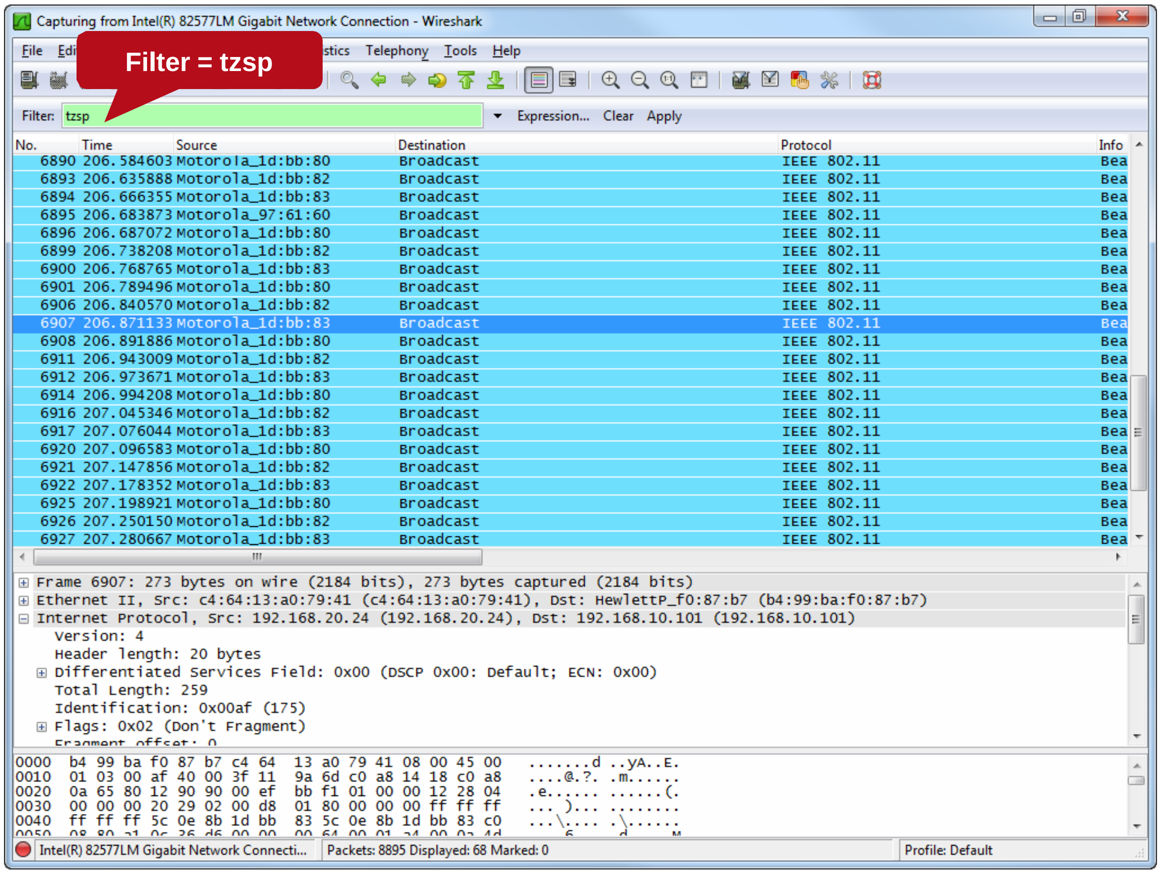
The live packet capture stream is forwarded from the WiNG 5.x devices to the IP address of the host running the Wireshark application. The Wireshark application monitors the physical Ethernet interface that is receiving the TZSP encapsulated data stream and will decode and display the live packet capture in real-time.

The Wireshark application includes a built in display filter which allows Wireshark to filtering out non TZSP traffic. Without the display filter applied Wireshark will display the decoded TZSP frames in addition to any background traffic that is transmitted or received from the Wireshark host.

Command Syntax:

```
remote-debug live-pktcap [Scope] write tzsp <ip-address> | <hostname> [Capture Point] [Count]
[Filters]

        <ip-address> - IP Address of the device running Wireshark
        <hostname> - Hostname of the device running Wireshark
```



```
No.  Time       Source              Destination  Protocol     Info
6890 206.584603 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6893 206.635888 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6894 206.666355 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6895 206.683873 Motorola_97:61:60   Broadcast    IEEE 802.11  Bea
6896 206.687072 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6899 206.738208 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6900 206.768765 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6901 206.789496 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6906 206.840570 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6907 206.871133 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6908 206.891886 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6911 206.943009 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6912 206.973671 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6914 206.994208 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6916 207.045346 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6917 207.076044 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6920 207.096583 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6921 207.147856 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6922 207.178352 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
6925 207.198921 Motorola_1d:bb:80   Broadcast    IEEE 802.11  Bea
6926 207.250150 Motorola_1d:bb:82   Broadcast    IEEE 802.11  Bea
6927 207.280667 Motorola_1d:bb:83   Broadcast    IEEE 802.11  Bea
```

```
⊞ Frame 6907: 273 bytes on wire (2184 bits), 273 bytes captured (2184 bits)
⊞ Ethernet II, Src: c4:64:13:a0:79:41 (c4:64:13:a0:79:41), Dst: HewlettP_f0:87:b7 (b4:99:ba:f0:87:b7)
⊟ Internet Protocol, Src: 192.168.20.24 (192.168.20.24), Dst: 192.168.10.101 (192.168.10.101)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 259
    Identification: 0x00af (175)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
```

```
0000  b4 99 ba f0 87 b7 c4 64  13 a0 79 41 08 00 45 00   .......d ..yA..E.
0010  01 03 00 af 40 00 3f 11  9a 6d c0 a8 14 18 c0 a8   ....@.?. .m......
0020  0a 65 80 12 90 90 00 ef  bb f1 01 00 00 12 28 04   .e...... ......(.
0030  00 00 00 20 29 02 00 d8  01 80 00 00 00 ff ff ff   ... )... ........
0040  ff ff ff 5c 0e 8b 1d bb  83 5c 0e 8b 1d bb 83 c0   ...\.... .\......
```

Intel(R) 82577LM Gigabit Network Connecti... | Packets: 8895 Displayed: 68 Marked: 0 | Profile: Default

Figure 2.1.2.2 – Wireshark TZSP Display Filter

2.1.2.2.1 Listener Application

The TZSP protocol operates over UDP port 37008 and requires a listener application such as ***Netcat*** or ***Iperf*** to be running on the Wireshark host receiving the live packet capture stream. If a listener application is not present on the Wireshark host, the host will respond to each TZSP packet with an ICMP unreachable message which will also be captured and displayed. The listener application provides a valid termination point for the TZSP data stream and eliminates the ICMP unreachable messages.

The following provides an example of how to configure Netcat to terminate the TZSP packet stream:

Example:

```
C:\Program Files (x86)\nc11nt> nc -l -p 37008 -u
```

The following provides an example of how to configure IPerf to terminate the TZSP packet stream:

Example:

```
C:\Program Files (x86)\iperf> iperf -s -u  -p 37008
```

Note – Most current operating systems include an integrated firewall which blocks all inbound traffic by default. It's important to configure the firewall to permit the listener application or UDP traffic destined to port 37008 or the TZSP data stream will be dropped.

2.1.2.3 FTP Server

Live packet captures can be exported to an external FTP server for offline viewing. When an FTP server is specified, the live packet capture data is uploaded as the live packet capture is being performed. White the live packet capture feature supports external TFTP servers; it is recommended that an FTP be utilized as TFTP transfers can timeout as the live packet capture file is being streamed to the server.

Command Syntax:

remote-debug live-pktcap [Scope] write ftp://<user>|<password>@<hostname|IP>/<path>/<filename> [Capture Point] [Filters]

```
        <user> - FTP Username
        <password> - FTP Password
        <hostname|IP> - Hostname or IP Address of the FTP server
        /<path>/ - Optional pathname
        <filename> - Packet capture filename
```

2.1.3 Capture Points

Each Controller and Access Point provides various capture points within the data plane for which traffic can be captured and re-directed to a centralized console in real-time. The capture point you select will determine which point within the data plane on the WiNG 5.X device the traffic is captured from and the format of the Ethernet headers.

Traffic that is captured from the bridge, deny, drop, interface, router, VPN or wireless capture points is presented as Ethernet II frames while traffic captured from a radio will be presented as 802.11 frames. Traffic captured from a radio will also be encrypted depending on the encryption ciphers enabled on the Wireless LANs the radio is servicing.

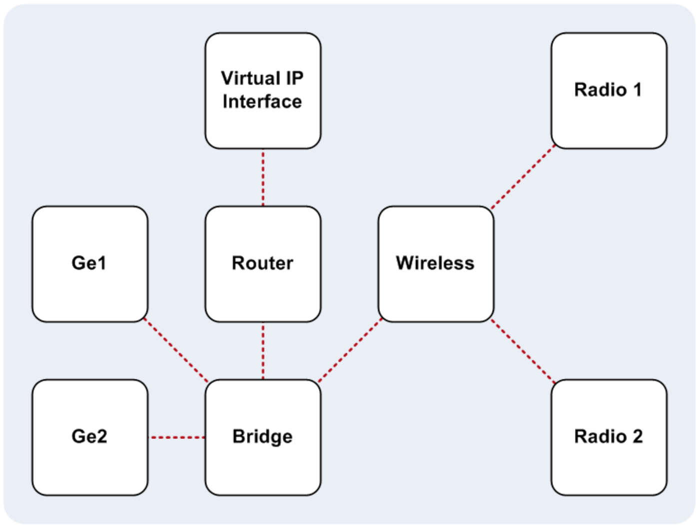


Figure 2.1.3 – Live Packet Capture Points

The following table provides a list of common capture points and the traffic types that they capture:

Capture Point	Description
bridge	Captures packets that are translated through the Ethernet bridge. This includes wired → wireless, wireless → wired and wireless → wireless traffic.
deny	Captures packets that are denied by an IP or MAC Access Control List (ACL).
drop	Captures packets that are dropped by the device for any reason. This includes traffic that is dropped by an Access Control List, traffic dropped when client-to-client communications is denied on a Wireless LAN or traffic destined to a client that has disassociated.
interface	Captures packets transmitted / received over a physical interface, port channel or VLAN.
radio	Captures 802.11 packets transmitted / received over an 802.11N radio. Traffic captured from a radio will be presented as 802.11 frames. The 802.11 frames maybe encrypted depending on the encryption settings for the Wireless LANs the radio is servicing.
router	Captures routed packets transmitted / received between Virtual IP Interfaces. This includes traffic that is routed between virtual IP interfaces on the WiNG 5.x device.

Capture Point	Description
vpn	Captures packets transmitted / received over an IPsec VPN tunnel. This includes wired or wireless traffic that is captured by the assigned IPsec Access Control List and forwarded over the IPsec VPN tunnel. Traffic will be displayed as unencrypted Ethernet II frames.
wireless	Captures packets transmitted / received over Wireless LANs. Traffic that is captured includes wired → wireless, wireless → wired and wireless → wireless communications. Traffic will be displayed as unencrypted Ethernet II frames.

Table 2.1.3 – Live Packet Capture Points

2.1.4 Counts

By default WiNG 5.x will capture up to 50 packets per WiNG 5.x host defined in the capture scope and will stop the packet capture trace once the first host in the scope has received 50 packets you have filtered on. In most cases the default packet capture count is adequate for most remote troubleshooting tasks; however you can increase the packet capture count if required.

The packet capture count can be increased or decreased by specifying a count after the capture point and before the filters in the ***remote-debug live-pktcap*** command. WiNG 5.x can capture a minimum of 1 packet and a maximum of 1,000,000 packets.

Command Structure:

```
remote-debug live-pktcap [Scope] [Presentation] [Capture Point] count <1-1000000> [Filters]
```

When a live packet capture has been initiated at a remote site, the live packet capture will run until the default or specified packet capture count has been reached on one of the WiNG 5.x devices. You will not be able to initiate a new live packet capture until the packet capture count on the current live packet capture has been reached.

To initiate a new packet capture before the capture count has been reached you must issue the ***remote-debug end-session live-pktcap*** command. This will stop the current live packet capture session and allow a new live packet capture session to be initiated.

2.1.5 Filters

The packet capture engine provides flexible filtering support which allows administrators to restrict the scope of packets that are captured and forwarded to the centralized console. By default without any filters applied the first 50 packets from the capture point will be forwarded to the defined centralized console or offline file for viewing.

Filters can be applied to any live packet capture and it strongly recommended that filters be utilized to restrict the number of packets that are captured and forwarded over a wide area network. If the filter scope is too large, the wide area network or centralized console can be quickly saturated. The CPU performance of a WiNG 5.x device can also be impacted if the filter scope is too large.

Filters can be defined to capture traffic at Layer 2 – Layer 4 of the OSI model allowing administrators to capture packets from specific hosts, protocols or applications. Multiple filters can also be defined using a logical AND or OR. For example a filter can be defined to capture traffic between specific hosts, from multiple applications or a specific application and hosts.

The following table provides a list of filters which can be applied to a live packet capture:

Filter	Description
arp	Match ARP Packets
capwap	Match CAPWAP Packets: ▪ ctrl – Match CAPWAP Control Packets ▪ data – Match CAPWAP Data Packets
cdp	Match CDP Packets
dot11	Match 802.11 Packets: ▪ addr – Match 802.11 Address (1 – 3) ▪ beacon – Match 802.11 Beacon Frames ▪ bssid – Match 802.11 BSSID (AA-BB-CC-DD-EE-FF) ▪ ctl – Match 802.11 Control Frames ▪ data – Match 802.11 Data Frames ▪ mgmt. – Match 802.11 Management Frames ▪ probe – Match 802.11 Probe Frames ▪ type – Match 802.11 Sub Types (0 – 7)
dropreason	Match Packet Drop Reason (0 – 65535)
dst	Match IPv4 Destination (A.B.C.D): ▪ net – Match IP in Subnet (A.B.C.D/M) ▪ port – Match TCP or UDP Port (0 – 65535)

Filter	Description
ether	Match Ethernet Frames: ▪ broadcast – Match Ethernet Broadcasts ▪ dst – Match Ethernet Destination (AA-BB-CC-DD-EE-FF) ▪ host – Match Ethernet Host (AA-BB-CC-DD-EE-FF) ▪ multicast – Match Ethernet Multicasts ▪ proto – Match Ethernet Protocol (0 – 65535) ▪ src – Match Ethernet Source (AA-BB-CC-DD-EE-FF)
host	Match IPv4 Address (A.B.C.D)
icmp	Match ICMP Packets
igmp	Match IGMP Packets
ip	Match IPv4 Packets: ▪ multicast – Match IPv4 Multicast Packets ▪ proto – Match IP Protocol (0 – 255)
ipv6	Match IPv6 Packets
l2	Match L2 Header: ▪ u8 – Match 8 bits (Offset 0 – 127) ▪ u16 – Match 16 bits (Offset 0 – 126) ▪ u32 – Match 32 bits (Offset 0 – 124)
l3	Match L3 Header: ▪ u8 – Match 8 bits (Offset 0 – 127) ▪ u16 – Match 16 bits (Offset 0 – 126) ▪ u32 – Match 32 bits (Offset 0 – 124)
l4	Match L4 Header: ▪ u8 – Match 8 bits (Offset 0 – 127) ▪ u16 – Match 16 bits (Offset 0 – 126) ▪ u32 – Match 32 bits (Offset 0 – 124)
lldp	Match LLDP Packets

Filter	Description
mint	Match MINT Packets: ▪ dst – Match MINT Destination (AA.BB.CC.DD) ▪ host – Match MINT Host Address (AA.BB.CC.DD) ▪ port – Match MINT Port (0 – 65535) ▪ proto – Match MINT Protocol (0 – 7) ▪ src – Match MINT Source (AA.BB.CC.DD)
net	Match IPv4 Subnet (A.B.C.D/M)
not	Logical NOT
port	Match TCP or UDP Port (0 – 65535)
priority	Match 802.1p Priority (0 – 7)
radio	Match Radio (1 – 3)
src	Match IPv4 Source (A.B.C.D): ▪ net – Match IP in Subnet (A.B.C.D/M) ▪ port – Match TCP or UDP Port (0 – 65535)
stp	Match Spanning Tree Protocol Packets
tcp	Match TCP Packets: ▪ ack – Match TCP ACK Packets ▪ fin – Match TCP FIN Packets ▪ rst – Match TCP Reset Packets ▪ syn – Match TCP SYN Packets
udp	Match UDP Packets:
vlan	Match VLAN (1 – 4095)
wlan	Match Wireless LAN (1 – 256)

Table 2.1.4 – Live Packet Capture Filters

Note – To reduce WAN and device overhead, it is strongly recommended that filters be defined for each live packet capture that is performed. Failure to define a filter may result in WAN saturation in addition to loss of communication with the devices performing the live packet capture.

2.2 Use Cases / Examples

2.2.1 Use Case 1 – Wireless Firewall Dropping Packets

The following example demonstrates how to use live packet captures to aid in remotely troubleshooting wireless firewall communication issues. In this scenario a wireless host ***E0-F8-47-0F-0E-14*** in a remote distribution center is unable to reach a centralized web-based application server ***192.168.10.6***. The wireless host is able to access all other applications fine.

A live packet capture is first initiated on the distribution centers RF Domain to determine if traffic from the remote Wireless host destined to the application server is being forwarded onto the wired network. If the traffic destined to the application server is being forwarded onto the wired network, the issue resides outside the Wireless LAN infrastructure. However no traffic is seen on the Access Points Ge1 interfaces destined to the application server:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain dc-102 interface ge1 filter ether host E0-F8-47-0F-
0E-14
```

```
[ap6532-2] 1 16:28:51.476167 O UDP: 192.168.13.101 > 192.168.10.6 ports 61596 > 53, data length 39, DSCP 0

[ap6532-2] 2 16:28:51.476675 O UDP: 192.168.13.101 > 192.168.10.6 ports 54818 > 53, data length 39, DSCP 0

[ap6532-2] 3 16:28:51.524509 I UDP: 192.168.10.6 > 192.168.13.101 ports 53 > 61596, data length 55, DSCP 0

[ap6532-2] 4 16:28:51.604880 I UDP: 192.168.10.6 > 192.168.13.101 ports 53 > 54818, data length 39, DSCP 0

[ap6532-2] 5 16:28:51.606427 O TCP: 192.168.13.101 > 209.59.186.108 ports 49344 > 80, S, MSS 1460, WS <<3,
SACK, DF,

P 0

[ap6532-2] 6 16:28:51.680608 I TCP: 209.59.186.108 > 192.168.13.101 ports 80 > 49344, SA, MSS 1380, SACK, DF,
DSCP 0

[ap6532-2] 7 16:28:51.681317 O TCP: 192.168.13.101 > 209.59.186.108 ports 49344 > 80, A, DF, DSCP 0

[ap6532-2] 8 16:28:51.681762 O TCP: 192.168.13.101 > 209.59.186.108 ports 49344 > 80, data length 544, PA, DF,
DSCP 0

[ap6532-2] 9 16:28:51.768580 I TCP: 209.59.186.108 > 192.168.13.101 ports 80 > 49344, data length 417, PA, DF,
DSCP 0

[ap6532-2] 10 16:28:51.769428 O TCP: 192.168.13.101 > 209.59.186.108 ports 49344 > 80, A, DF, DSCP 0
```

A live packet capture is then initiated to determine if the wireless firewall is dropping any traffic from the Wireless host. The live packet capture on the distribution centers RF Domain reveals that an ACL named ***Corp*** applied to all inbound traffic on the corporate Wireless LAN is dropping traffic destined to the application server in question. The ***Corp*** ACL was recently added to the Wireless LAN for additional security and based on these results it needs to be modified to permit access to this specific application server:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain dc-102 deny filter ether host E0-F8-47-0F-0E-14
```

```
[ap6532-2] 1 16:00:05.864855 I "Corp:ip"/0 TCP: 192.168.13.101 > 192.168.10.6 ports 49333 > 80, S, MSS 1460,
WS <<3, SACK, DF, DSCP 0

[ap6532-2] 2 16:00:06.776872 I "Corp:ip"/0 TCP: 192.168.13.101 > 192.168.10.6 ports 49333 > 80, S, MSS 1460,
WS <<3, SACK, DF, DSCP 0

[ap6532-2] 3 16:00:07.779865 I "Corp:ip"/0 TCP: 192.168.13.101 > 192.168.10.6 ports 49333 > 80, S, MSS 1460,
WS <<3, SACK, DF, DSCP 0

[ap6532-2] 4 16:00:08.781000 I "Corp:ip"/0 TCP: 192.168.13.101 > 192.168.10.6 ports 49333 > 80, S, MSS 1460,
WS <<3, SACK, DF, DSCP 0
```

2.2.2 Use Case 2 – Application Disconnect Issues

The following example demonstrates how to use live packet capture to remotely troubleshoot application connection issues. In this scenario multiple wireless hosts at a distribution center are experiencing intermittent connectivity issues with a centralized application running over ***SSHv2***. All wireless clients at the site experiences frequent disconnects from the application which is being attributed to a Wireless LAN issue.

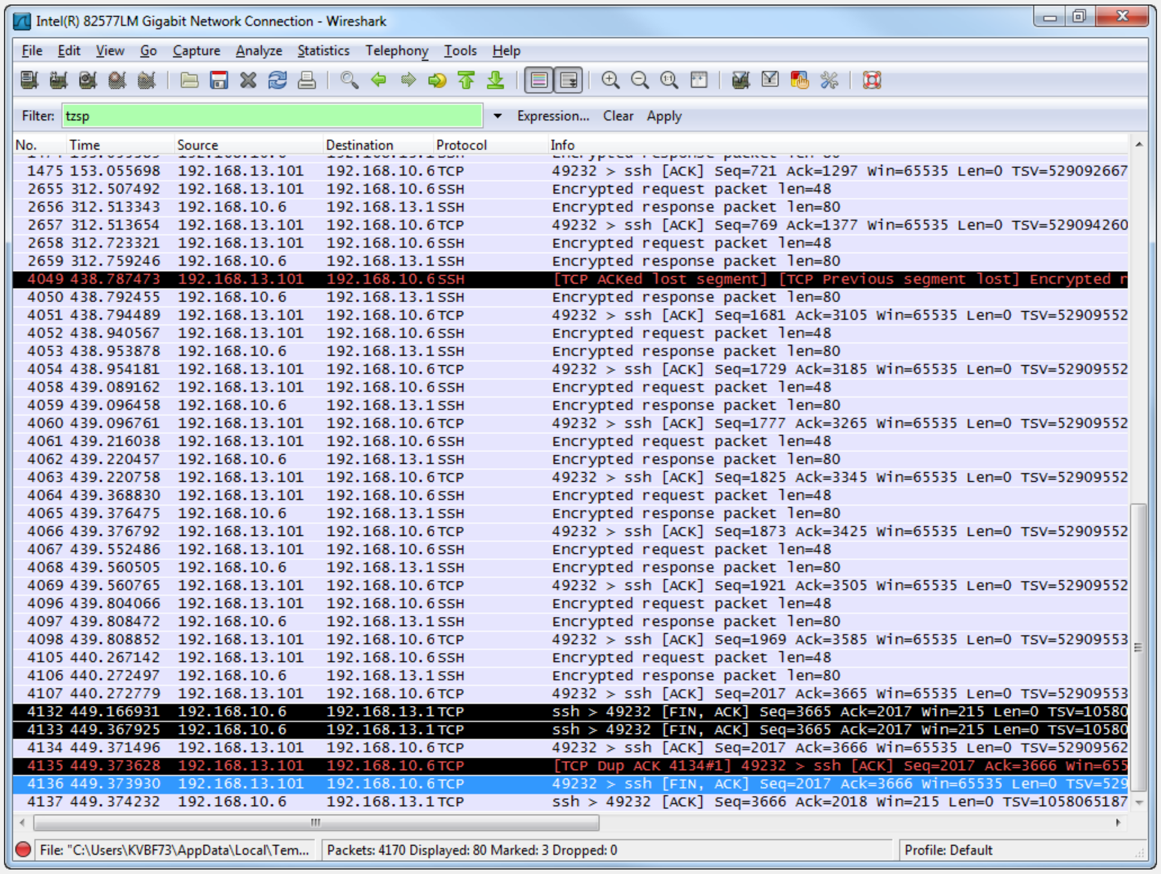
A live packet capture is first initiated on the distribution centers RF Domain to monitor the SSHv2 session between an effected wireless client and the application server over the Wireless LAN. The live packet capture is streamed using TZSP to a centralized host ***192.168.10.101*** running Wireshark for full decoding.

The live packet capture shows that the wireless client ***192.168.13.101*** is communicating fine with the application server ***192.168.10.6*** until the application server abruptly sends ***TCP FIN*** packet to the wireless client. This causes the wireless client to disconnect from the application.

The cause of the application disconnect is related to the application server prematurely disconnecting the clients session either due to inactivity or an internal application error. As the issue is not Wireless LAN related, the trouble ticket is forwarded to the application team for resolution:

Example:

```
RFSX000# ap6532-1#remote-debug live-pktcap rf-domain store100 write
ftp://ftpuser:hellomoto@192.168.10.101/capture1.cap radio 2 filter ether host E0-F8-47-0F-0E-14
```


Intel(R) 82577LM Gigabit Network Connection - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: tzsp
Expression... Clear Apply

```
No.  Time        Source          Destination    Protocol  Info
1475 153.055698  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=721 Ack=1297 Win=65535 Len=0 TSV=529092667
2655 312.507492  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
2656 312.513343  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
2657 312.513654  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=769 Ack=1377 Win=65535 Len=0 TSV=529094260
2658 312.723321  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
2659 312.759246  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4049 438.787473  192.168.13.101  192.168.10.6   SSH       [TCP ACKed lost segment] [TCP Previous segment lost] Encrypted r
4050 438.792455  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4051 438.794489  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1681 Ack=3105 Win=65535 Len=0 TSV=52909552
4052 438.940567  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4053 438.953878  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4054 438.954181  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1729 Ack=3185 Win=65535 Len=0 TSV=52909552
4058 439.089162  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4059 439.096458  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4060 439.096761  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1777 Ack=3265 Win=65535 Len=0 TSV=52909552
4061 439.216038  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4062 439.220457  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4063 439.220758  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1825 Ack=3345 Win=65535 Len=0 TSV=52909552
4064 439.368830  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4065 439.376475  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4066 439.376792  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1873 Ack=3425 Win=65535 Len=0 TSV=52909552
4067 439.552486  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4068 439.560505  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4069 439.560765  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1921 Ack=3505 Win=65535 Len=0 TSV=52909552
4096 439.804066  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4097 439.808472  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4098 439.808852  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=1969 Ack=3585 Win=65535 Len=0 TSV=52909553
4105 440.267142  192.168.13.101  192.168.10.6   SSH       Encrypted request packet len=48
4106 440.272497  192.168.10.6    192.168.13.1   SSH       Encrypted response packet len=80
4107 440.272779  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=2017 Ack=3665 Win=65535 Len=0 TSV=52909553
4132 449.166931  192.168.10.6    192.168.13.1   TCP       ssh > 49232 [FIN, ACK] Seq=3665 Ack=2017 Win=215 Len=0 TSV=10580
4133 449.367925  192.168.10.6    192.168.13.1   TCP       ssh > 49232 [FIN, ACK] Seq=3665 Ack=2017 Win=215 Len=0 TSV=10580
4134 449.371496  192.168.13.101  192.168.10.6   TCP       49232 > ssh [ACK] Seq=2017 Ack=3666 Win=65535 Len=0 TSV=52909562
4135 449.373628  192.168.13.101  192.168.10.6   TCP       [TCP Dup ACK 4134#1] 49232 > ssh [ACK] Seq=2017 Ack=3666 Win=655
4136 449.373930  192.168.13.101  192.168.10.6   TCP       49232 > ssh [FIN, ACK] Seq=2017 Ack=3666 Win=65535 Len=0 TSV=529
4137 449.374232  192.168.10.6    192.168.13.1   TCP       ssh > 49232 [ACK] Seq=3666 Ack=2018 Win=215 Len=0 TSV=1058065187
```

File: "C:\Users\KVBF73\AppData\Local\Tem... Packets: 4170 Displayed: 80 Marked: 3 Dropped: 0 Profile: Default

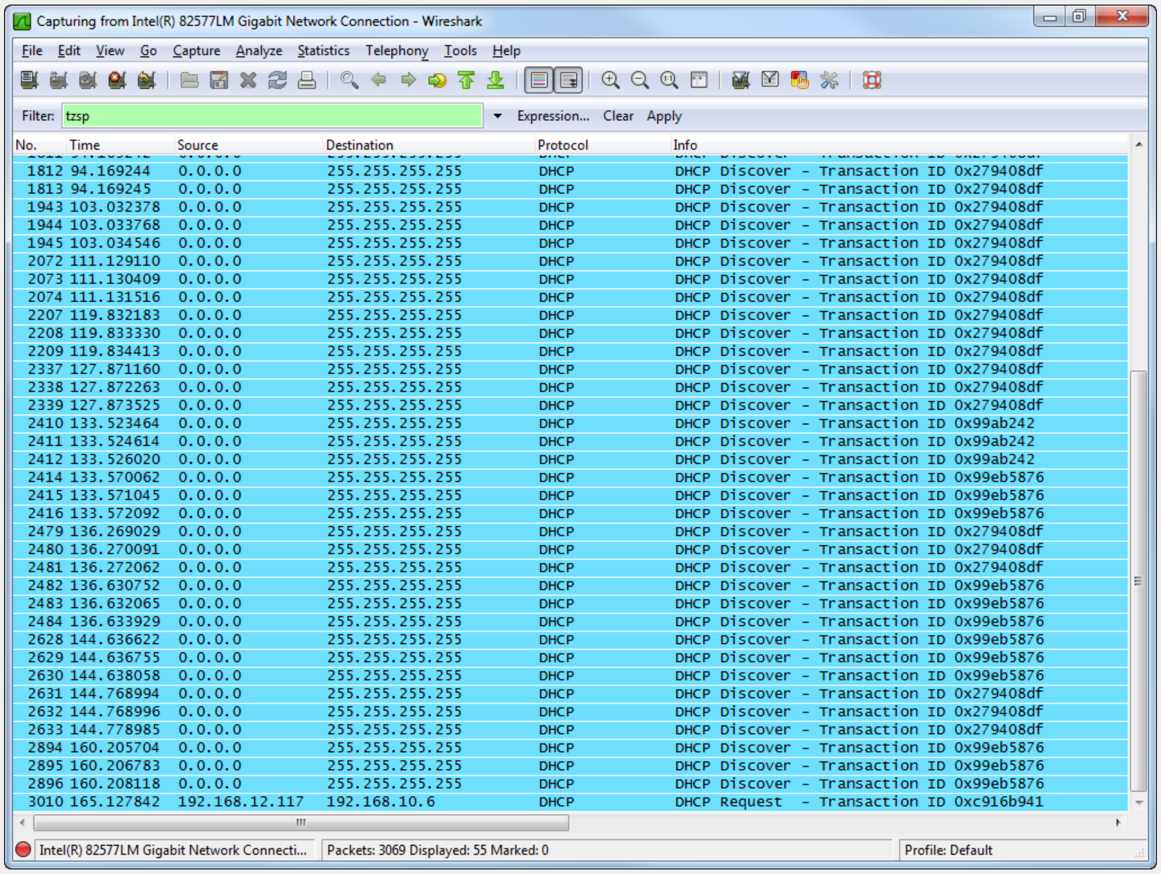
2.2.3 Use Case 3 – Clients not Obtaining IP Addresses

The following example demonstrates how to use live packet captures to aid in remotely troubleshooting wireless clients which are unable to obtain network addressing from a local ***DHCP*** server. In the following scenario a wireless host ***E0-F8-47-0F-0E-14*** in a remote store is unable to obtain an IP address on the POS Wireless LAN. The device is associated and authenticated to the Wireless LAN and assigned to the correct VLAN.

A live packet capture is first initiated on the stores RF Domain to monitor DHCP requests being bridged by the Access Points at the remote store. To provide a full decode the packet capture is streamed using TZSP to a centralized host ***192.168.10.101*** at the headquarters running Wireshark. The live packet capture reveals that ***DHCP Discover*** packets are being received by the Access Point from the Wireless client but no ***DHCP Offers*** or ***DHCP ACKs*** are seen from the DHCP server:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain store100 write tzsp 192.168.10.101 bridge filter port
67 and port 68
```


Capturing from Intel(R) 82577LM Gigabit Network Connection - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: tzsp
Expression... Clear Apply

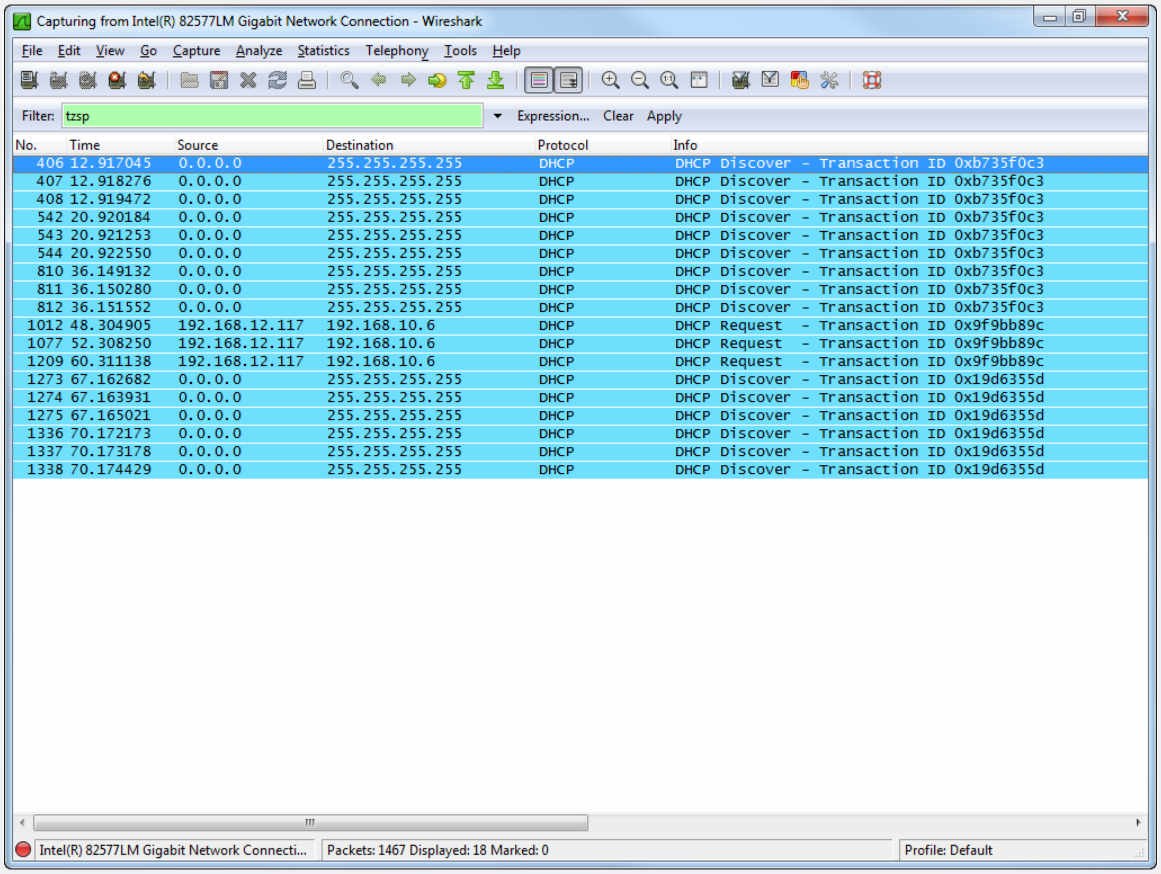
```
No.  Time       Source          Destination      Protocol  Info
1812 94.169244  0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
1813 94.169245  0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
1943 103.032378 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
1944 103.033768 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
1945 103.034546 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2072 111.129110 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2073 111.130409 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2074 111.131516 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2207 119.832183 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2208 119.833330 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2209 119.834413 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2337 127.871160 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2338 127.872263 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2339 127.873525 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2410 133.523464 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99ab242
2411 133.524614 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99ab242
2412 133.526020 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99ab242
2414 133.570062 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2415 133.571045 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2416 133.572092 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2479 136.269029 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2480 136.270091 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2481 136.272062 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2482 136.630752 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2483 136.632065 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2484 136.633929 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2628 144.636622 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2629 144.636755 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2630 144.638058 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2631 144.768994 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2632 144.768996 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2633 144.778985 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x279408df
2894 160.205704 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2895 160.206783 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
2896 160.208118 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x99eb5876
3010 165.127842 192.168.12.117  192.168.10.6     DHCP      DHCP Request  - Transaction ID 0xc916b941

Intel(R) 82577LM Gigabit Network Connecti...   Packets: 3069 Displayed: 55 Marked: 0   Profile: Default
```

A live packet capture is then initiated to see if the ***DHCP Discover*** packets are being forwarded onto the wired network by the Access Points where the DHCP server resides. The live packet capture reveals that the ***DHCP Discover*** packets are being forwarded onto the wired network but no ***DHCP Offers*** or ***DHCP ACKs*** are being received on the Access Points Ge1 interfaces. This indicates that the DHCP server at store is down or the DHCP service has stopped. As this is not a Wireless LAN issue the ticket is forwarded to the appropriate server team for resolution:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain store100 write tzsp 192.168.10.101 interface ge1
filter port 67 and port 68
```


Capturing from Intel(R) 82577LM Gigabit Network Connection - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: tzsp
Expression... Clear Apply

```
No.  Time      Source          Destination      Protocol  Info
 406 12.917045 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 407 12.918276 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 408 12.919472 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 542 20.920184 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 543 20.921253 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 544 20.922550 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 810 36.149132 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 811 36.150280 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
 812 36.151552 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0xb735f0c3
1012 48.304905 192.168.12.117  192.168.10.6     DHCP      DHCP Request  - Transaction ID 0x9f9bb89c
1077 52.308250 192.168.12.117  192.168.10.6     DHCP      DHCP Request  - Transaction ID 0x9f9bb89c
1209 60.311138 192.168.12.117  192.168.10.6     DHCP      DHCP Request  - Transaction ID 0x9f9bb89c
1273 67.162682 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
1274 67.163931 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
1275 67.165021 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
1336 70.172173 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
1337 70.173178 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
1338 70.174429 0.0.0.0         255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x19d6355d
```

Intel(R) 82577LM Gigabit Network Connecti... | Packets: 1467 Displayed: 18 Marked: 0 | Profile: Default

2.2.4 Use Case 4 – QoS Markings

The following example demonstrates how to use live packet captures to verify QoS markings for voice deployments. In this scenario a Wireless IP phone ***00-40-96-AD-4C-F6*** has been deployed in a remote store. For correct treatment over the wired and wireless network SIP signaling traffic is marked with a DSCP value ***26*** (AF31) and RTP traffic is marked with a DSCP value ***46*** (EF) which is prioritized by the wired and Wireless LAN infrastructure.

A live packet capture is first initiated on the stores RF Domain to determine if ***SIP*** (UDP 5060) traffic received by the Access Points from the wired network and the IP phone is marked correctly.

The live packet capture shows that SIP traffic from the call server ***192.168.10.10*** destined to the wireless IP phone ***192.168.12.129*** is being correctly marked with a DSCP value ***26***. In addition the live packet capture shows that SIP traffic from the wireless IP phone destined to the call server is also being marked with the DSCP value ***26***:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain store100 interface ge1 filter port 5060 and ether
host 00-40-96-AD-4C-F6
```

```
[ap6532-2] 1 10:40:23.791015 O UDP: 192.168.12.129 > 192.168.10.10 ports 1055 > 5060, data length
1478, DSCP 26

[ap6532-2] 2 10:40:23.969083 O UDP: 192.168.10.10 > 192.168.12.129 ports 1055 > 5060, data length
1478, DSCP 26

[ap6532-2] 3 10:40:23.969494 O UDP: 192.168.12.129 > 192.168.10.10 ports 1055 > 5060, data length
1478, DSCP 26

[ap6532-2] 4 10:40:23.969910 O UDP: 192.168.10.10 > 192.168.12.129 ports 1055 > 5060, data length
1478, DSCP 26

[ap6532-2] 5 10:40:23.970279 O UDP: 192.168.12.129 > 192.168.10.10 ports 1055 > 5060, data length
1478, DSCP 26
```

A live packet capture is then initiated to determine if RTP traffic received by the Access Points from the IP phones or media gateways on the wired network destined and the wireless IP Phone is being marked correctly. As the RTP port range is large, a filter is applied to only capture UDP traffic from the wireless IP phone during an active voice call.

The live packet capture shows that RTP traffic from a media gateway ***192.168.10.11*** destined to the wireless IP phone ***192.168.12.129*** is being marked with a DSCP value ***46***. In addition the live packet capture shows that RTP traffic from the wireless IP phone destined to the media gateway is also being marked with the DSCP value ***46***:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain store100 wireless filter udp and ether host 00-40-96-
AD-4C-F6
```

```
[ap6532-2] 1 11:09:01.568320 I UDP: 192.168.12.129 > 192.168.10.11 ports 16888 > 16384, data
length 1478, DSCP 46

[ap6532-2] 2 11:09:01.742544 I UDP: 192.168.12.129 > 192.168.10.11 ports 16888 > 16384, data
length 1478, DSCP 46

[ap6532-2] 3 11:09:01.742923 I UDP: 192.168.12.129 > 192.168.10.11 ports 16888 > 16384, data
length 1478, DSCP 46

[ap6532-2] 4 11:09:01.743407 I UDP: 192.168.10.11 > 192.168.12.129 ports 16888 > 16384, data
length 1478, DSCP 46
```

2.2.5 Use Case 5 – Client not Associating

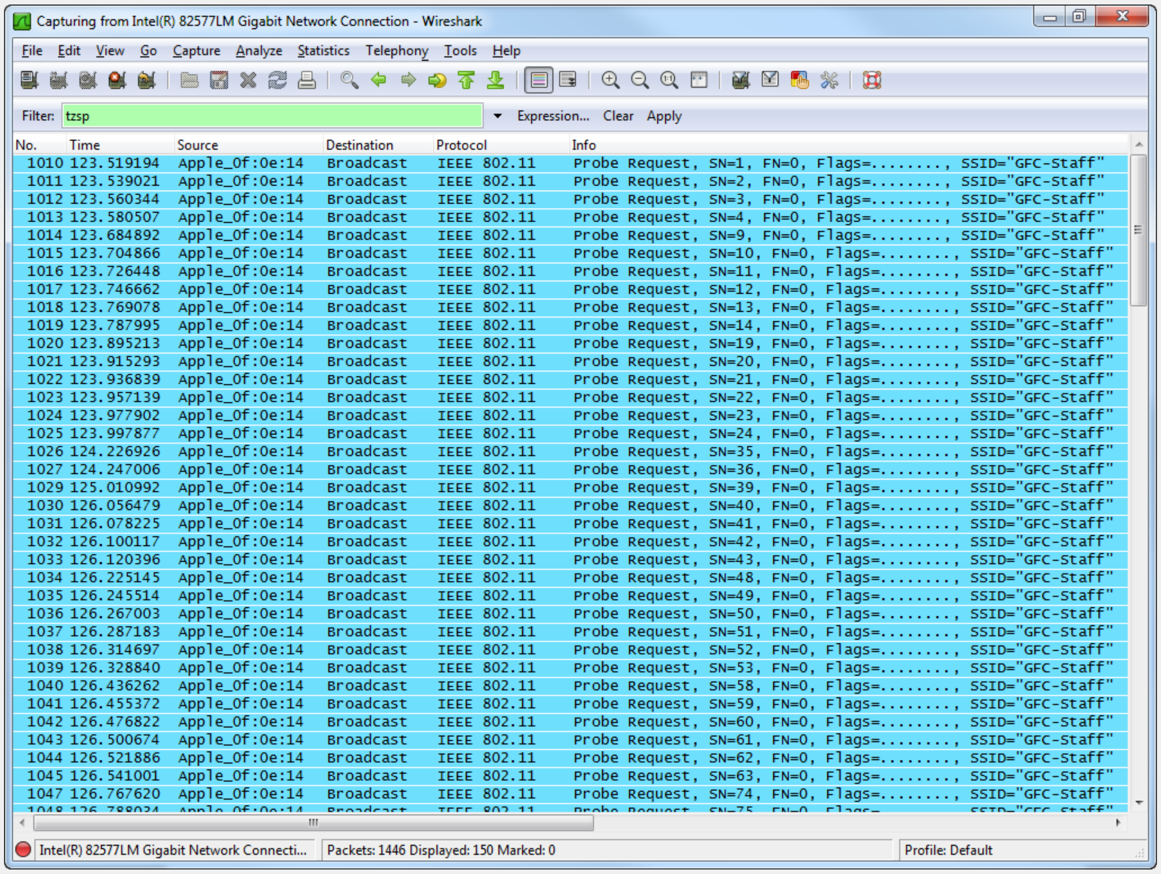
The following example demonstrates how to use a live packet capture to aid in remotely troubleshooting association issues. In this scenario an Apple device ***E0-F8-47-0F-0E-14*** is unable to connect to the corporate Wireless LAN in a remote store.

A live packet capture is first initiated on the stores RF Domain to determine if the device is attempting to associate. As the device supports operation in both 2.4 GHz and 5.0 GHz bands a filter is applied to capture 802.11 frames from the device on all radios. In addition the live packet capture is streamed using TZSP to a centralized host ***192.168.10.101*** running Wireshark for full decoding.

The live packet capture shows that the wireless client is probing for the SSID named GFC-Staff which is not a valid SSID for the store. As the device is not probing for any store networks it is determined that the issue is client related and the trouble ticket is forwarded to the desktop team who can re-configure the device:

Example:

```
RFSX000# remote-debug live-pktcap rf-domain store100 write tzsp 192.168.10.101 radio all filter
ether host E0-F8-47-0F-0E-14
```


Capturing from Intel(R) 82577LM Gigabit Network Connection - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: tzsp
Expression... Clear Apply

```
No.  Time        Source          Destination  Protocol     Info
1010 123.519194  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=1, FN=0, Flags=........, SSID="GFC-Staff"
1011 123.539021  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=2, FN=0, Flags=........, SSID="GFC-Staff"
1012 123.560344  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=3, FN=0, Flags=........, SSID="GFC-Staff"
1013 123.580507  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=4, FN=0, Flags=........, SSID="GFC-Staff"
1014 123.684892  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=9, FN=0, Flags=........, SSID="GFC-Staff"
1015 123.704866  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=10, FN=0, Flags=........, SSID="GFC-Staff"
1016 123.726448  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=11, FN=0, Flags=........, SSID="GFC-Staff"
1017 123.746662  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=12, FN=0, Flags=........, SSID="GFC-Staff"
1018 123.769078  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=13, FN=0, Flags=........, SSID="GFC-Staff"
1019 123.787995  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=14, FN=0, Flags=........, SSID="GFC-Staff"
1020 123.895213  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=19, FN=0, Flags=........, SSID="GFC-Staff"
1021 123.915293  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=20, FN=0, Flags=........, SSID="GFC-Staff"
1022 123.936839  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=21, FN=0, Flags=........, SSID="GFC-Staff"
1023 123.957139  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=22, FN=0, Flags=........, SSID="GFC-Staff"
1024 123.977902  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=23, FN=0, Flags=........, SSID="GFC-Staff"
1025 123.997877  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=24, FN=0, Flags=........, SSID="GFC-Staff"
1026 124.226926  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=35, FN=0, Flags=........, SSID="GFC-Staff"
1027 124.247006  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=36, FN=0, Flags=........, SSID="GFC-Staff"
1029 125.010992  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=39, FN=0, Flags=........, SSID="GFC-Staff"
1030 126.056479  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=40, FN=0, Flags=........, SSID="GFC-Staff"
1031 126.078225  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=41, FN=0, Flags=........, SSID="GFC-Staff"
1032 126.100117  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=42, FN=0, Flags=........, SSID="GFC-Staff"
1033 126.120396  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=43, FN=0, Flags=........, SSID="GFC-Staff"
1034 126.225145  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=48, FN=0, Flags=........, SSID="GFC-Staff"
1035 126.245514  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=49, FN=0, Flags=........, SSID="GFC-Staff"
1036 126.267003  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=50, FN=0, Flags=........, SSID="GFC-Staff"
1037 126.287183  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=51, FN=0, Flags=........, SSID="GFC-Staff"
1038 126.314697  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=52, FN=0, Flags=........, SSID="GFC-Staff"
1039 126.328840  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=53, FN=0, Flags=........, SSID="GFC-Staff"
1040 126.436262  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=58, FN=0, Flags=........, SSID="GFC-Staff"
1041 126.455372  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=59, FN=0, Flags=........, SSID="GFC-Staff"
1042 126.476822  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=60, FN=0, Flags=........, SSID="GFC-Staff"
1043 126.500674  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=61, FN=0, Flags=........, SSID="GFC-Staff"
1044 126.521886  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=62, FN=0, Flags=........, SSID="GFC-Staff"
1045 126.541001  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=63, FN=0, Flags=........, SSID="GFC-Staff"
1047 126.767620  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=74, FN=0, Flags=........, SSID="GFC-Staff"
1048 126.788034  Apple_0f:0e:14  Broadcast    IEEE 802.11  Probe Request, SN=75, FN=0, Flags=........, SSID="GFC-Staff"
```

Intel(R) 82577LM Gigabit Network Connecti...
Packets: 1446 Displayed: 150 Marked: 0
Profile: Default

2.2.6 Use Case 6 – Client Not Authenticating

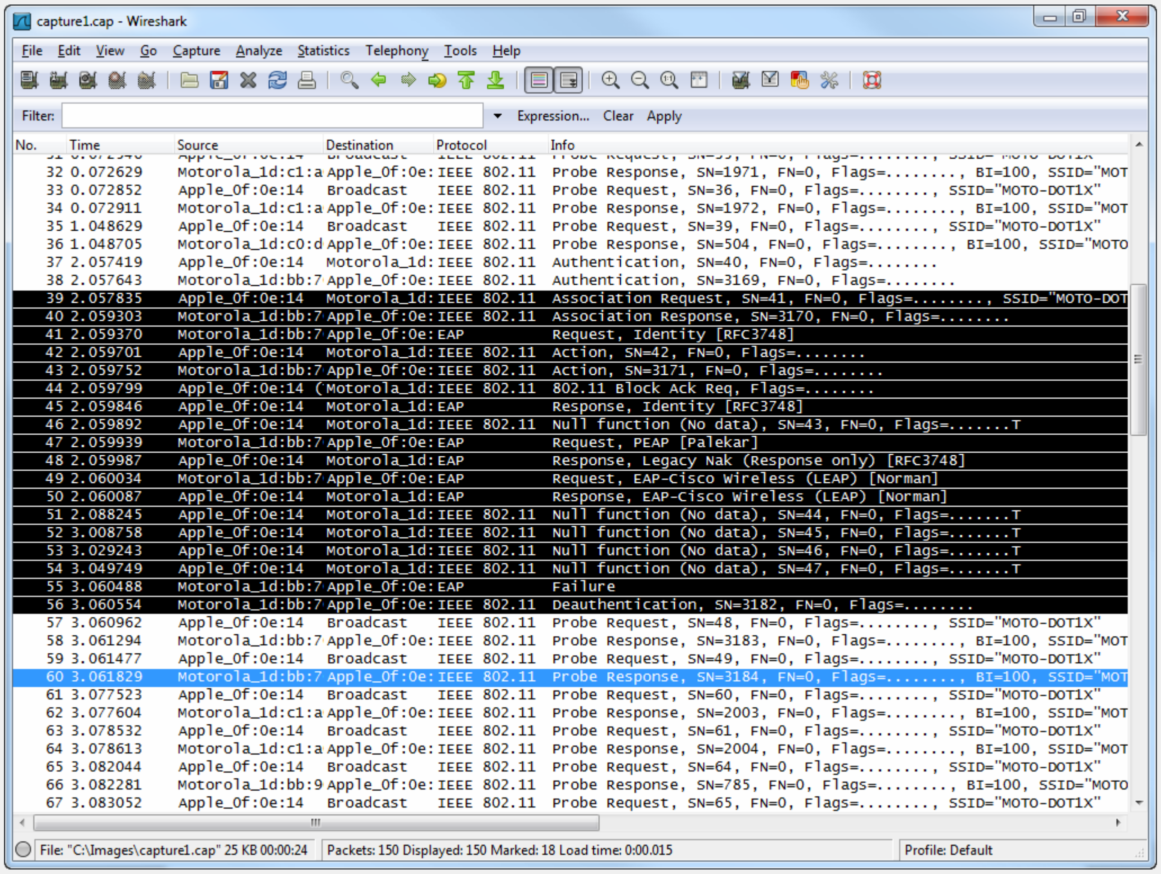
The following example demonstrates how to use a live packet capture to aid in remotely troubleshooting client authentication issues. In this scenario an Apple device ***E0-F8-47-0F-0E-14*** is unable to authenticate using LEAP to the corporate Wireless LAN in a remote store.

A live packet capture is first initiated on the stores RF Domain to monitor the EAP exchange between the Access Point and the client. As the corporate Wireless LAN only operates on the 5.0 GHz band a filter is applied to capture 802.11 frames from the device on 5.0 GHz radios. In addition the live packet capture exported to a centralized FTP server ***192.168.10.101*** for offline analysis.

The live packet capture shows that the wireless client is associating to the corporate Wireless LAN, performs the EAP handshake but is failing LEAP authentication. Further analysis of the RADIUS server logs show that the user's password was invalid which resulted in the authentication failure:

Example:

```
RFSX000# ap6532-1#remote-debug live-pktcap rf-domain store100 write
ftp://ftpuser:hellomoto@192.168.10.101/capture1.cap radio 2 filter ether host E0-F8-47-0F-0E-14
```



capture1.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
No.  Time     Source           Destination    Protocol     Info
  32 0.072629 Motorola_1d:c1:a Apple_0f:0e: IEEE 802.11  Probe Response, SN=1971, FN=0, Flags=........, BI=100, SSID="MOT
  33 0.072852 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=36, FN=0, Flags=........, SSID="MOTO-DOT1X"
  34 0.072911 Motorola_1d:c1:a Apple_0f:0e: IEEE 802.11  Probe Response, SN=1972, FN=0, Flags=........, BI=100, SSID="MOT
  35 1.048629 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=39, FN=0, Flags=........, SSID="MOTO-DOT1X"
  36 1.048705 Motorola_1d:c0:d Apple_0f:0e: IEEE 802.11  Probe Response, SN=504, FN=0, Flags=........, BI=100, SSID="MOTO
  37 2.057419 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Authentication, SN=40, FN=0, Flags=........
  38 2.057643 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Authentication, SN=3169, FN=0, Flags=........
  39 2.057835 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Association Request, SN=41, FN=0, Flags=........, SSID="MOTO-DOT
  40 2.059303 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Association Response, SN=3170, FN=0, Flags=........
  41 2.059370 Motorola_1d:bb:7 Apple_0f:0e: EAP          Request, Identity [RFC3748]
  42 2.059701 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Action, SN=42, FN=0, Flags=........
  43 2.059752 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Action, SN=3171, FN=0, Flags=........
  44 2.059799 Apple_0f:0e:14 ( Motorola_1d: IEEE 802.11  802.11 Block Ack Req, Flags=........
  45 2.059846 Apple_0f:0e:14   Motorola_1d: EAP          Response, Identity [RFC3748]
  46 2.059892 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Null function (No data), SN=43, FN=0, Flags=.......T
  47 2.059939 Motorola_1d:bb:7 Apple_0f:0e: EAP          Request, PEAP [Palekar]
  48 2.059987 Apple_0f:0e:14   Motorola_1d: EAP          Response, Legacy Nak (Response only) [RFC3748]
  49 2.060034 Motorola_1d:bb:7 Apple_0f:0e: EAP          Request, EAP-Cisco Wireless (LEAP) [Norman]
  50 2.060087 Apple_0f:0e:14   Motorola_1d: EAP          Response, EAP-Cisco Wireless (LEAP) [Norman]
  51 2.088245 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Null function (No data), SN=44, FN=0, Flags=.......T
  52 3.008758 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Null function (No data), SN=45, FN=0, Flags=.......T
  53 3.029243 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Null function (No data), SN=46, FN=0, Flags=.......T
  54 3.049749 Apple_0f:0e:14   Motorola_1d: IEEE 802.11  Null function (No data), SN=47, FN=0, Flags=.......T
  55 3.060488 Motorola_1d:bb:7 Apple_0f:0e: EAP          Failure
  56 3.060554 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Deauthentication, SN=3182, FN=0, Flags=........
  57 3.060962 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=48, FN=0, Flags=........, SSID="MOTO-DOT1X"
  58 3.061294 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Probe Response, SN=3183, FN=0, Flags=........, BI=100, SSID="MOT
  59 3.061477 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=49, FN=0, Flags=........, SSID="MOTO-DOT1X"
  60 3.061829 Motorola_1d:bb:7 Apple_0f:0e: IEEE 802.11  Probe Response, SN=3184, FN=0, Flags=........, BI=100, SSID="MOT
  61 3.077523 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=60, FN=0, Flags=........, SSID="MOTO-DOT1X"
  62 3.077604 Motorola_1d:c1:a Apple_0f:0e: IEEE 802.11  Probe Response, SN=2003, FN=0, Flags=........, BI=100, SSID="MOT
  63 3.078532 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=61, FN=0, Flags=........, SSID="MOTO-DOT1X"
  64 3.078613 Motorola_1d:c1:a Apple_0f:0e: IEEE 802.11  Probe Response, SN=2004, FN=0, Flags=........, BI=100, SSID="MOT
  65 3.082044 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=64, FN=0, Flags=........, SSID="MOTO-DOT1X"
  66 3.082281 Motorola_1d:bb:9 Apple_0f:0e: IEEE 802.11  Probe Response, SN=785, FN=0, Flags=........, BI=100, SSID="MOTO
  67 3.083052 Apple_0f:0e:14   Broadcast    IEEE 802.11  Probe Request, SN=65, FN=0, Flags=........, SSID="MOTO-DOT1X"
```

```
File: "C:\Images\capture1.cap" 25 KB 00:00:24   Packets: 150 Displayed: 150 Marked: 18 Load time: 0:00.015   Profile: Default
```

3. Wireless Debugging

Each WiNG 5.x provides a sophisticated wireless debugging facility which can capture specific wireless debugging events. The wireless debugging feature allows administrators to enable debugging on one or more remote WiNG 5.x devices or RF Domains and centrally view the wireless debugging events in real-time.

The main benefit of the wireless debugging feature is that it provides administrators with full visibility into remote clients as they associate, authenticate, re-authenticate and roam throughout the remote site without having to enable debugging on each individual Access Point and correlate the logs.

3.1 CLI Command Structure

The following provides the command structure for the ***remote-debug wireless*** command:

Command Structure:

```
remote-debug wireless [Scope] [Presentation] [Clients] [Max-Events] [Duration] [Events]
```

3.1.1 Scope

Wireless debugging can be enabled on one or more WiNG 5.x hosts or all WiNG 5.x devices within a specified RF Domain.

When a wireless debugging is enabled one or more WiNG 5.x hosts, wireless debugging event logs will only be captured by the specified WiNG 5.x hosts. It is possible that capturing event logs on a limited scope of devices may miss the interesting events as wireless hosts roam to Access Points not defined within the scope.

Command Syntax:

```
remote-debug wireless hosts <host1> <host2> .. [Presentation] [Clients] [Max-Events] [Duration]
[Events]
```

When an RF Domain is specified, wireless debugging event logs will be captured by all the devices within the specified RF Domain. This allows wireless debug events to be captured from all Access Points within a remote site providing full visibility in to the interesting events as wireless hosts roam within the site.

Command Syntax:

```
remote-debug wireless rf-domain <rf-domain-name> [Presentation] [Clients] [Max-Events] [Duration]
[Events]
```

3.1.2 Presentation

Wireless debugging evens can be viewed in real-time on the centralized Controllers console, streamed to an external syslog server or saved to an external FTP server for offline viewing.

3.1.2.1 Controller Console

By default when no syslog or external FTP server is specified, wireless debug events are displayed in real-time on the centralized Controllers console where the wireless debugging is initiated. Filtered events will be displayed on the centralized Controllers console as soon as the filtered wireless debugging event is captured.

The following example shows wireless debug events on a centralized Controllers console from a client associating to an 802.11N radio:

Example:

```
[ap6532-3] 14:25:53.322: mgmt:rx auth-req from E0-F8-47-0F-0E-14 on radio 0 (mgmt.c:1805)

[ap6532-3] 14:25:53.322: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14 on radio 0. status: success (mgmt.c:635)

[ap6532-3] 14:25:53.325: mgmt:rx association-req from E0-F8-47-0F-0E-14 on radio ap6532-3:R1 (mgmt.c:1786)

[ap6532-3] 14:25:53.325: mgmt:tx association-rsp success to E0-F8-47-0F-0E-14 on wlan (MOTO-PSK) (ssid:MOTO-
PSK) (mgmt.c:1516)
```

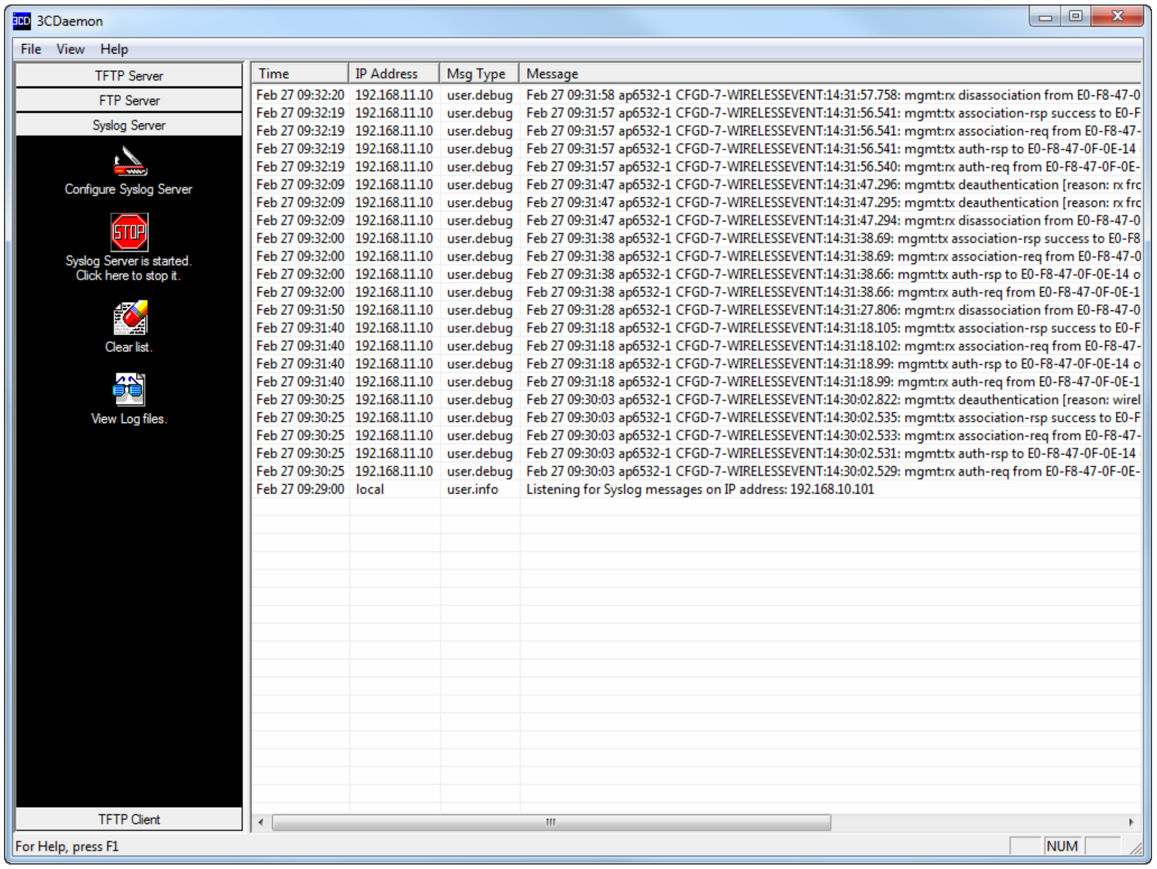
3.1.2.2 Syslog Server

Wireless debugging events can be forwarded in real-time to a centralized syslog server for archiving and viewing. Filtered wireless debugging events are forwarded from the centralized Controller to the external syslog server in real-time as the wireless debugging events are generated.

Command Syntax:

```
remote-debug wireless [Scope] write syslog <ip-address> [Clients] [Max-Events] [Duration]
[Events]

       <ip-address> - IP Address of the external Syslog server
```



```
Time             IP Address     Msg Type    Message
Feb 27 09:32:20  192.168.11.10  user.debug  Feb 27 09:31:58 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:57.758: mgmt:rx disassociation from E0-F8-47-0
Feb 27 09:32:19  192.168.11.10  user.debug  Feb 27 09:31:57 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:56.541: mgmt:tx association-rsp success to E0-F
Feb 27 09:32:19  192.168.11.10  user.debug  Feb 27 09:31:57 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:56.541: mgmt:rx association-req from E0-F8-47-
Feb 27 09:32:19  192.168.11.10  user.debug  Feb 27 09:31:57 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:56.541: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14
Feb 27 09:32:19  192.168.11.10  user.debug  Feb 27 09:31:57 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:56.540: mgmt:rx auth-req from E0-F8-47-0F-0E-
Feb 27 09:32:09  192.168.11.10  user.debug  Feb 27 09:31:47 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:47.296: mgmt:tx deauthentication [reason: rx frc
Feb 27 09:32:09  192.168.11.10  user.debug  Feb 27 09:31:47 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:47.295: mgmt:tx deauthentication [reason: rx frc
Feb 27 09:32:09  192.168.11.10  user.debug  Feb 27 09:31:47 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:47.294: mgmt:rx disassociation from E0-F8-47-0
Feb 27 09:32:00  192.168.11.10  user.debug  Feb 27 09:31:38 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:38.69: mgmt:tx association-rsp success to E0-F8
Feb 27 09:32:00  192.168.11.10  user.debug  Feb 27 09:31:38 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:38.69: mgmt:rx association-req from E0-F8-47-0
Feb 27 09:32:00  192.168.11.10  user.debug  Feb 27 09:31:38 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:38.66: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14 o
Feb 27 09:32:00  192.168.11.10  user.debug  Feb 27 09:31:38 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:38.66: mgmt:rx auth-req from E0-F8-47-0F-0E-1
Feb 27 09:31:50  192.168.11.10  user.debug  Feb 27 09:31:28 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:27.806: mgmt:rx disassociation from E0-F8-47-0
Feb 27 09:31:40  192.168.11.10  user.debug  Feb 27 09:31:18 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:18.105: mgmt:tx association-rsp success to E0-F
Feb 27 09:31:40  192.168.11.10  user.debug  Feb 27 09:31:18 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:18.102: mgmt:rx association-req from E0-F8-47-
Feb 27 09:31:40  192.168.11.10  user.debug  Feb 27 09:31:18 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:18.99: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14 o
Feb 27 09:31:40  192.168.11.10  user.debug  Feb 27 09:31:18 ap6532-1 CFGD-7-WIRELESSEVENT:14:31:18.99: mgmt:rx auth-req from E0-F8-47-0F-0E-1
Feb 27 09:30:25  192.168.11.10  user.debug  Feb 27 09:30:03 ap6532-1 CFGD-7-WIRELESSEVENT:14:30:02.822: mgmt:tx deauthentication [reason: wirel
Feb 27 09:30:25  192.168.11.10  user.debug  Feb 27 09:30:03 ap6532-1 CFGD-7-WIRELESSEVENT:14:30:02.535: mgmt:tx association-rsp success to E0-F
Feb 27 09:30:25  192.168.11.10  user.debug  Feb 27 09:30:03 ap6532-1 CFGD-7-WIRELESSEVENT:14:30:02.533: mgmt:rx association-req from E0-F8-47-
Feb 27 09:30:25  192.168.11.10  user.debug  Feb 27 09:30:03 ap6532-1 CFGD-7-WIRELESSEVENT:14:30:02.531: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14
Feb 27 09:30:25  192.168.11.10  user.debug  Feb 27 09:30:03 ap6532-1 CFGD-7-WIRELESSEVENT:14:30:02.529: mgmt:rx auth-req from E0-F8-47-0F-0E-
Feb 27 09:29:00  local          user.info   Listening for Syslog messages on IP address: 192.168.10.101
```

Figure 3.1.2.2 – Syslog Server

3.1.2.3 FTP / TFTP Server

Wireless debugging events can be captured and exported to an external FTP or TFTP server for offline viewing. When an FTP or TFTP server is specified, the filtered wireless debugging event data is uploaded to the external server as soon as all the wireless debugging events have been captured. The wireless debugging events are not steamed in real-time and are only available once the max number of events has been reached or the duration interval has expired.

FTP Command Syntax:

```
remote-debug wireless [Scope] write ftp://<user>|<password>@<hostname|IP>/<path>/<filename>
[Clients] [Max-Events] [Duration] [Events]

        <user> - FTP Username
        <password> - FTP Password
        <hostname|IP> - Hostname or IP Address of the FTP server
        /<path>/ - Optional pathname
        <filename> - Packet capture filename
```

TFTP Command Syntax:

```
remote-debug wireless [Scope] write tftp://<hostname|IP>/<path>/<filename> [Clients] [Max-Events] [Duration] [Events]

        <hostname|IP> - Hostname or IP Address of the FTP server
        /<path>/ - Optional pathname
        <filename> - Packet capture filename
```

3.1.3 Clients

Wireless debugging events can be captured for up to five specific wireless clients or all wireless clients within the defined scope. This allows Wireless LAN administrators to track wireless debug events from individual or groups of devices as well as all devices within the remote site.

Command Syntax:

```
remote-debug wireless [Scope] [Presentation] clients <MAC-Address> <MAC-Address> .. | all [Max-Events] [Duration] [Events]

        <MAC-Address> - MAC Address of the wireless client aa-bb-cc-dd-ee-ff
        all – Logs related to all clients
```

*Note – The use of the **all** option should only be enabled if the scope of events being captured is limited. Enabling wireless debugging events for all devices and events is not recommended for large remote sites supporting a large number of wireless clients.*

3.1.4 Max Events / Duration

By default WiNG 5.x will capture up to 60 wireless debugging events within a 50 second duration interval. The number of events and the duration interval can be increased or decreased as required. WiNG 5.x can capture a minimum of 1 event and a maximum of 10,000 events:

Command Syntax:

```
remote-debug wireless [Scope] [Presentation] [Clients] max-events <1 – 10,000> [Duration] [Events]
```

By default WiNG 5.x will capture the specified number of events in a 50 second time interval. The duration time for the capturing wireless debugging events can be increased or decreased as required and is specified in seconds. WiNG 5.x can capture wireless debugging events for a minimum duration of 1 second to a maximum duration of 86,400 seconds:

Command Syntax:

```
remote-debug wireless [Scope] [Presentation] [Clients] [Max-Events] duration <1-86400> [Events]
```

When wireless event debugging has been initiated at a remote site, the amount of events captured is dependent on the maximum event and duration interval. The capture of wireless debugging events will stop if:

1. If the number of events to be captured is reached before the duration interval expires.
2. The duration interval expires before the number of specified events has been reached.

You will not be able to initiate a wireless debugging session until the number of specified events has been reached or the duration interval expires. If the current wireless debugging session has not expired, initiation of the new wireless debugging session will require that you issue the ***remote-debug end-session wireless*** command. This will stop the current wireless debugging session and allow a new wireless debugging session to be initiated.

3.1.5 Events

Each wireless debugging session can include one or more events which can be monitored. A wireless debugging session can include all wireless events or one or more specific wireless events. For example an administrator troubleshooting EAP authentication issue for a wireless client can simultaneously enable ***eap***, ***radius*** and **wpa-wpa2** events to provide full visibility into to the authentication and key exchange.

Event	Description
all	All debug messages
eap	EAP debug messages
management	802.11 management debug messages
migration	Flow migration debug messages (firewall)
radius	RADIUS debug messages
system	System internal debug messages
wpa-wpa2	WPA / WPA2 debug messages

Table 3.1.5 – Wireless Debug Events

All wireless debugging events are forwarded to the centralized Controller where the wireless debugging session is initiated. For performance it is recommended that the scope of events to be captured be limited. If for example all wireless debugging events for all devices at a remote site are enabled, the wireless debugging session will add a significant amount of CPU overhead on the centralized Controller processing the events. In addition the WAN bandwidth will be increased as the remote Access Points at the site forward the wireless debugging events to the centralized Controller

*Note – The use of the **all** option should only be enabled if the scope of events being captured is limited. Enabling wireless debugging events for all devices and events is not recommended for large remote sites supporting a large number of wireless clients.*

3.2 Use Cases / Examples

3.2.1 Use Case 1 – EAP & RADIUS Authentications

The ***eap*** and ***radius*** event filters can be enabled to display EAP events between the Access Points and wireless clients as well as RADIUS events between the Access Points and RADIUS authentication servers. The ***eap*** and ***radius*** event filters are useful for troubleshooting clients that are unable to authenticate to the Wireless LAN or experience issues re-authenticating during roaming.

The following wireless debug events show a wireless client ***E0-F8-47-0F-0E-14*** in an RF Domain named ***store110*** that successfully authenticates to a WPA2 EAP Wireless LAN named ***MOTO-DOT1X***. In this example the authenticator receives a ***RADIUS Accept*** from the RADIUS authentication server. Note that the WPA/WPA2 4-way handshake can also be displayed by including the ***wpa-wpa2*** event:

Example – Successful EAP Authentication:

```
RFSX000# remote-debug wireless rf-domain store110 clients E0-F8-47-0F-0E-14 events eap radius
```

```
[ap6532-2] 17:58:13.817: eap:no response from mu E0-F8-47-0F-0E-14 retrying eap-id-request. attempt #2
(eap.c:281)

[ap6532-2] 17:58:13.820: eap:sending eap-id-req to E0-F8-47-0F-0E-14 (eap.c:785)

[ap6532-2] 17:58:14.919: eap:sending eap-id-req to E0-F8-47-0F-0E-14 (eap.c:785)

[ap6532-2] 17:58:14.923: eap:rx eap id-response from E0-F8-47-0F-0E-14 (eap.c:536)

[ap6532-2] 17:58:14.925: radius:aaa-policy external user: kvbf73 mac: E0-F8-47-0F-0E-14 server_is_candidate: 1
0 0 0 0 0  (radius.c:2194)

[ap6532-2] 17:58:14.927: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:58:14.930: radius:rx access-challenge from radius server for E0-F8-47-0F-0E-14 (radius.c:1495)

[ap6532-2] 17:58:14.930: eap:sending eap-req [eap_type:25(peap)] to E0-F8-47-0F-0E-14 (eap.c:793)

[ap6532-2] 17:58:14.932: eap:rx eap pkt from E0-F8-47-0F-0E-14 (eap.c:558)

[ap6532-2] 17:58:14.933: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:58:14.935: radius:rx access-challenge from radius server for E0-F8-47-0F-0E-14 (radius.c:1495)

[ap6532-2] 17:58:14.936: eap:sending eap-req [eap_type:17(eap-cisco wireless)] to E0-F8-47-0F-0E-14
(eap.c:793)

[ap6532-2] 17:58:14.938: eap:rx eap pkt from E0-F8-47-0F-0E-14 (eap.c:558)

[ap6532-2] 17:58:14.938: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:58:14.939: radius:rx access-challenge from radius server for E0-F8-47-0F-0E-14 (radius.c:1495)

[ap6532-2] 17:58:14.939: eap:sending eap-req [eap_type:17(eap-cisco wireless)] to E0-F8-47-0F-0E-14
(eap.c:793)

[ap6532-2] 17:58:14.940: eap:rx eap pkt from E0-F8-47-0F-0E-14 (eap.c:558)

[ap6532-2] 17:58:14.940: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:58:14.941: radius:rx UserName kvbf73 for E0-F8-47-0F-0E-14 (radius.c:443)

[ap6532-2] 17:58:14.941: radius:rx access-accept for E0-F8-47-0F-0E-14 (radius.c:1365)

[ap6532-2] 17:58:14.941: radius:radius: updating interim acct timeout of E0-F8-47-0F-0E-14 to 1800 seconds
(radius.c:757)

[ap6532-2] 17:58:14.941: eap:sending eap-req [eap_type:17(eap-cisco wireless)] to E0-F8-47-0F-0E-14
(eap.c:793)
```

The following wireless debug events show a wireless client ***E0-F8-47-0F-0E-14*** in an RF Domain named ***store110*** that is unable to authenticate to the WPA2 EAP Wireless LAN named ***MOTO-DOT1X***. In this example the authenticator receives a ***RADIUS Reject*** from the RADIUS authentication server indicating invalid credentials, EAP method, deactivated account or expired user certificate:

Example – Unsuccessful EAP Authentication:

```
RFSX000# remote-debug wireless rf-domain store110 clients E0-F8-47-0F-0E-14 events eap radius
```

```
Printing upto 50 messages from each remote system for upto 300 seconds. Use Ctrl-C to abort

[ap6532-2] 17:56:18.187: eap:no response from mu E0-F8-47-0F-0E-14 retrying eap-id-request. attempt #2
(eap.c:281)

[ap6532-2] 17:56:18.187: eap:sending eap-id-req to E0-F8-47-0F-0E-14 (eap.c:785)

[ap6532-2] 17:56:21.188: eap:no response from mu E0-F8-47-0F-0E-14 retrying eap-id-request. attempt #3
(eap.c:281)

[ap6532-2] 17:56:21.188: eap:sending eap-id-req to E0-F8-47-0F-0E-14 (eap.c:785)

[ap6532-2] 17:56:21.722: eap:rx eap-start from E0-F8-47-0F-0E-14 (eap.c:502)

[ap6532-2] 17:56:21.722: eap:sending eap-id-req to E0-F8-47-0F-0E-14 (eap.c:785)

[ap6532-2] 17:56:21.723: eap:rx eap id-response from E0-F8-47-0F-0E-14 (eap.c:536)

[ap6532-2] 17:56:21.723: radius:aaa-policy external user: kvbf73 mac: E0-F8-47-0F-0E-14 server_is_candidate: 1
0 0 0 0 0  (radius.c:2194)

[ap6532-2] 17:56:21.724: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:56:21.725: radius:rx access-challenge from radius server for E0-F8-47-0F-0E-14 (radius.c:1495)

[ap6532-2] 17:56:21.725: eap:sending eap-req [eap_type:25(peap)] to E0-F8-47-0F-0E-14 (eap.c:793)

[ap6532-2] 17:56:21.726: eap:rx eap pkt from E0-F8-47-0F-0E-14 (eap.c:558)

[ap6532-2] 17:56:21.726: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:56:21.727: radius:rx access-challenge from radius server for E0-F8-47-0F-0E-14 (radius.c:1495)

[ap6532-2] 17:56:21.727: eap:sending eap-req [eap_type:17(eap-cisco wireless)] to E0-F8-47-0F-0E-14
(eap.c:793)

[ap6532-2] 17:56:21.728: eap:rx eap pkt from E0-F8-47-0F-0E-14 (eap.c:558)

[ap6532-2] 17:56:21.729: radius:access-req sent to 192.168.10.6:1812 (attempt 1) for E0-F8-47-0F-0E-14
(radius.c:1210)

[ap6532-2] 17:56:22.729: radius:rx access-reject for E0-F8-47-0F-0E-14 (radius.c:1444)
```

3.2.2 User Case 2 – WPA / WPA2

The ***wpa-wpa2*** event filters can be enabled to display WPA / WPA2 events between the Access Point and wireless client. The ***wpa-wpa2*** event filter is useful for troubleshooting 4-way handshake issues during initial association, re-authentications or roaming. The ***management*** event can also be added to provide additional visibility into association, authentication and roaming events allowing administrators to monitor the state of the client before the wpa-wpa2 event occurs.

The following wireless debug events show a wireless client ***E0-F8-47-0F-0E-14*** in an RF Domain named ***store110*** that successfully associates and authenticates to a WPA2-PSK Wireless LAN named ***MOTO-PSK***. In this example the wireless client is initially connecting to the Wireless LAN for the first time and WPA/WPA2 4-way handshake is successfully completed. Note that the wireless client is place into a ***DATA-READY*** state:

Example – Successful WPA / WPA2 4-Way Handshake:

```
RFSX000# remote-debug wireless rf-domain default clients E0-F8-47-0F-0E-14 events wpa-wpa2
management
```

```
[ap6532-3] 16:25:22. 5: mgmt:rx auth-req from E0-F8-47-0F-0E-14 on radio 0 (mgmt.c:1805)

[ap6532-3] 16:25:22. 5: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14 on radio 0. status: success (mgmt.c:635)

[ap6532-3] 16:25:22. 8: mgmt:rx association-req from E0-F8-47-0F-0E-14 on radio ap6532-3:R1 (mgmt.c:1786)

[ap6532-3] 16:25:22. 8: mgmt:tx association-rsp success to E0-F8-47-0F-0E-14 on wlan (MOTO-PSK) (ssid:MOTO-
PSK) (mgmt.c:1516)

[ap6532-3] 16:25:22. 9: wpa-wpa2:tx msg #1 to E0-F8-47-0F-0E-14 attempt: 1 (80211i.c:472)

[ap6532-3] 16:25:22.16: wpa-wpa2:rx msg #2 from mu E0-F8-47-0F-0E-14 (80211i.c:926)

[ap6532-3] 16:25:22.17: wpa-wpa2:tx msg #3 to E0-F8-47-0F-0E-14 attempt: 1 (80211i.c:675)

[ap6532-3] 16:25:22.23: wpa-wpa2:rx msg #4. WPA2-AES handshake done. E0-F8-47-0F-0E-14 DATA-READY
(80211i.c:902)
```

The following wireless debug events show a wireless client ***E0-F8-47-0F-0E-14*** in an RF Domain named ***store110*** that cannot connect to the WPA2-PSK Wireless LAN named ***MOTO-PSK***. In this example the WPA/WPA2 4-way handshake fails indicating an incorrect passphrase has been defined on the wireless client or Wireless LAN profile:

Example – Unsuccessful WPA / WPA2 4-Way Handshake:

```
RFSX000# remote-debug wireless rf-domain default clients E0-F8-47-0F-0E-14 events wpa-wpa2
management
```

```
[ap6532-3] 16:26:13.397: mgmt:rx auth-req from E0-F8-47-0F-0E-14 on radio 0 (mgmt.c:1805)

[ap6532-3] 16:26:13.399: mgmt:tx auth-rsp to E0-F8-47-0F-0E-14 on radio 0. status: success (mgmt.c:635)

[ap6532-3] 16:26:13.402: mgmt:rx association-req from E0-F8-47-0F-0E-14 on radio ap6532-3:R1 (mgmt.c:1786)

[ap6532-3] 16:26:13.404: mgmt:tx association-rsp success to E0-F8-47-0F-0E-14 on wlan (MOTO-PSK) (ssid:MOTO-
PSK) (mgmt.c:1516)

[ap6532-3] 16:26:13.417: wpa-wpa2:tx msg #1 to E0-F8-47-0F-0E-14 attempt: 1 (80211i.c:472)

[ap6532-3] 16:26:13.918: wpa-wpa2:dot11i timeout for E0-F8-47-0F-0E-14 (80211i.c:1166)

[ap6532-3] 16:26:13.920: wpa-wpa2:tx msg #1 to E0-F8-47-0F-0E-14 attempt: 2 (80211i.c:472)

[ap6532-3] 16:26:14.421: wpa-wpa2:dot11i timeout for E0-F8-47-0F-0E-14 (80211i.c:1166)

[ap6532-3] 16:26:14.421: mgmt:tx deauthentication [reason: dot11i 4way handshake timeout (code:15)] to E0-F8-
47-0F-0E-14 (mgmt.c:925)
```

3.2.3 Use Case 3 – Management

The ***management*** event filters can be enabled to display 802.11 management events as wireless clients associate, authenticate and roam throughout a site. The management event filter is useful for troubleshooting mobility and SMART client load-balancing issues as it allows administrators to follow wireless clients as they roam between Access Points at a site.

The following wireless debug events show an active wireless client ***00-21-6A-60-82-0C*** in an RF Domain named ***CA107-SJC*** as it roams throughout the remote site. Each time the wireless client roams the wireless debug events shows the Access Point the client has roamed to in addition to association and auth events. If client SMART client load-balancing is enabled, event messages will also display ignored association requests as wireless clients are steered to preferred radios or neighboring Access Points:

Example – Management Frames during Roaming:

```
RFSX000# remote-debug wireless rf-domain CA107-SJC clients 00-21-6A-60-82-0C max-events 1000
duration 600 events management
```

```
[AN-14-8644B4] 18:44:47.379: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1900)
[AN-14-8644B4] 18:44:47.380: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 1. status: success (mgmt.c:661)
[AN-14-8644B4] 18:44:47.381: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1887)
[AN-14-8644B4] 18:44:47.381: mgmt:tx association-rsp success to 00-21-6A-60-82-0C on wlan (STCWLB)
(ssid:stcwlb) (mgmt.c:1593)
[AN-08-41DDF0] 18:44:47.530: mgmt:moving client 00-21-6A-60-82-0C to hold (roam detected) (mgmt.c:1006)
[AN-11-8644B8] 18:45:25.59: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1900)
[AN-11-8644B8] 18:45:25.60: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 1. status: success (mgmt.c:661)
[AN-11-8644B8] 18:45:25.61: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1887)
[AN-11-8644B8] 18:45:25.61: mgmt:tx association-rsp success to 00-21-6A-60-82-0C on wlan (STCWLB)
(ssid:stcwlb) (mgmt.c:1593)
[AN-14-8644B4] 18:45:24.853: mgmt:moving client 00-21-6A-60-82-0C to hold (roam detected) (mgmt.c:1006)
[AN-13-9E5170] 18:45:45.958: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 0 (mgmt.c:1900)
[AN-13-9E5170] 18:45:45.959: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 0. status: success (mgmt.c:661)
[AN-13-9E5170] 18:45:45.967: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 0 (mgmt.c:1887)
[AN-13-9E5170] 18:45:45.967: mgmt:00-21-6A-60-82-0C ignoring association req on wlan [STCWLB], radio 0 is
overloaded (mgmt.c:1581)
[AN-13-9E5170] 18:45:46.967: mgmt:rx deauthentication from 00-21-6A-60-82-0C on radio 0 (mgmt.c:1906)
[AN-13-9E5170] 18:45:46.975: mgmt:rx deauthentication from 00-21-6A-60-82-0C on radio 0 (mgmt.c:1906)
[AN-02-41DC50] 18:45:46.954: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1900)
[AN-02-41DC50] 18:45:46.956: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 1. status: success (mgmt.c:661)
[AN-02-41DC50] 18:45:46.958: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1887)
[AN-11-8644B8] 18:45:47.85: mgmt:moving client 00-21-6A-60-82-0C to hold (roam detected) (mgmt.c:1006)
[AN-02-41DC50] 18:45:46.958: mgmt:tx association-rsp success to 00-21-6A-60-82-0C on wlan (STCWLB)
(ssid:stcwlb) (mgmt.c:1593)
[AN-16-86458C] 18:46:41.781: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1900)
[AN-16-86458C] 18:46:41.781: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 1. status: success (mgmt.c:661)
[AN-16-86458C] 18:46:41.782: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1887)
[AN-16-86458C] 18:46:41.783: mgmt:tx association-rsp success to 00-21-6A-60-82-0C on wlan (STCWLB)
(ssid:stcwlb) (mgmt.c:1593)
[AN-02-41DC50] 18:46:41.612: mgmt:moving client 00-21-6A-60-82-0C to hold (roam detected) (mgmt.c:1006)
[AN-15-864538] 18:47:07.972: mgmt:rx auth-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1900)
[AN-15-864538] 18:47:07.972: mgmt:tx auth-rsp to 00-21-6A-60-82-0C on radio 1. status: success (mgmt.c:661)
[AN-15-864538] 18:47:07.974: mgmt:rx re-association-req from 00-21-6A-60-82-0C on radio 1 (mgmt.c:1887)
```