WiNG 5 Feature Guide

Locationing API (RSSI Sensor Feed)

Published: April 2017

Extreme Networks, Inc.
Phone / +1 408.579.2800
Toll-free / +1 888.257.3000

www.extremenetworks.com

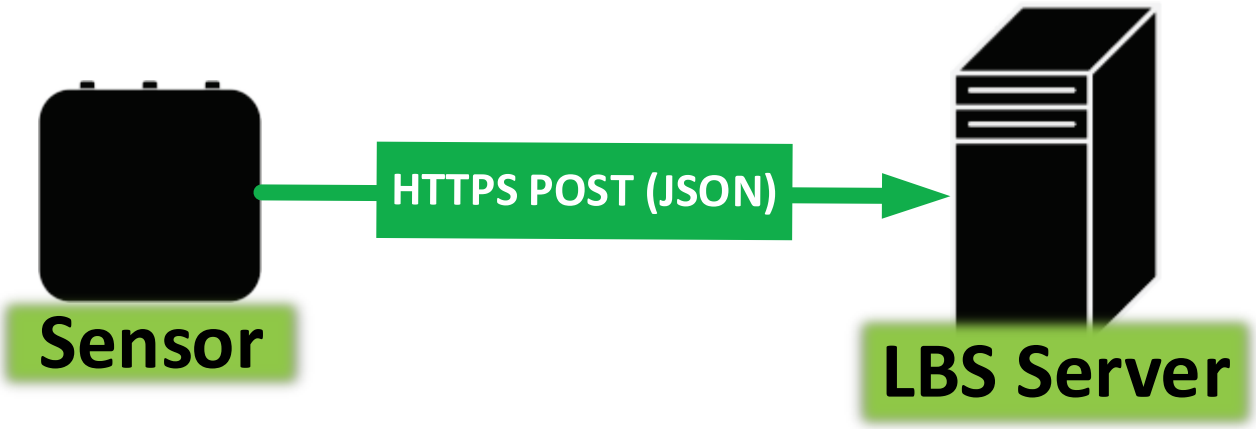
P/N XXXXX-XX

Contents

Introduction

Before WiNG 5.8.4 Access Point dedicated or radioshare sensors were exclusively used with ADSP to send RSSI data for wireless client location tracking. With ADSP an Access Point sends data in binary format over raw TCP socket. With WiNG 5.8.4 release an Access Points can now forward the RSSI data directly to a 3rd party Locationing Server over the raw HTTP/HTTPS connection, which makes possible to integrate with any 3rd party Locationing Systems. The integration in 5.8.4 release has been done with Euclid Analytics, but can be used with any other locationing system as well.

This document covers the configuration needed on the WiNG5 to integrate with a 3rd party Locationing System. This document also provides steps to create a sample subscriber on Debian server to listen to the HTTPS stream and forward this information to syslog file, which can be used for demonstration or test purposes.



Sensor Modes

To find the location of wifi devices, the sensors collect RSSI (signal strength) information of all wifi devices that it hears on air. Depending on the sensor configuration an Access Point can either listen on its current operating channel or use a dedicated sensor mode.

With the new RSSI Sensor Feed feature, WiNG5 sensors (radioshare or dedicated) will send the rssi information to the 3rd party Locationing System, which will be responsible for calculation of the location of wifi devices.

Currently all Access Point platforms that support dedicated or radioshare sensor functionality support RSSI Sensor Feed feature.

It is important to outline all the possible modes for sensor operation. This section provides an overview of supported sensing mechanisms on WiNG5 Access Points. Generally, two sensor modes are supported by WiNG5 APs with some minor deviations:

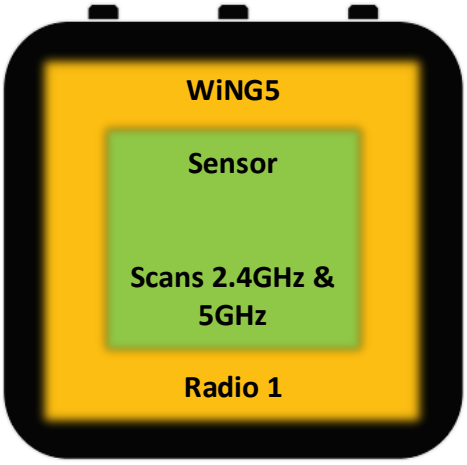
Dedicated Sensor – an AP with one of all of its radios set to sensor mode. Dedicated Sensor radio cannot provide data services for wireless clients.

RadioShare Sensor – Allows an AP radio to simultaneously provide data services for wireless clients, as well as send RSSI Feed to a Locationing Server. By default radioshare sensor scans only on the current operating channel. Optionally Off Channel Scanning can be enabled to allow radioshare sensor to scan on operating channels, additionally performing periodic off channel scan based on configured channel list.

Dedicated Sensor – Single Radio APs

For single radio APs one of the radios can be configured in sensor mode to scan both 2.4GHz and 5GHz bands. The channels that are scanned by the sensor radio are determined by the Sensor Policy in WiNG5.

Supported platforms: AP6521, AP6511

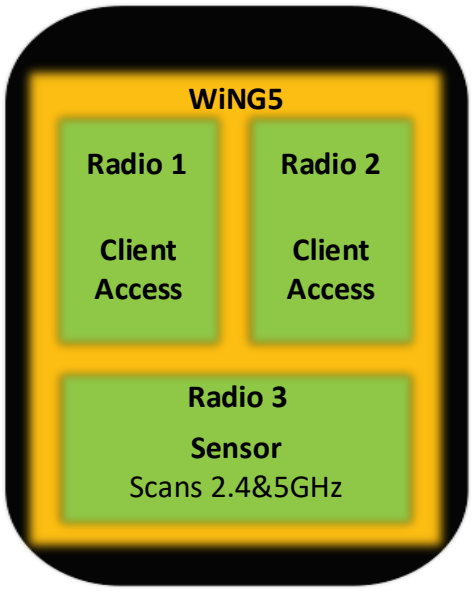


Dedicated Sensor on One Radio– Dual Radio APs

For dual radio APs with band unlocked radios one of the radios can be configured in sensor mode to scan both 2.4GHz and 5GHz bands, other radio will provide client data services on 2.4GHz or 5GHz. The channels that are scanned by the sensor radio are determined by the Sensor Policy in WiNG5.

This is useful in high density deployments, where there are too many 2.4GHz radios in one area, therefore one out of four 2.4GHz radios can be used as a dedicated sensor.

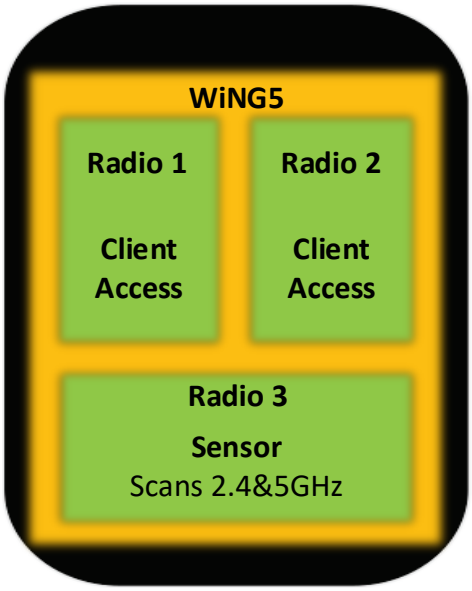
Supported platforms: AP6522, AP6532, AP7131, AP8122, AP8432



Dedicated Sensor – Tri Radio APs

For tri radio APs the third radio can be configured in sensor mode to scan both 2.4GHz and 5GHz bands, while Radio 1 and Radio 2 will provide client data services on 2.4GHz or 5GHz. The channels that are scanned by the sensor radio are determined by the Sensor Policy in WiNG5.

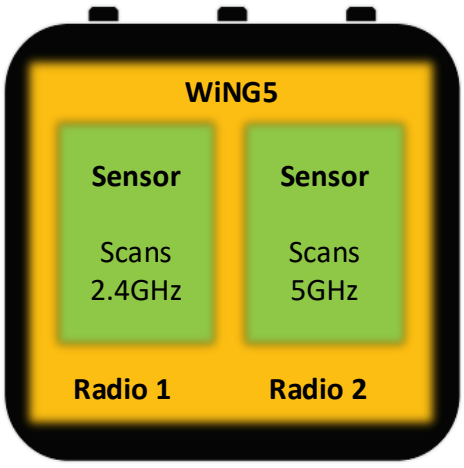
Supported platforms: AP7131N, AP8132 + USB Sensor Module, AP8232 + USB Sensor Module, AP7161, AP8163, AP8533



Dedicated Sensor – Both Radios

When both radios on a dual radio AP (band-locked or band-unlocked) are set to sensor mode, one radio will scan 2.4GHz and the other will scan 5GHz. The channels that are scanned are determined by the Sensor Policy in WiNG5.

Supported platforms: All, except AP6511, AP6521, AP7502.



Radioshare Inline Sensor

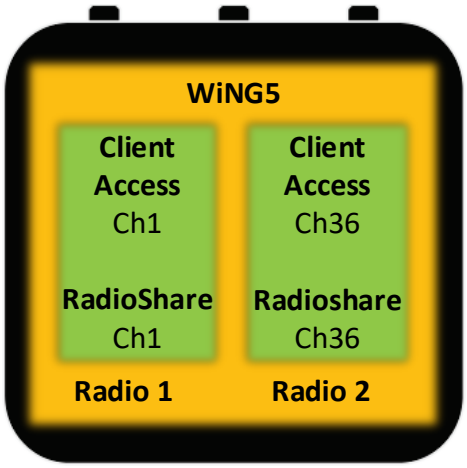
Radioshare Inline mode only listens for the frames directed to its BSSIDs, where most of them would be data frames from associated wireless clients. If there is a requirement to scan only for associated clients, then inline mode should be used.

Supported platforms: All, except AP7502, AP82XX

Radioshare Promiscuous Sensor

Radioshare sensor in promiscuous mode sends all frames it can hear on the current operating channel. This mode is typically used for Presence/Zone or Location Tracking.

Supported platforms: All, except AP7502, AP82XX

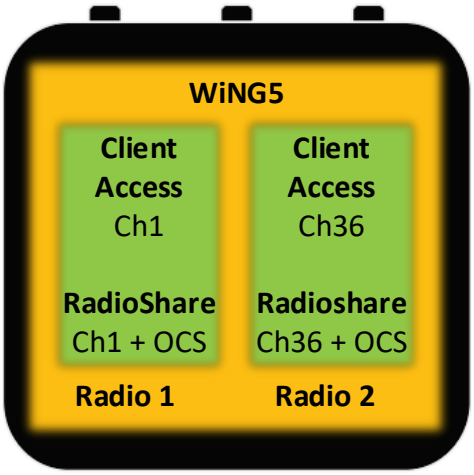


Radioshare Promiscouos Sensor with Off Channel Scanning

In certain cases, Off Channel Scanning (OCS) can be enabled on the RadioShare sensor to allow the radio to perform periodic scanning on other channels other than the operating one. It is important not to confuse radioshare OCS with SmartRF Off Channel Scanning, as they are using different processes. By default RadioShare sensor will go off channel every 20 DTIMs, which by default would equal to 40 beacons or 4096ms.

Note
RadioShare sensor with Off Channel Scanning cannot be used in combination with SmartRF. If SmartRF is enabled on the RF Domain then Off Channel Scanning settings won't take effect.

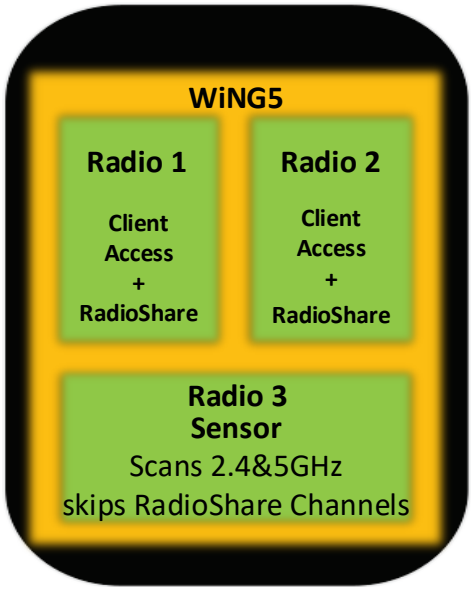
Supported platforms: AP7522, AP7532, AP7562



Mixed Dedicated and Radio Share Sensor

With tri radio Access Points like AP71XX, AP81XX or AP8533 an AP can have both dedicated and radioshare sensors enabled at the same time to increase the number of samples takes, thus improving overall location accuracy.

Supported platforms: AP7131N, AP7161, AP8132 with USB Sensor Module, AP8163, AP8533



Sensor – Supported AP Platforms:

AP Model	Dedicated Sensor	3rd Radio Dedicated Sensor	Radioshare Sensor
AP6511	☑	☒	☑
AP6521	☑	☒	☑
AP6522	☑	☒	☑
AP6532	☑	☒	☑
AP7131	☑	☒	☑
AP7131N	☑	☑	☑
AP7161	☑	☑	☑
AP8122	☑	☒	☑
AP8132	☑	☑*	☑
AP8163	☑	☑	☑
AP8222	☒	☒	☒
AP8232	☒	☑*	☒
AP7502	☒	☒	☒
AP7522	☑**	☒	☑
AP7532	☑**	☒	☑
AP7562	☑**	☒	☑
AP8432	☑***	☒	☑
AP8533	☒	☑	☑

*available with a USB Sensor module

**both radios must be set to dedicated sensor mode, because they are band-locked.

***only 1st radio can be used as a dedicated sensor

JSON data format:

Each RSSI Feed POST will contain information about detected devices (Wireless Client or Access Point) in the following JSON format:

JSON RSSI Feed Message Example:


```
{
"sq": 3,
"ht":
[
 { "si": "90:B6:86:42:2C:9F",
   "sm": "90B686",
   "bi": "00:00:00:00:00:00",
   "ap": 0,
   "ot": 1473246736,
   "ct": 1473246736,
   "sh": -90,
   "sl": -90,
   "ss": -90,
   "cn": 1
 },
 { "si": "60:A3:7D:D8:A7:F0",
   "sm": "60A37D",
   "bi": "28:28:5D:69:14:0A",
   "ap": 0,
   "ot": 1473246736,
   "ct": 1473246736,
   "sh": -77,
   "sl": -77,
   "ss": -77,
   "cn": 1
 },
 { "si": "60:D9:C7:9E:DC:E4",
   "sm": "60D9C7",
   "bi": "28:28:5D:69:14:0A",
   "ap": 0,
   "ot": 1473246736,
   "ct": 1473246736,
   "sh": -88,
   "sl": -88,
   "ss": -88,
   "cn": 1 }
],
"vs": 3,
"zver": 1,
"pf": 12,
"sn": "84:24:8D:86:45:84"
}
```

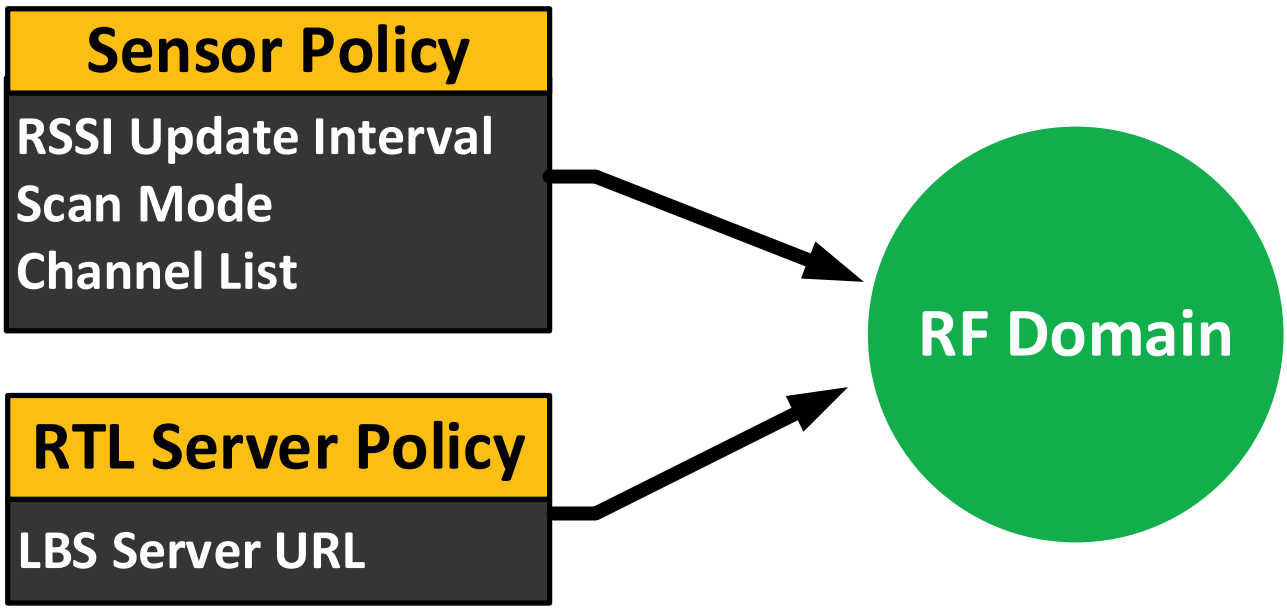

Key	Description	Value Format
sq	number of devices detected in the same JSON message	Integer
ht	Container with detected device information	
si	Detected device MAC address (AP or Client)	String in AA:BB:CC:DD:EE:FF format
sm	OUI of the detected device MAC	String in AABBCC format
bi	Associated BSSID (if device is associated or if the device is an AP)	String in AA:BB:CC:DD:EE:FF format
ap	The device is an Access Point	String 1 (yes) or 0 (no)
ot	Start of the observation window	String in UNIX epoch time stamp format
ct	End of the observation window	String in UNIX epoch time stamp format
sl	Minimum RSSI detected during the observation window	Integer in dBm
sh	Maximum RSSI detected during the observation window	Integer in dBm
ss	Average RSSI during the observation window	Integer in dBm
cn	Number of RSSI samples taken during the observation windows	Integer
vs	Version (default to 3).	Integer
zver	Zebra version. Reserved for future use.	Integer
pf	Euclid-assigned partner ID. It is hard set for 12 for Zebra.	Integer
sn	Sensor MAC Address (BASE MAC of the AP)	String in AA:BB:CC:DD:EE:FF format

Configuration

Components

The components of RSSI Sensor Feed are listed below:

- Sensor Policy – determines scanning parameters for dedicated sensors and location update intervals for both dedicated and radioshare sensors.
- RTL Server Policy – provides a URL of the Locationing Server.



Sensor Policy

Sensor Policy defines three main parameters of the sensor operation:

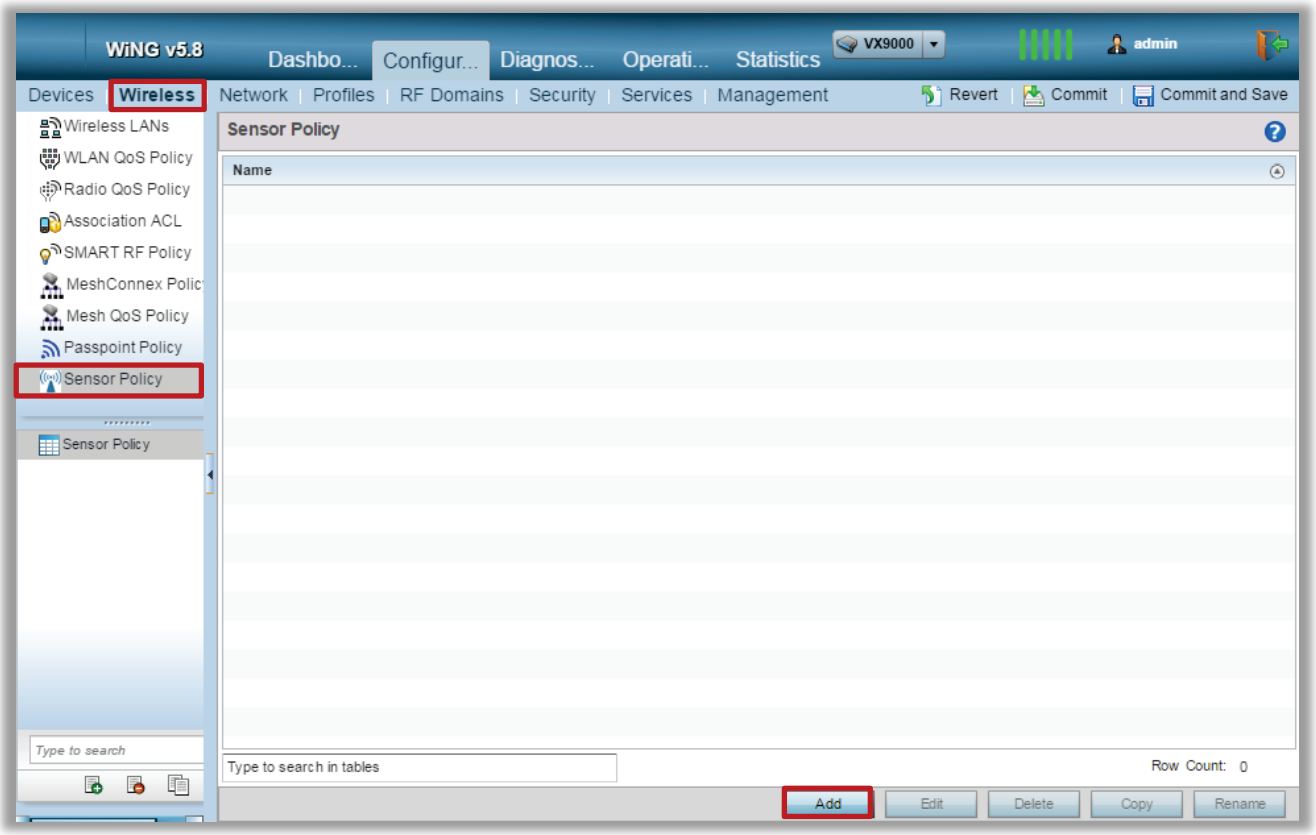
RSSI Scan Interval	This is the scan period sensors utilize for RSSI (signal strength) assessments. In other words, it is an update interval at which sensor sends samples to the locationing server. Default is 1 second.
Scan-mode	• Default-Scan – predefined list of channels to scan. Channel map: 1,6,11,36,40,44,48. • Custom-Scan – a list of channels in the 2.4GHz ,4.9GHz ,5GHz and 6GHz can be defined with appropriate channel width and scan weight. • Channel-Lock – instruct the radio to lock itself on specific 20MHz channel. Useful to increase scanning accuracy in certain scenarios.
Channel / Channel Width / Channel Weight	For Custom-Scan mode administrator can define a list of channels with appropriate channel width and scan weight to use. For Channel-Lock administrator can select a specific 20MHz channel to lock on.

Note
If a dedicated sensor is used with ADSP, sensor policy configuration will not take effect. In such scenario channel list and RSSI Scan Interval should be set by the ADSP server under “Sensor Operation” tab.

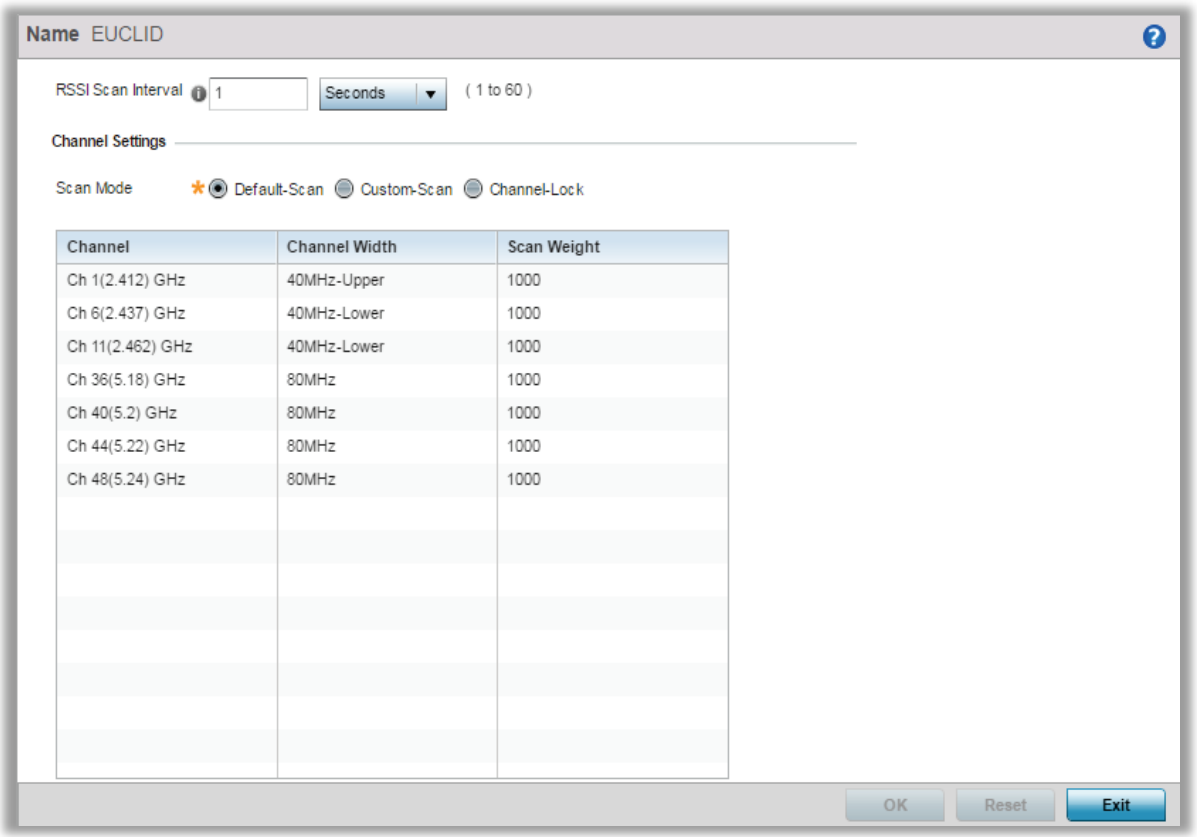
It is important to understand the difference between radioshare and dedicated sensors and which parameters under sensor policy will take effect:

- For RadioShare Sensors the only parameter that will take effect is “RSSI Scan Interval”. Channel Settings will be determined by current operating channel of the data radio and optionally off-channel-scan channel-list configured under radio interface.
- For Dedicated Sensors RSSI Scan Interval determines frequency of locationing data updates, while Channel Settings will define channel scan mode and list of channels to scan.

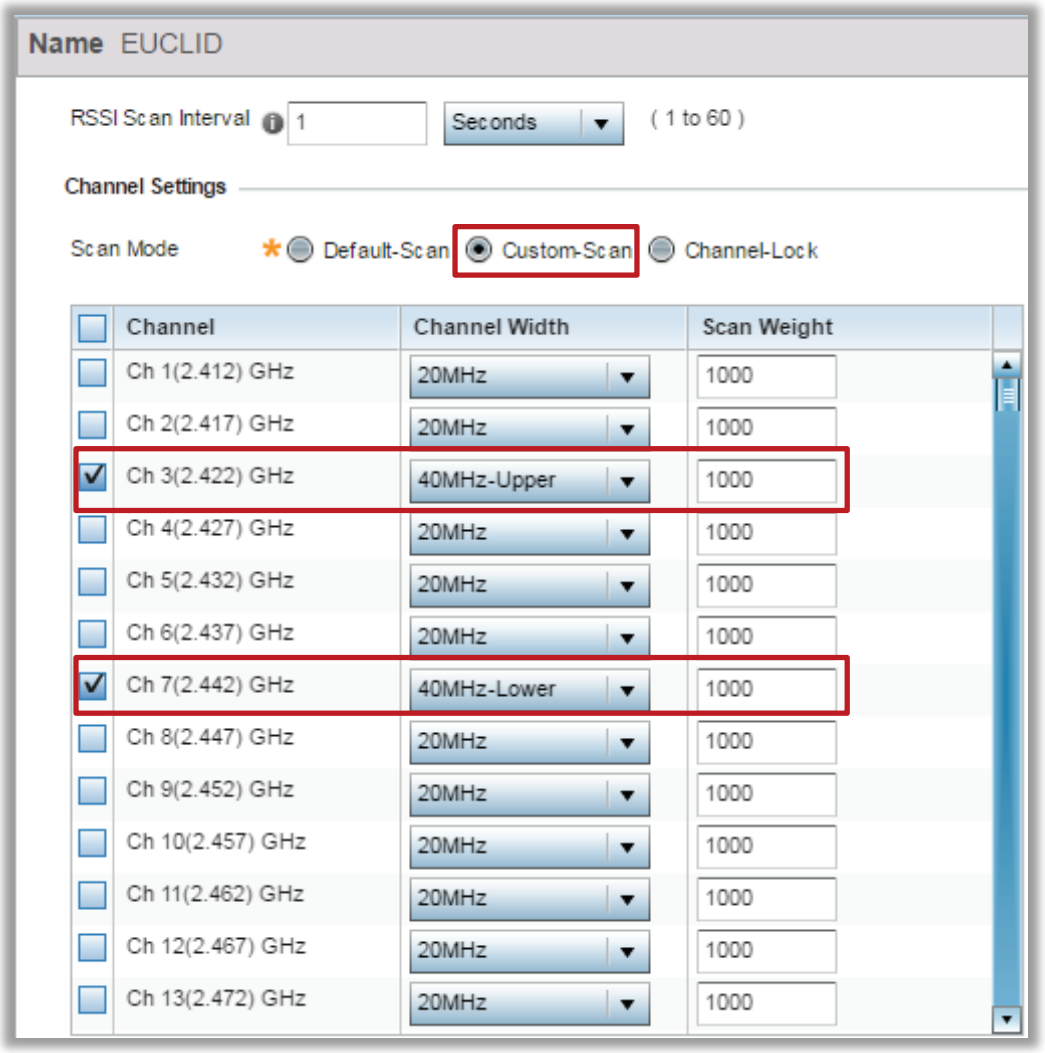
Sensor Policy Configuration – Web UI



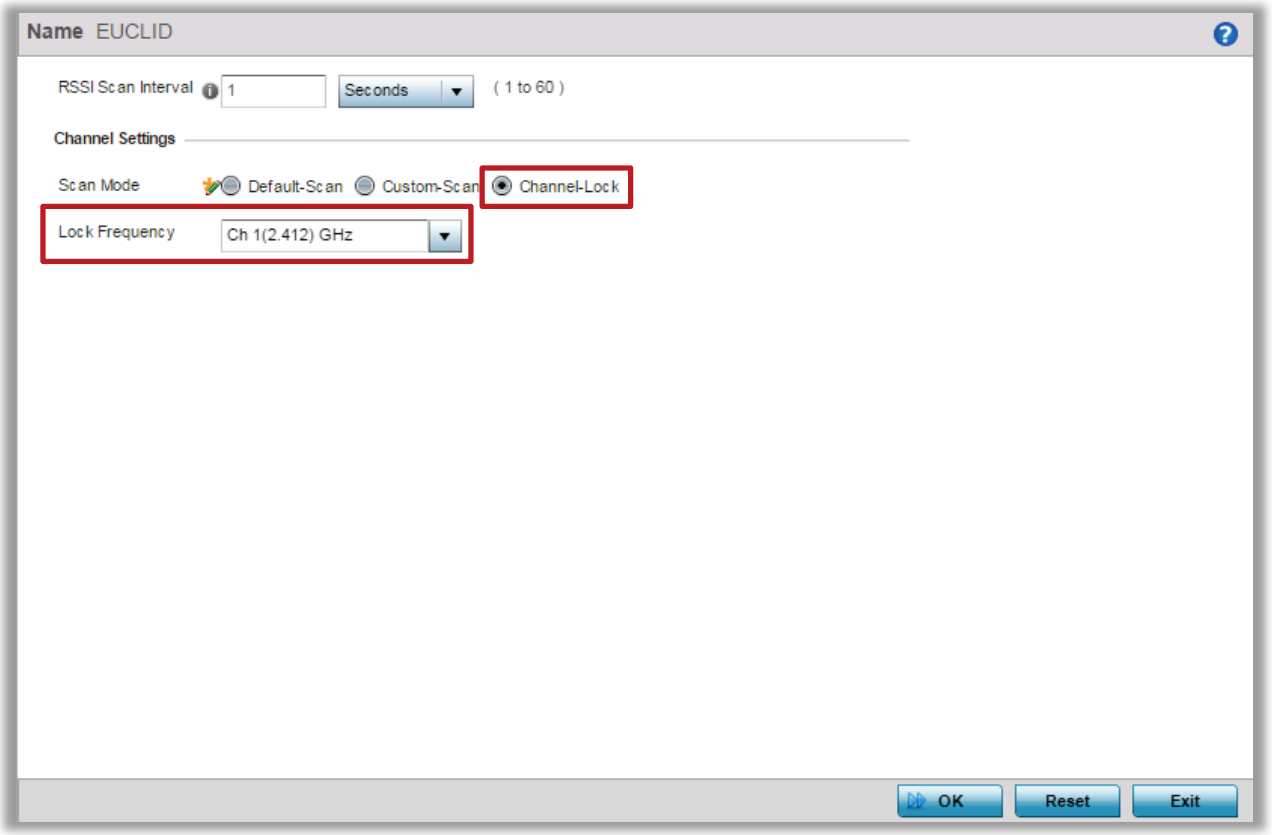
Option 1: Default Scan Pattern

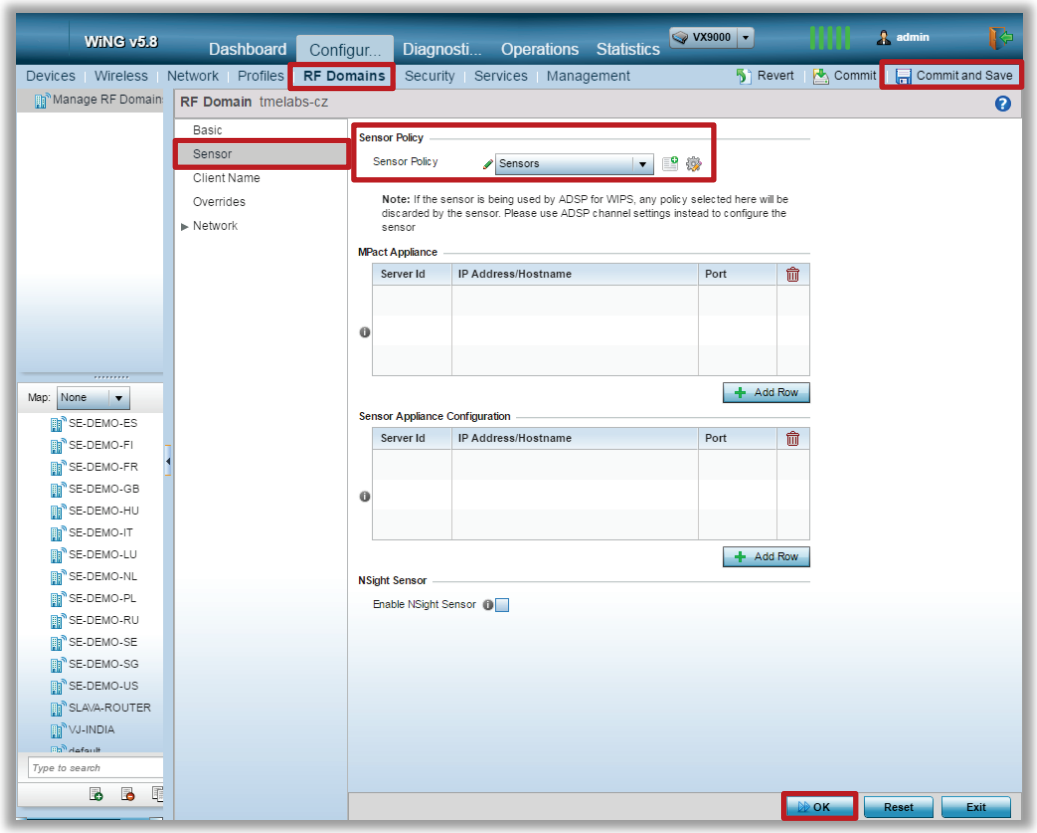


Option 2: Custom Scan Pattern



Option 3: Channel Lock





Sensor Policy Configuration – CLI

```
vx9000#conf t

vx9000(config)#sensor-policy Sensors
```

Option 1: Default Scan

```
sensor-policy Sensors
 rssi-interval-duration 1
 scan-mode Default-Scan
```

Option 2: Custom Scan

```
sensor-policy Sensors
 rssi-interval-duration 1
 scan-mode Custom-Scan
 custom-scan channel-frequency 5180 width 80MHz scan-weight 1000
 custom-scan channel-frequency 2442 width 80MHz scan-weight 1000
 custom-scan channel-frequency 2422 width 80MHz scan-weight 1000
```

Option 3: Channel Lock

```
sensor-policy Sensors
 rssi-interval-duration 1
 scan-mode Channel-Lock lock-freq 2442
```

Sensor Policy RF Domain Assignment

```
rf-domain tmelabs-cz
 timezone CET
 country-code cz
 use smart-rf-policy SMRT
 use sensor-policy Sensors
 control-vlan 1
```

RTL Server Policy

RTL Server Policy provides a URL to a sensor where it should POST RSSI information. The information is sent using HTTP POST method in JSON format. Configuration is available in CLI only.

For integration with Euclid Analytics the following URL format must be followed:

https://test.euclidanalytics.com/{partner name}/{client name}
Where:

{partner name} = zebra

{client name} = client identifier provided by Euclid Analytics that is tied to client's account.

For integration with other Locationing Systems parties refer to 3rd party documentation, URL format is dictated by a locationing server.

RTL Server Policy Configuration – CLI Only

```
VX-1#conf
Enter configuration commands, one per line.  End with CNTL/Z.

VX-1(config)#rtl-server-policy EUCLID
VX-1(config-rtl-server-policy-EUCLID)#url https://test.euclidporisms.net/zebra/testzebra
VX-1(config-rtl-server-policy-EUCLID)#exit

VX-1(config)#rf-domain <RF Domain Name>
VX-1(config-rf-domain-tmelabs-cz)#use rtl-server-policy EUCLID
VX-1(config-rf-domain-tmelabs-cz)#commit write
```

Sample Subscriber using Apache2 on Debian

This section provides instructions on how to quickly setup sample subscriber on a Debian 8 server that will listen for RSSI Feed info and write this information to syslog.

1. Update sources and upgrade the server

```
$sudo apt-get update
$sudo apt-get upgrade
```

2. Install Apache2 and all the dependencies

```
$su –
#apt-get install apache2 libtext-string-hexconvert-perl libjson-perl libapache2-mod-perl2
```

3. Enable CGI

```
#a2enmod cgi
Using vim or any other text editor edit /etc/apache2/sites-enabled/000-default.conf and add the following
lines before </VirtualHost>:
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin/">
   AllowedOverride None
   Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
   Order allow,deny
   Allow from all
</Directory>
```

4. Restart Apache

```
#service apache2 restart
```

5. Enable SSL and add SSL server certificate

```
#a2enmod ssl
Edit /etc/apache2/sites-enabled/000-default.conf. In this example the server will listen on port 5443:
LoadModule ssl_module modules/mod_ssl.so
Listen 5443
<VirtualHost *:5443>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName rssi-feed-consumer.zebranoc.com
        SSLEngine on
        SSLCertificateFile "/usr/certs/zebranoc.crt"
        SSLCertificateKeyFile "/usr/certs/zebranoc.key"
        SSLCipherSuite RC4-SHA:AES128-SHA:HIGH:!aNULL:!MD5
        SSLHonorCipherOrder on
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
  Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
  Order allow,deny
```

```
  Allow from all
</Directory>

</VirtualHost>
#service apache2 restart
```

6. Create Perl Script to listen for HTTPS stream from Access Points

```
#cd /usr/lib/cgi-bin/
#vim rssi-feed-subscriber.pl
#!/usr/bin/perl
use CGI;
use Sys::Syslog;
use Data::Dumper;
use String::HexConvert ':all';
use JSON;
my $q = new CGI;
print $q->header('text/html');
my $data = $q->param('POSTDATA');
syslog('info', '[data] =  '.$data.' ');
print "OK";
exit 0;
#chmod +x rssi-feed-subrscriber.pl
#./rssi-feed-subscriber.pl
```

7. Configure RTL Server Policy on the AP

```
VX-1#conf
Enter configuration commands, one per line.  End with CNTL/Z.
VX-1(config)#rtl-server-policy RSSI-FEED
VX-1(config-rtl-server-policy-RSSI-FEED)#url https://rssi-feed-consumer.zebranoc.com:5443/cgi-bin/rssi-feed-subscriber.pl
VX-1(config-rtl-server-policy-RSSI-FEED)#exit
VX-1(config)#rf-domain <RF Domain Name>
VX-1(config)#use rtl-server-policy RSSI-FEED
```

8. Monitor Incoming Data from the SYSLOG

```
#tail -f -n 10 /var/log/messages
Sep  7 20:29:30 ip-172-31-1-34 rssi-feed-subscriber.pl[13699]: [data] =  { "sq": 13, "ht": [ { "si":
"AC:CF:5C:80:28:06", "sm": "ACCF5C", "bi": "74:67:F7:64:99:83", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1 }, { "si": "10:A5:D0:48:EB:AB", "sm": "10A5D0", "bi":
"00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1
}, { "si": "5C:E0:C5:36:61:EF", "sm": "5CE0C5", "bi": "74:67:F7:64:B9:20", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -63, "sl": -63, "ss": -63, "cn": 1 }, { "si": "A0:A8:CD:8F:BF:88", "sm": "A0A8CD", "bi":
"74:67:F7:64:B3:A0", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -70, "sl": -70, "ss": -70, "cn": 1
}, { "si": "3C:A9:F4:3E:49:4C", "sm": "3CA9F4", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -82, "sl": -82, "ss": -82, "cn": 1 }, { "si": "28:B2:BD:B5:21:C4", "sm": "28B2BD", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -85, "sl": -85, "ss": -85, "cn": 1
}, { "si": "18:5E:0F:16:70:83", "sm": "185E0F", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -70, "sl": -70, "ss": -70, "cn": 1 }, { "si": "A4:4E:31:E2:7E:90", "sm": "A44E31", "bi":
"74:67:F7:64:99:83", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -82, "sl": -82, "ss": -82, "cn": 1
}, { "si": "18:5E:0F:16:71:0A", "sm": "185E0F", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1 }, { "si": "84:85:06:8D:AF:A2", "sm": "848506", "bi":
"00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1
}, { "si": "A4:D1:D2:94:2C:17", "sm": "A4D1D2", "bi": "00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -89, "sl": -89, "ss": -89, "cn": 1 }, { "si": "1C:99:4C:35:B0:ED", "sm": "1C994C", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1
}, { "si": "AA:A7:95:B1:59:F3", "sm": "AAA795", "bi": "AA:A7:95:B1:59:F3", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -82, "sl": -82, "ss": -82, "cn": 1 } ], "vs": 3, "zver": 1, "pf": 12, "sn":
"74:67:F7:07:05:A5" }
Sep  7 20:29:30 ip-172-31-1-34 rssi-feed-subscriber.pl[13701]: [data] =  { "sq": 20, "ht": [ { "si":
"90:B6:86:75:9E:28", "sm": "90B686", "bi": "00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1 }, { "si": "F0:24:75:6F:DF:85", "sm": "F02475", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -84, "sl": -84, "ss": -84, "cn": 1
}, { "si": "10:A5:D0:48:EB:AB", "sm": "10A5D0", "bi": "00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1 }, { "si": "5C:E0:C5:36:61:EF", "sm": "5CE0C5", "bi":
"74:67:F7:64:B9:20", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -58, "sl": -58, "ss": -58, "cn": 1
}, { "si": "98:FC:11:E6:C5:D7", "sm": "98FC11", "bi": "00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -78, "sl": -78, "ss": -78, "cn": 1 }, { "si": "A4:B8:05:6D:8C:DB", "sm": "A4B805", "bi":
"84:24:8D:30:C5:70", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -74, "sl": -74, "ss": -74, "cn": 1
}, { "si": "A4:B8:05:6D:8C:DB", "sm": "A4B805", "bi": "84:24:8D:30:C5:70", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1 }, { "si": "64:BC:0C:4C:B7:68", "sm": "64BC0C", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -80, "sl": -80, "ss": -80, "cn": 1
}, { "si": "3C:A9:F4:3E:49:4C", "sm": "3CA9F4", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -74, "sl": -74, "ss": -74, "cn": 1 }, { "si": "00:CD:FE:02:12:3C", "sm": "00CDFE", "bi":
"00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1
}, { "si": "28:B2:BD:B5:21:C4", "sm": "28B2BD", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
```

```
1473280164, "sh": -76, "sl": -76, "ss": -76, "cn": 1 }, { "si": "18:5E:0F:16:70:AB", "sm": "185E0F", "bi":
"00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1
}, { "si": "78:31:C1:1B:EB:CE", "sm": "7831C1", "bi": "74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -89, "sl": -89, "ss": -89, "cn": 1 }, { "si": "18:5E:0F:16:70:83", "sm": "185E0F", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -44, "sl": -44, "ss": -44, "cn": 1
}, { "si": "E8:50:8B:EF:AC:32", "sm": "E8508B", "bi": "84:24:8D:BA:4D:D2", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -85, "sl": -85, "ss": -85, "cn": 1 }, { "si": "18:5E:0F:16:71:0A", "sm": "185E0F", "bi":
"74:67:F7:64:99:80", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -67, "sl": -67, "ss": -67, "cn": 1
}, { "si": "96:06:92:C0:86:BE", "sm": "960692", "bi": "00:00:00:00:00:00", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -87, "sl": -87, "ss": -87, "cn": 1 }, { "si": "28:5A:EB:6E:BE:74", "sm": "285AEB", "bi":
"84:24:8D:30:DE:40", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1
}, { "si": "D8:CF:9C:16:AF:72", "sm": "D8CF9C", "bi": "84:24:8D:30:DE:40", "ap": 0, "ot": 1473280164, "ct":
1473280164, "sh": -90, "sl": -90, "ss": -90, "cn": 1 }, { "si": "78:C3:E9:BA:8F:DA", "sm": "78C3E9", "bi":
"2C:C5:D3:32:02:B8", "ap": 0, "ot": 1473280164, "ct": 1473280164, "sh": -77, "sl": -77, "ss": -77, "cn": 1
} ], "vs": 3, "zver": 1, "pf": 12, "sn": "74:67:F7:07:05:41" }
```

Troubleshooting

The easiest way to troubleshoot or verify RSSI Sensor Feed functionality is to enable logging on the Access Point to debug level.

Further granular debugging can be enabled for the wipsd module.

1. General Info Logging

```
7532-bqs-1#show logging

Logging module: enabled
    Aggregation time: 60 seconds
    Console logging: level warnings
    Monitor logging: disabled
    Buffered logging: level debugging
    Syslog logging: level warnings
        Facility: local7

Log Buffer (2264926 bytes):


Sep 08 05:56:44 2016: USER: wipsd: WIPS[ LOCN][Info] RSSI Feed POST is OK
Sep 08 05:54:44 2016: USER: wipsd: WIPS[ LOCN][Info] RSSI Feed POST is OK
```

2. WIPSD debugging

```
7532-bqs-1#debug wipsd LOCN  < Crit | Err | Warn | Info | Dbg1 | Dbg2 >

7532-bqs-1#show logging

Logging module: enabled
    Aggregation time: 60 seconds
    Console logging: level warnings
    Monitor logging: disabled
    Buffered logging: level debugging
    Syslog logging: level warnings
        Facility: local7

Log Buffer (2273237 bytes):


Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Info] RSSI Feed POST is OK
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] Location_client_ThreadFunc: Invalid loaction-server
ip. not connecting
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_location_server_ip: location-server port:0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_location_server_ip: location-server ip:
0x0/0.0.0.0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] Location_client_ThreadFunc: Begin thread loop
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] sending to Euclid directly
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] RTL URL to send to is https://rssi-feed-
consumer.zebranoc.com:5443/cgi-bin/adsp.pl
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] Inside send_to_euclid_server()
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] JSON message formed
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x8000  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x4000  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1000  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x400  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x100  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] count_wls: 2: sta-mac: 00:02:BE:91:C1:DE, freq: 2462
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]                          signal strength: -62
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1  value: 7013
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x8000  value: b38d
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x4000  value: 2484
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1000  value: 7a
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x400  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x100  value: 7c
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] count_wls: 1: sta-mac: 90:B6:86:42:2C:9F, freq: 5660
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]                          signal strength: -62
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1  value: 7013
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x8000  value: b38d
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x4000  value: 2484
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x1000  value: 0
```

```
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x400  value: 0
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1]             type:0x100  value: 7c
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] count_wls: 0: sta-mac: 90:B6:86:42:2C:9F, freq: 5660
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_json_msg_from_rssi_msg: 84:24:8D:86:45:84,
msg_len:116, num-records:3, flags:0xD501, record-size:32, tlv_count:6
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] Inside get_json_msg_from_rssi_msg()
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_client_threadfunc: Message being worked on
before forming JSON object, message len:116
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] put_qentry_to_free_list: 0x13db0c added to free-list.
now, free-list:0x13db0c, free-count:9
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] put_qentry_to_free_list: adding q_entry 0x13db0c to
free-list 0x147764
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] print_euclid_rssi_msg_to_euclid_svr_detail:
84:24:8D:86:45:84, msg_len:116, num-records:3, flags:0xD501, record-size:32, tlv_count:6
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_euclid_location_msg_to_send: copying to websocket
buffer, src:0x13db18, dst:0xfed00, len:116
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_euclid_location_msg_to_send, (next is NULL)
head:0x142938, tail:0x142938
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_euclid_location_msg_to_send: Entering
Sep 08 06:00:53 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_client_threadfunc: Begin thread loop
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] Comms_Mgr_Sensor_Rx: V5_MSG_DEVICE_SIGSTR_DYM msg,
Location-server[configured:0, connected:0]
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_msg_send: enqueued to tx_queue
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_msg_send: q_entry: 0x142938, copying to
q_entry's buffer: 0x142944
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] get_qentry_from_free_list:dequeued entry:0x142938, now
free count:8
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_msg_send: available:9
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] euclid_msg_send: frame:0x101d54, len:52
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] print_euclid_rssi_msg_to_euclid_svr_detail:
00:00:00:00:05:00, msg_len:52, num-records:1, flags:0xD501, record-size:32, tlv_count:6
Sep 08 06:00:52 2016: USER: wipsd: WIPS[ LOCN][Dbg1] V5_MSG_DEVICE_SIGSTR_DYM msg, Sending to Euclid server
```