Fundamentals of 802.11 Wireless Sniffing

Contents

Checklist for a successful capture
Mac OS X Wireless Sniffing Tools
airportd
airport utility
tcpdump
Wi-Fi Diagnostic
Airtool
Wireless Sniffing using Windows 7 with Netmon 3.4 (deprecated method)
Introduction
Wireless Sniffing using Cisco Lightweight Access Point (LAP) in Sniffer mode
Introduction
Configuration steps
1) WLC / AP side
2) Sniffer side: Wireshark
3) Sniffer side: OmniPeek
Wireless Sniffing using OmniPeek Remote Assistant (ORA)
Introduction
Supported Wireless Adapters and Drivers
Driver Installation for Linksys USB600N with Windows XP
Running OmniPeek Remote Assistant
Capture Settings
File Properties
Capture Control
Wireless Sniffing using Cisco Autonomous (IOS) AP
802.11 Sniffer Capture Analysis - Physical Layer
Intro: physical layer info in wireless packet captures
Wireless packet headers – examples
Mac OS X 10.7 Wireless Diagnostics (Broadcom adapter?)
OmniPeek 6.8 (Ralink USB adapter)
Netmon 3.4
Applying wireless files as Wireshark columns
802.11 Sniffer Capture Analysis -Wireshark filtering
Introduction
Wireshark Filtering-wlan
Objective
Prerequisites
Why do we need to capture wireless sniffer trace?
Why do we need to use wireless sniffer capture filter?
When to use DISPLAY FILTERS and CAPTURE FILTERS?
How to filter?
802.11 Sniffer Capture Analysis - Management Frames and Open Auth

Introduction

802.11 – Frames and open authentication

802.11 Client Authentication Process

Management Frames

Control Frames

Data Frames

References

802.11 Sniffer Capture Analysis - WPA/WPA2 with PSK or EAP

WPA-PSK(TKIP)

WPA2-PSK(AES/TKIP)

How to decrypt WPA2 AES data on Over the Air Packet Captures with Wireshark

Requirements:

Process

WPA/WPA2 Enterprise

WPA(TKIP)/WPA2(AES) with dot1x (EAP-TLS)

802.11 Sniffer Capture Analysis – Multicast

Introduction

Solution

IGMP Snooping on WLC

Guidelines for Using Multicast Mode

Configuring Multicast (Using Multicast-Multicast Mode)

On Wireless LAN Controller

Multicast configuration on Wired network

Packet Captures

Topology

MCAST Traffic Generator Tool

Windows Wireshark packet capture on the MCAST generator

Windows Netmon Capture on the Mcast packet generator

Wireshark Captures on the Wireless interface of the Wireless client

Netmon Capture on the Wireless interface of the Wireless Client

802.11 Sniffer Capture Analysis – Web Authentication

Introduction

Configuration Webauth
Configuration on the WLC
Here is the client Debug when the client tried connecting

Introduction

The process of collecting a good wireless sniffer trace, in order to analyze and troubleshoot 802.11 behavior, can be a difficult and time consuming operation. But there are a few things to bear in mind that will help simplify and speed up this process. With Wireless sniffing it helps to have an idea of what you are really trying to do - you are trying to capture the raw wireless frames from over the air, as seen by the wireless sniffing device itself.

Checklist for a successful capture

**Step 1:** Since the sniffing device, client device and AP are all using RF generating radios for transmission or reception, it helps to have your wireless sniffer close to your target device (the client machine). This will allow your sniffing device to capture a good approximation of what your client device is hearing over the air.

**Step 2:** Use a separate device to act as your wireless sniffer - you cannot take a good wireless sniffer trace if it is running on the device under test (the client machine you are trying to get a wireless trace of).

**Step 3:** Understand exactly what 802.11 Channel and Band your client device is using before setting up your capture. Lock your sniffer to the channel of interest - do not use the sniffer's "scan channels" mode! (With "scan channels", the sniffer will cycle from channel to channel every second or so - useful for a site survey or to find "rogues", but not when attempting to capture an 802.11 problem.)

Also bear in mind that your client device may roam to another AP which is on a different RF channel or Band, so you need to plan accordingly. Typically in the 802.11b/g (2.4GHz) environment, using a three channel sniffer may be required. This involves using 3 Wireless adapters on your sniffing device, with each one set to channel 1, 6 and 11. Using USB wireless adapters works best for this type of setup.

**Step 4:** If you are troubleshooting 5GHz, then the number of channels will dramatically increase. Since you might not have enough cards to capture all channels, it is a good practice for the test, to operate on not more than 4 channels on your surrounding Access Points.

**Step 5:** If you can reproduce the problem when a client roams from one channel to another, then a 2-channel sniff should suffice. If you have only a single channel sniffer available, then have it sniff the roamed-to channel.

**Step 6:** Always NTP sync your sniffers. The packet capture will need to be collated with debug captures, and with other wired and/or wireless captures. Having your timestamps even one second off will make the collation much more difficult.

**Step 7:** If you are capturing for a long period of time (hours), then configure your sniffer to cut a new capture file every 30MB or so. In order to avoid filling up your hard drive, you will want to put an upper limit on the number of files written.
Note: The Linksys USB600N does not reliably collect 11n packets with short guard interval. Missing 20% to 30% of short guard interval packets. If necessary the WLC configuration can be changed to only use the slower long guard interval. This should be only a temporary configuration change. The command is: `config 802.11 {a | b}11nsupport guard-interval {any | long}

**Sniffer Tools**

Wireless Sniffing using a Mac with OS X 10.6 and above

Wireless sniffing on the Mac works well, as Mac OS X has built in tools to capture a wireless trace. However, depending on what versions of OS X you are running, the commands may vary. This document covers OS X 10.6 through the latest version. Wi-Fi diagnostics is the preferred method in the latest macbooks. It is always good to remember that your macbook sniffer needs to be at least as capable as the client you are sniffing (sniffing an 802.11ac smartphone with an 802.11n macbook is not optimal)

**Mac OS X Wireless Sniffing Tools**

- `airportd` (10.6-10.8)
- `airport utility` (10.6 - 10.8)
- `tcpdump` (10.8)
- Wi-Fi Diagnostics (10.7->10.12)
- Wireshark (10.6 - 10.8)
- 3rd party tool : Airtool

**airportd**

If you are running OS X 10.6 (Snow Leopard) or above, then you can easily use the command line utility “`airportd`”. Use the following steps:

1. Use the “command” + “Space bar” key combo to bring up the search dialog box in the upper right top of the screen and type in the word “terminal”, this will search for the terminal application, select this application to run it. A terminal window will appear.
2. Once you have a terminal window open, you can run the follow command to capture a Wireless sniffer trace on RF channel 11 (802.11b/g):
   ```
sudo /usr/libexec/airportd en1 sniff 11
   ```

Some things to note:

- You will be prompted to enter in your account password for verification.
- You cannot specify the name of the capture file or where you will place the output.
- You will lose any wireless connectivity to your network while the capture is occurring.
- If you are using an Air, the wireless adapter is en0 rather than en1
- Once you are finished with the trace, hit “Cntl-C” to stop the trace and the utility will display the name and location of the capture file. The file format is your standard wireshark PCAP file that can be read on the MAC or Windows via Wireshark.
airport utility

The airport utility is not a sniffer program; however, it can provide interesting information about
the wireless LAN. Also, it has the ability to set the default wireless channel - which is crucial for
sniffer programs (tcpdump, Wireshark) that are themselves unable to set the channel

Note: because the path to the airport utility is so ugly, it may be a good idea to set a symbolic link
to it from a directory in the path, e.g.

```
# sudo ln -s
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport
/usr/sbin/airport
```

set the wireless channel

```
# sudo
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport --
channel=48
```

dump out info on the SSIDs/BSSIDs seen:

```
# sudo
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport -s

SSID BSSID RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
Test 00:24:97:89:cb:41 -53 11 Y -- WPA(PSK/TKIP/TKIP)
WPA2(PSK/AES/TKIP)
Test2 00:24:97:89:cb:40 -53 11 N -- WPA(PSK/TKIP/TKIP)
Guest 00:22:75:e6:73:df -64 6,-1 Y -- WPA(PSK/AES,TKIP/TKIP)
WPA2(PSK/AES,TKIP/TKIP)
```

detailed information on the current association:

```
# sudo
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport –l

agrCtlRSSI: -54
agrExtRSSI: 0
agrCtlNoise: -89
agrExtNoise: 0
state: running
op mode: station
lastTxRate: 300
maxRate: 300
lastAssocStatus: 0
802.11 auth: open
link auth: wpa2-psk
BSSID: 0:24:97:95:47:60
SSID: GuestNet
MCS: 15
channel: 36,1

tcpdump

Tcpdump is a command line utility shipped with OS X that can perform packet capture. (The tshark utility bundled with Wireshark is very similar.) To perform a wireless packet capture using tcpdump:

- first set the channel using the airport utility as shown above
- then perform a wireless packet capture, saving to a file. When done, type Control/C to exit.

Example:

bash-3.2# tcpdump -I -P -i en1 -w /tmp/channel-11.pcap
tcpdump: WARNING: en1: no IPv4 address assigned
tcpdump: listening on en1, link-type IEEE802_11_RADIO (802.11 plus radiotap header), capture size 65535 bytes
^C
897 packets captured
968 packets received by filter
0 packets dropped by kernel
bash-3.2#

Wi-Fi Diagnostic

The easiest capture method is to use the graphical program called Wi-Fi Diagnostics.

It can be accessed by holding the ALT key and clicking on the top-right wifi icon (the one where you typically select the SSID you want to connect to)
Click on the "Open Wireless diagnostics" option in the list.

It will bring a window that will run a default report on troubleshooting. This is typically NOT what you are interested in.

Keep that window opened and go on the menu bar on top of the screen. click "Window". You will see a list of other interesting tools (useful for site survey or signal analysis). In the scope of
wireless sniffer capture, we are interested in the "Sniffer" option, click on it.

You then simply have to choose the primary channel as well as channel width.

The sniffer capture will be saved either on the Desktop or in /var/tmp/ as of Mac Os Sierra.

**Airtool**

Some 3rd party tools also exist that will support many mac os x versions and will enhance the embedded sniffing features with easier options to choose channels. One example is Airtool: https://www.adriangranados.com/apps/airtool

**Wireless Sniffing using Windows 7 with Netmon 3.4 (deprecated method)**

**Introduction**

With Microsoft Network Monitor (Netmon) 3.4, you can now perform some decent 802.11a/b/g
(and maybe 11n) wireless sniffing in Windows 7, using your standard wireless adapter. The file saved from Netmon can be read by latest (1.5 and above) Wireshark, though not in OmniPeek. It is important to note that Netmon is not supported by Microsoft anymore and will most often not work properly on 11n and 11ac adapters (most frames missing).

Netmon 3.4 is supported with XP SP3; however, it does not support wireless sniffing when running XP. As to Vista, experience is mixed; a reliable source reports that wireless sniffing does work in 64-bit Vista on a Macbook with BCM43xx 1.0 adapter.

We have removed the Netmon detailed section of this document since it is deprecated and will not reliably capture 802.11ac frames.


**Wireless Sniffing using Cisco Lightweight Access Point (LAP) in Sniffer mode**

**Introduction**

You can use the Cisco WLC and LAPs in sniffer mode, in conjunction with a wired sniffer (best results with Wireshark. Omnipeek decrypts the protocol differently as of version 10).

A single wired sniffer can collect packets from multiple APs, so this method is very useful to run multi-channel traces. For static scenarios, if it’s possible to move the sniffer AP, this can be used as an effective alternative to other sniffing options.

For roaming scenarios, the sniffer APs are usually installed in the proximity of the APs the client roams through, and this will report the “point of view” of the static APs rather than the client.

In order to see the RF from the point of view of the client while roaming, a multi-channel wireless trace should be captured using a laptop with multiple Wireless NICs that will follow the test client.

**Configuration steps**

1) **WLC / AP side**

Here are the steps in order to collect a trace using a sniffer mode LAP

- Configure the AP in Sniffer mode:
The AP will reboot and it will not be able to serve clients.

- Once the AP has re-joined the WLC, configure the radio of the AP (802.11b/g/n or 802.11a/n): specify the sniffer IP address, select the channel, enable sniffing.

- The sniffer will receive the 802.11 traffic encapsulated using the airepeek protocol, from the WLC management IP address with source port UDP/5555 and destination UDP/5000.

2) Sniffer side: Wireshark

If using Wireshark to receive the traffic, (Note: you must use wireshark 1.6.8 or earlier, newer versions have this support broken and the packets will not be decoded correctly, for 802.11ac sniffing we even recommend running the latest wireshark as enhancements are constantly made to the decoding) follow the steps below:

- Set the capture options to receive only traffic on UDP/5555:
This filter is optional but strongly recommended as it excludes all the non-wireless related traffic from the capture. Consider that the WLC sends traffic to a UDP port there’s no application listening on the sniffer side; this results in having a ICMP port-unreachable response for each packet received from the WLC.

Although this is expected, the filter above helps to exclude also this traffic which is useless and so it can only cause the trace to be bigger and more difficult to read.

- Then, start the capture:

- The captured traffic has to be “decoded as..” PEEKREMOTE in order to be able to see the 802.11 traffic:
The 802.11 traffic will now be visible:

- The 802.11 traffic will now be visible:
  - User Datagram Protocol, Src Port: 5555, Dst Port: 5000
  - AiroPeek/OmniPeek encapsulated IEEE 802.11
  - 802.11 radio information
  - IEEE 802.11 QoS Data, Flags: .......TC
    - Type/Subtype: QoS Data (0x0028)
    - Frame Control Field: 0x8801
      .00 0000 0011 1100 = Duration: 60 microseconds
      Receiver address: Cisco_0a:cf:0a (00:42:5a:0a:cf:0a)
      Destination address: Cisco_5f:f7:ca (00:14:f1:5f:f7:ca)
      Transmitter address: MurataMa_78:36:89 (00:ae:fa:78:36:89)
      Source address: MurataMa_78:36:89 (00:ae:fa:78:36:89)
      BSS Id: Cisco_0a:cf:0a (00:42:5a:0a:cf:0a)
      STA address: MurataMa_78:36:89 (00:ae:fa:78:36:89)

The RF info shown above (e.g. the channel, signal strength, noise..) are added by the AP.

3) Sniffer side: OmniPeek

When using OmniPeek as the receiver of the traffic stream from the WLC/AP in sniffer mode, it’s first of all necessary to create a “Cisco Remote Adapter” under the “Adapter” menu of the “Capture Options” window:
At least one adapter is required; the name is a mandatory field, whereas the “IP Address” field can be left blank if you don’t want OmniPeek to filter the incoming traffic from a specific WLC.

In this case it’s not necessary to filter out any traffic (such as the ICMP port-unreachable) as OmniPeek will listen on the UDP port to specifically capture the data stream from the Wireless LAN Controller.

Before starting the capture, confirm the settings on the main OmniPeek window:

At this point the capture can be started and the result will be a trace including the RF info reported by the AP:
NOTE: By default OmniPeek remote adapter picks up the timestamp sent by the AP itself; this info has nothing to do with the AP clock, so the resulting timestamp will be incorrect. If you use a single sniffer AP the timestamps will be wrong but at least consistent; this is no longer true if you use multiple APs as sniffers (as every AP will send its own timestamp info, causing funky time jumps on the merged capture).

**Solution**

You can explicitly tell OmniPeek to use the local sniffer PC clock to set the packet timestamp.

This solves both the single and multi AP scenario, having correct and consistent timestamps as long as the PC running OmniPeek has a NTP-sync’d clock.

**How-to steps:**

In OmniPeek, do the following:

1. Go to Tools>Option>Analysis Modules

2. Search for cisco remote adapter then double click to bring out the options.

3. Tick on the Timestamp option then click OK and test again the capture.
Wireless Sniffing using OmniPeek Remote Assistant (ORA)

Introduction

Omnipeek Remote Assistant (ORA) - Cisco TAC can provide the Omnipeek Remote Assistant application to assist in performing wireless packet captures. The tool will capture wireless packets and encrypt them for processing by the TAC. A full version of Omnipeek Enterprise is required to decrypt and analyze the capture files.

You should receive a ZIP file from TAC – such as “ORADist_Default_7.0.zip” (the filename may change with different release versions). Unzip this file to some folder - to run ORA, simply launch OmniPeekRemoteAssistant.exe from that folder.

Supported Wireless Adapters and Drivers

Capturing Wireless Packets with ORA requires the use of supported Wireless Network Adapters along with the appropriate driver version. To view a complete list of supported adapters and drivers, please see:-

http://www.wildpackets.com/support/downloads/drivers

In most cases, the Ralink USB adapters will be the easiest to install - and, because you can install multiple USB adapters on a single laptop - they are the best way to get a multichannel capture.
The following Ralink adapters have been tested by Cisco TAC:

Linksys WUSB600N (V1 and V2), Linksys AE1000, ALFA AWUS051NH

**Driver Installation for Linksys USB600N with Windows XP**

**Step 1.** TAC can provide the OmniPeek driver for the Ralink USB adapters. You should receive a ZIP file “RALINKUSB-1_4_0_18.ZIP”. There will be 2 folders in the archive -- “Win2kXP” for 32-bit Windows and “WinXPx64” for 64-bit Windows. Extract the contents of the appropriate folder for your operating system to a specified location.

![Folders in WinZip File](image)

32-Bit ➔ Win2kXP

64-Bit ➔ WinXPx64

**Step 2.** Insert the Linksys USB600N adapter.

a. If this is the first time using the adapter on the workstation, Windows will start the New Hardware Wizard. Do not search for a driver automatically and click Next. Skip to step 3.

b. If you have previously installed the Linksys USB600N on your workstation, you will need to change the driver to the Omnipeek version. Go to Start > Control Panel > Network Connections and Right Click on the Linksys adapter and click Properties. In this example, the interface is “Wireless Network Connection 3”.

![Wireless Network Connection 3 Menu](image)

Under the General Tab, Click the “Configure…” button, and then click on the Driver Tab > Update Driver. This will prompt the Hardware Update Wizard.

**Step 3.** Select “Install from a list or specific location (Advanced)” and click Next. Select “Search for the best driver in these locations.”, include the location of your extracted driver files and click Next:-
Step 4. Windows will now search and install the Omnipeek driver. If you receive the following warning message, click “Continue Anyway”.
Step 5. The driver installation should complete and the adapter is now ready for capturing packets with ORA.

Running Omnipeek Remote Assistant

If the correct driver isn’t loaded, ORA may appear to work, but not provide the option to select the desired channel to monitor. The Channel cell will read ‘Ethernet’ or ‘Wireless’ and not offer the option to select a channel:

Capture Settings

Select the desired adapter(s) to perform the capture and indicate the desired channel. If you have multiple supported adapters installed you can capture on multiple channels simultaneously (but you cannot mix wired and wireless interfaces at the same time). You can select either an 802.11b/g channel or 802.11a channel in the dropdown. You can select 40 MHz 802.11n channels using the (n40l) or (n40h) options. The n40l will be the selected channel and adjacent lower channel, while n40h will be the selected channel and adjacent higher channel.
File Properties

Select the folder you would like to store the capture files in. You can then also specify the file rollover size. Each new filename will include a timestamp so data will not be overwritten.

Capture Control

If you have selected correct adapter/channel settings, you will now be able to click the Start/Stop buttons at the bottom. You will not be able to see the packets, but you will see the counters incrementing. Click Stop when finished.

Wireless Sniffing using Cisco Autonomous (IOS) AP

An autonomous AP can be used to gather air packet captures. Follow these instructions to perform the air capture.
1. Enter the dot11radio interface on which you wish to perform the capture. Set the station-role to sniffer, add the server/PC IP that will run Wireshark and collect the captures, and select the channel. The port you specify with the `monitor frames` command will be the destination UDP port to which the AP sends the captures.

   Step 1 `int {d0 | d1}` Enter the interface configuration command mode for configuring the radio interfaces.
   Step 2 `station-role sniffer` Changing the station role to sniffer.
   Step 3 `channel number` Selecting the channel in which to operate in sniffer mode.
   Step 4 `no shut` Reverses the shutdown of the interface.
   Step 5 `exit` Exits interface configuration command mode.
   Step 6 `sniffer ip-address destination-ip port port-number` Sets the IP address and port number, to which the AP will redirect all the packets. You can specify an IP address on any port number between 1024 to 65535.
   Step 7 `wireshark enable` If you are using Wireshark at the end point, this adds a Wireshark header to the packets.

Sample configuration:
```
ap(config)# int d0
ap(config-if)# station-role sniffer
ap(config-if)# channel 11
ap(config-if)# no shut
ap(config-if) # exit
ap(config)# sniffer ip-address 10.126.251.30 port 5555
ap(config)# wireshark enable
```

2. Start Wireshark on the server/PC. Navigate to Capture > Options. Select the Ethernet NIC (LAN) and add a filter to capture only traffic with the UDP port you specified in step 1.

   ![Wireshark Capture Options](image)

   3. Start the Wireshark capture.

   Uploading capture files to TAC Service
Request

If the capture file(s) are too large for email, you can upload them to your TAC Service Request:

https://tools.cisco.com/ServiceRequestTool/query/

Enter your SR Number, and then click on File Upload.

Sniffer Analysis

To analyze wireless captures, refer to the links below. They are designed to be read in order since each document will build upon the preceding document. Bear in mind that when reading any wireless trace, it's a good idea to understand the 802.11 Wireless specifications. While these documents will do a great job at helping you understand the packet flow and what to look for in a wireless trace, they are not meant to teach the 802.11 Wireless specifications.

802.11 Sniffer Capture Analysis - Physical Layer

Intro: physical layer info in wireless packet captures

A captured packet contains a copy of the frame data – but prepended to each frame is a metadata header, giving you information about how the frame was captured. With wired packets, the metadata isn’t much – the frame number, date when the packet was captured, the packet’s length. When doing wired packet analysis, you rarely care too much about the physical layer – with a bit error rate of $10^{-10}$, you usually assume that the captured bits are what they say they are.

Wireless is another story entirely – the physical layer is vastly more complex – and treacherous – than wired. Before diving into an attempt to analyze a capture based upon the upper layers, it is usually a good idea to get an understanding of the physical layer in which the capture was taken. If the physical layer isn’t working right – then the upper layers will never have a chance.

The following physical layer qualities are particularly important to be aware of:

- Signal strength (RSSI, “signal strength”, Signal/Noise Ratio.) It’s generally best to focus on RSSI, if available – i.e. the power level in dBm at which the sniffing adapter received the packet.
  - RSSI < -90 dBm: this signal is extremely weak, at the edge of what a receiver can receive
  - RSSI -67dBm: this is a fairly strong signal – the edge of what Cisco considers to be adequate to support Voice over WLAN.
  - RSSI > -55dBm: this is a very strong signal
  - RSSI > -30dBm: your sniffer is sitting right next to the transmitter

- Channel (frequency.) As a wireless LAN may support anywhere from 3 to 25 or so different channels, it’s crucial to know exactly which channel(s) your capture was taken from. If your intention is to get a sniff from a specific AP, then lock your sniff to that AP’s channel, and validate
that the capture was on that channel – otherwise, the capture will be worthless.

- Data rate – this can be anywhere from 1Mbps up to 300Mbps or more. To understand why data transmissions don’t always make it from transmitter to receiver, you must know what data rates are being used. A “marginal” RSSI of -80dBm may work horribly for a packet modulated at 54Mbps, but can be quite satisfactory at 6Mbps.

**Wireless packet headers – examples**

Different wireless sniffer may use different metadata header formats to encode the wireless physical layer. Do be aware that the accuracy of the information is dependent upon the specific adapter hardware and driver in use. Some values, such as noise, should generally be taken with a grain of salt.

Below are some samples, with the data rate, frequency and RSSI fields highlighted.

**Mac OS X 10.7 Wireless Diagnostics (Broadcom adapter?)**

OS X 10.7 uses a Radiotap v0 header, which looks like this in Wireshark:

![Wireshark screenshot of a Radiotap v0 header](image)

**OmniPeek 6.8 (Ralink USB adapter)**

In Wireshark, an OmniPeek capture uses an Airopeek header, which looks like this:

![Wireshark screenshot of an Airopeek header](image)
Note that Wireshark (as of 1.6.x) doesn’t know how to decode all the wireless metadata in an OmniPeek capture – the same frame viewed in OmniPeek itself shows Signal dBm, Noise Level and Noise dBm:

Netmon 3.4

Applying wireless files as Wireshark columns

It is very often much easier to understand what’s going on with a wireless sniff, if you have applied the wireless fields as columns. Here’s how to do this:

1. Locate the field of interest in the packet details section (first expanding the applicable header section if necessary) and right-click it. Select Apply as Column:
2. The new column appears. Now you can resize, rename (by right clicking the column header and selecting “Edit Column Details”), and move the column as desired.

3. Repeat for other columns of interest. Now you have a better handle on the physical layer aspects of your capture:

4. Once you’ve applied the new column, the next time you run Wireshark, the column will be available (if you don’t see it, right-click the column headers and select Displayed Columns.)

**802.11 Sniffer Capture Analysis - Wireshark filtering**

**Introduction**

802.11 Sniffer Capture Analysis - Wireshark filtering

**Wireshark Filtering-wlan**

**Objective**

This document will help you in guiding how to set up the wireshark and analyze the interesting packets using a versatile tool within the wireshark program called the wireshark filters.

**Prerequisites**
The wireshark tool in itself will not help us in getting through the troubleshooting unless we have a good knowledge and understanding of the protocol, the topology of the network and which data points to consider taking sniffer traces. This is true for whether its for a wired or for wireless network where we capture the packets over the air before they are put on the network. The stripping of the wireless mac address is done by the AP.

**Why do we need to capture wireless sniffer trace?**

When we inspect a traffic or data on a wired network using wired sniffer trace and cant find our interesting packets we need to know where is it missing. Our suspicion may get us to verify if it even made it through the first point of the source of origination which being wireless is working fine or not or it being missed over the air. If it did not make it correctly over the air then will obviously be missing or not get translated or sent over over to the wired side by the AP to the DS or distribution system. It then becomes critical for us identify and localize the wireless network issue using wireless sniffer trace.

**Why do we need to use wireless sniffer capture filter?**

When it comes to troubleshooting network related issues there are many dependencies and all work in layered model and each layer data depend on its lower layer below it. There are many components or network elements and configuration and proper operation of the devices helps us achieve a smooth running network. When a working network stops functioning a logical approach is required to localize the issue. Once identified still the exact point of failure is difficult to find. Those situations sniffer comes to our aid. Since this troubleshooting process can become so complicated despite using the best approach and having a good understanding and troubleshooting skills. The problem is that if you capture the packets traveling through a network device we may end having huge file and may even end up to 1G if you capture long enough with lot packets details in it. With the such a large amount of overwhelming data it may be very time-consuming to pin point the problem and gets practically a very difficult task almost tending to impossible.

Filtering comes to our rescue to help a good troubleshooting engineer to spot the problems quickly by eliminating the unwanted traffic cutting the variables to a few or minimum variables to focus on at one time. This will help in quickly finding whether the interesting traffic is present or absent from the traffic collected. Use of filters then becomes an art and complements the troubleshooters skill greatly.

It enhances the time to resolution rapidly hence the need to understand how to use the wireshark filtering.

- **DISPLAY FILTERS** – after you capture a lot of information, they help you to visualize only the packets that you are interested in

- **CAPTURE FILTERS** – from the beginning you know what the packet of interest for you and capture only those packets

Filters for coloring the packets- this is used as a visual aid to enhance the display filter or capture filter or can be used just without any filter to just classify the many interesting packets as different colors for high level approach.

**When to use DISPLAY FILTERS and CAPTURE FILTERS?**
It is recommended to use the Capture filters, when you know what are you looking for and trying to verify that in a running traffic to that event is captured when run that for more than couple of hours in a heavy traffic environment. This will help in keeping the data collected to stay in a reasonable amount in terms of file size.

If we are at a point we are not sure what might be causing the issue and is more of a behavioral random nature of problem then run the packet capture for less time within the probable window of problem occurrence pattern, like one or two hours, capture all the traffic and then use Display filters to visualize only the information that you are searching for.

Besides the use of above one can see all the capture and use coloring rules to catch the attention of certain type of packets assigned different colors for easy sorting or distinguishing packet flow.

How to filter?

Lets get a good understanding of the various fields within a typical wireshark sniffer trace. By breaking it down and defining each field.

We will be focusing on 3 items which we need to understand to start using Filtering.

- Capture filter
- Display Filter
- Coloring rules Filter

Before we delve in to details, here is the example of the sniffer capture window for wireshark, let dissect.

**MENU BAR**

[File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help]

This is the called the Menu bar of the wire shark window

It contains the following items:

- **File**: This menu contains items to open and merge capture files, save / print / export capture files in whole or in part, and to quit from Wireshark.

- **Edit**: This menu contains items to find a packet, time reference or mark one or more packets, handle configuration profiles, and set your preferences; (cut, copy, and paste are not presently implemented).

- **View**: This menu controls the display of the captured data, including colorization of packets, zooming the font, showing a packet in a separate window, expanding and collapsing trees in packet details

- **Go**: This menu contains items to go to a specific packet.

- **Capture**: This menu allows you to start and stop captures and to edit capture filters
• **Analyze** This menu contains items to manipulate display filters, enable or disable the dissection of protocols, configure user specified decodes and follow a TCP stream.

• **Statistics** This menu contains items to display various statistic windows, including a summary of the packets that have been captured, display protocol hierarchy statistics and much more.

• **Telephony** This menu contains items to display various telephony related statistic windows, including a media analysis, flow diagrams, display protocol hierarchy statistics and much more.

• **Tools** This menu contains various tools available in Wireshark, such as creating Firewall ACL Rules.

• **Internals** This menu contains items that show information about the internals of Wireshark.

• **Help** This menu contains items to help the user, e.g. access to some basic help, manual pages of the various command line tools, online access to some of the webpages, and the usual about dialog.

**The Main TOOL BAR**

![Main Toolbar](image)

The main toolbar provides quick access to frequently used items from the menu. This toolbar cannot be customized by the user, but it can be hidden using the View menu, if the space on the screen is needed to show even more packet data. As in the menu, only the items useful in the current program state will be available. The others will be greyed out (e.g. you cannot save a capture file if you haven't loaded one).

**The "Filter" toolbar**

![Filter Toolbar](image)

The filter toolbar lets you quickly edit and apply display filters

• **Filter:** Brings up the filter construction dialog, "The "Capture Filters" and "Display Filters" dialog boxes".

Filter input the area to enter or edit a display filter string expressions. A syntax check of your filterstring is done while you are typing. The background will turn red if you enter an incomplete or invalid string, and will become green when you enter a valid string. You can click on the pull down arrow to select a previously-entered filter string from a list. The entries in the pull down list will remain available even after a program restart.

• **Note:** After you've changed something in this field, don't forget to press the Apply button (or
the Enter/Return key), to apply this filter string to the display. This field is also where the current filter in effect is displayed.

- **Expression**: The middle button labeled "Add Expression..." opens a dialog box that lets you edit a display filter from a list of protocol fields, described in, “The "Filter Expression" dialog box"
- **Clear** Reset the current display filter and clears the edit area.
- **Apply** Apply the current value in the edit area as the new display filter.

**The "Packet List" pane**

The packet list pane displays all the packets in the current capture file.

Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes.

**The "Packet Details" pane**

The packet details pane shows the current packet (selected in the "Packet List" pane) in a more detailed form.

**The "Packet Bytes" pane**

The packet bytes pane shows the data of the current packet (selected in the "Packet List" pane) in a hexdump style.

```
0000  80 00 00 00 ff ff ff ff ff ff 00 24 d2 10 f9 a1 ... ... $...
0010  00 24 d2 10 f9 a1 50 ab 81 71 1e 32 0d 00 00 00 $. P . q. 2...
0020  64 00 31 04 00 04 32 34 30 30 01 08 82 84 8b 96 d.1...24 00...
```

**The Statusbar**

The statusbar displays informational messages. In general, the left side will show context related information, the middle part will show the current number of packets, and the right side will show the selected configuration profile. Drag the handles between the text areas to change the size.
**The initial Statusbar**

This statusbar is shown while no capture file is loaded, e.g. when Wireshark is started.

![Ready to load or capture](image)

The context menu (right mouse click) of the tab labels will show a list of all available pages. This can be helpful if the size in the pane is too small for all the tab labels.

**The Statusbar**

![File: C:UsersparthipDocumentssnip-task-wiresharkCapture 1.pcap size 625 KB 00:00:29 Packets: 2227 Displayed: 2227 Marked: 0 Load time: 00:00:033 Profile: 802.11ac](image)

The status bar displays informational messages. In general, the left side will show context related information, the middle part will show the current number of packets, and the right side will show the selected configuration profile. Drag the handles between the text areas to change the size.

**The Status bar with a loaded capture file**

- The left side shows information about the capture file, its name, its size and the elapsed time while it was being captured.
- The middle part shows the current number of packets in the capture file. The following values are displayed:
  - Packets: the number of captured packets
  - Displayed: the number of packets currently being displayed
  - Marked: the number of marked packets
  - Dropped: the number of dropped packets (only displayed if Wireshark was unable to capture all packets)
  - Ignored: the number of ignored packets (only displayed if packets are ignored)
- The right side shows the selected configuration profile. Clicking in this part of the statusbar will bring up a menu with all available configuration profiles, and selecting from this list will change the configuration profile.
Using Capture filters

Click on “CAPTURE”, “INTERFACES” options and choose the Network adapter from drop down menu which will be used to capture running packets in the network on the PC. Click on the “CAPTURE FILTERS” and enter the filter name and Filter string or directly input the filter string you know in the box. Then hit button. Now the wire shark sniffer program captures packets which are of interest to you only among the huge flow of real time packets of all types of protocols.
Wireshark Capture Interfaces

Wireshark Capture Options

Capture File(s)

- Use multiple files
- Next file every 1 megabyte(s)
- Next file every 1 minute(s)
- Ring buffer with 2 files
- Stop capture after 1 file(s)

Stop Capture ...

- ... after 1 packet(s)
- ... after 1 megabyte(s)
- ... after 1 minute(s)

Buffer size: 1 megabyte(s)

Capture Filter:

File: [Browse...]

Update list of packets
Automatic scrolling in live capture
Hide capture info dialog

Name Resolution

- Enable MAC name resolution
- Enable network name resolution
- Enable transport name resolution

Enter a capture filter to reduce the amount of packets to be captured. See "Capture Filters" in the online help for further information on how to use it. Syntax checking can be disabled in Preferences -> Capture -> Syntax check capture filter.
Display Filter

Once you have the captured file loaded you now set up filters to display packets you are interested in looking or avoid seeing packets not interested. This can be done by using simple filter expression or a combination of expression using logical operators to form a complex filter string.

Click on “ANALYZE” Select “DISPLAY FILTER”.

In the example below we are creating a filter to filter out the only the BEACON packets from a 802.11 wireless packet capture trace as seen below In the yellow highlighter.
Similar to the display filter we can find a particular packet by applying filter after clicking “Find packet”

Find FILTER button and entering the filter value in the filter box, if you don't know the string then you can dig further by clicking filter and hit NEW button and naming the filter strings and applying or typing the filter string with in the box. If you don't know the specific filter spring you can form it by choosing the EXPRESSION button which has various protocol option.

Select the one you are looking, expand and you will get more options to select from.

You will also have a Logical operator box to choose from to use to match to input the value you want to put and apply completing the filter.
You can build display filters that compare values using a number of different comparison operators.
<table>
<thead>
<tr>
<th>English</th>
<th>C-like</th>
<th>Description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>eq</td>
<td><code>==</code></td>
<td>Equal</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ip.src==10.0.0.5</code></td>
</tr>
<tr>
<td>ne</td>
<td><code>!=</code></td>
<td>Not equal</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ip.src!=10.0.0.5</code></td>
</tr>
<tr>
<td>gt</td>
<td><code>&gt;</code></td>
<td>Greater than</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>frame.len &gt; 10</code></td>
</tr>
<tr>
<td>lt</td>
<td><code>&lt;</code></td>
<td>Less than</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>frame.len &lt; 128</code></td>
</tr>
<tr>
<td>ge</td>
<td><code>&gt;=</code></td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>frame.len &gt;= 0x100</code></td>
</tr>
<tr>
<td>le</td>
<td><code>&lt;=</code></td>
<td>Less than or equal to</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>frame.len &lt;= 0x20</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>English</th>
<th>C-like</th>
<th>Description and example</th>
</tr>
</thead>
<tbody>
<tr>
<td>and</td>
<td><code>&amp;&amp;</code></td>
<td>Logical AND</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ip.src==10.0.0.5</code> and <code>tcp.flags.fin</code></td>
</tr>
<tr>
<td>or</td>
<td>`</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ip.src==10.0.0.5</code> or <code>ip.src==192.1.1.1</code></td>
</tr>
<tr>
<td>xor</td>
<td><code>^</code></td>
<td>Logical XOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29</code></td>
</tr>
<tr>
<td>not</td>
<td><code>!</code></td>
<td>Logical NOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>not llc</code></td>
</tr>
</tbody>
</table>

### Substring Operator

Wireshark allows you to select subsequences of a sequence in rather elaborate ways. After a label you can place a pair of brackets `[]` containing a comma separated list of range specifiers.
Using Coloring filter rule

A very useful mechanism available in Wireshark is packet colorization. You can set-up Wireshark so that it will colorize packets according to a filter. This allows you to emphasize the packets you are (usually) interested in.

You can set-up Wireshark so that it will colorize packets according to a filter you choose to create. This allows you to emphasize the packets you are (usually) interested in.

In the example below the packets are colorized for Beacons, Acknowledgement, probe Response, Deauthentication based on the filters mentioned below.

Click on "View"

Select "Coloring rules" or use "Edit coloring rules" from the main tool bar.
This opens the coloring rules and we can add a new coloring filter using "New" or the "Edit". Select the packet or edit the filter string and assign or adjust the color desired.
In the Edit Color dialog box, simply enter a name for the color filter, and enter a filter string in the Filter text field. “The "Edit Color Filter" dialog box” shows the values beacon and wlan.fc.type_subtype == 8 which means that the name of the color filter is Beacon and the filter will select protocols of type wlan.fc.type_subtype == 8 which is the beacon filter string. Once you have entered these values, you can choose a foreground and background color for packets that match the filter expression. Click on Foreground color... or Background color... to achieve this.

A very useful feature is to export or form the coloring filter and save it by exporting the filter to a file “tac80211color” as seen below this can be imported , you can create multiple coloring rule files in your troubleshooting folder and use it as a template to your convenience every time you troubleshoot.

You can think innovatively and tailor make coloring filter template files such as routing, wlan, switching etc. Color filters files and just import them depending on the problem you are troubleshooting very easily.
There is a good coloring rules download which you may download and use at https://supportforums.cisco.com/docs/DOC-23792

This is how the final look of the Wireshark packets window looks like after color filter file “tac80211color” is imported and applied.
**802.11 Sniffer Capture Analysis - Management Frames and Open Auth**

**Introduction**

802.11 Sniffer Capture Analysis - Management Frames and Open Auth

**802.11 – Frames and open authentication**

Trying to analyze or troubleshoot a wireless LAN, network using 802.11 packet analyzer will require us to have a thorough understanding of different 802.11 frame types as a basis for finding pointers to localize the causes of the problem area in a wlan network. Taking wlan sniffer traces using tools like omnipeek and or wireshark one can monitor the communications between radio network interface cards (NICs) and access points. We will need to comprehend each frame type occurring in the operation of a wireless LAN and solving network problems. In a wlan RF environment the radio transmission conditions can change so dynamically, coordination becomes a large issue in WLANs. Management and control packets are dedicated to these coordination functions.

To find cause of the wlan problems occurring in the wlan network relating to RF environment it would be best to test the wlan network using open authentication without any security. By taking this approach the RF connectivity issues surface and can be corrected before we can move to stronger encryption and higher layers of the OSI layer.

Authentication in the 802.11 specification is based on authenticating a wireless station or device instead of authenticating a user.

As per the 802.11 specification client authentication process consists of the following transactions as mentioned below

1. The Access points continuously sends out Beacon Frames which are picked up by the nearby wlan clients.
2. The client can also broadcast on its own probe request frame on every channel
3. Access points within range respond with a probe response frame
4. The client decides which access point (AP) is the best for access and sends an authentication request
5. The access point will send an authentication reply
6. Upon successful authentication, the client will send an association request frame to the access point
7. The access point will reply with an association response
8. The client is now able to pass traffic to the access point

**802.11 Client Authentication Process**

There are 3 types of frames used in the 802.11 MAC layer 2 communications happening over the air which manages and controls the wireless link.

They are Management Frames, Control Frames and Data frames. Let’s take a peek at what those frames consist of in little details to help us in analyze the wlan problems better while working with
Management Frames

802.11 management frames enable stations to establish and maintain communications. Management packets are used to support authentication, association, and synchronization.

The following are common 802.11 management frame subtypes:

- **Authentication frame**: This is a frame signifying to the network membership within the wlan topology. 802.11 authentications is a process whereby the access point either accepts or rejects the identity of a radio NIC to create resources. Authentication restricts the ability to send and receive on the network. It is the first step for a device attempting to connect to an 802.11 WLAN. The function is handled by an exchange of management packets. Authentication is handled by a request/response exchange of management packets. The number of packets exchanged depends on the authentication method employed. In this document we are focusing on the simplest open authentication method to simplify our troubleshooting of RF issues.

```
wlan.fc.type_subtype == 0x0b
```

The NIC begins the process by sending an authentication frame containing its identity to the access point. With open system authentication (the default), the radio NIC sends only one authentication frame, and the access point responds with an authentication frame as a response indicating acceptance (or rejection). There is an associated authentication ID associated which is the name under which the current station is authenticated itself on joining the network.

- **Deauthentication frame**: This is an announcement packet by a station which sends a deauthentication frame to another station if it wishes to terminate secure communications. It is a one-way communication from the authenticating station (a BSS or functional equivalent), and must be accepted. It takes effect immediately.

```
wlan.fc.type_subtype == 0x0c
```

```
wlan.fc.type_subtype == 0x0
```

```
wlan.fc.type_subtype == 0x01
```

```
wlan.fc.type_subtype == 0x02
```

The filter used to apply and find only the Disassociation packets is “wlan.fc.type_subtype == 0x0a”

The filter used to apply and find only the Beacon packets is

“wlan.fc.type_subtype == 0x08”

The filter used to apply and find only the Probe request packets is

“wlan.fc.type_subtype ==0x04”

The filter used to apply and find only the Probe request packets is “wlan.fc.type_subtype ==0x05”
Control Frames

802.11 control frames assist in the delivery of data frames between stations. The following are common 802.11 control frame subtypes:

- **Request to Send (RTS) frame**: The RTS/CTS function is optional and reduces frame collisions present when hidden stations have associations with the same access point. A station sends a RTS frame to another station as the first phase of a two-way handshake necessary before sending a data frame.

  wlan.fc.type_subtype == 0x1B

  wlan.fc.type_subtype == 0x1D

Data Frames

These at the frames which come later in the game after the basic wlan communication is already established between the Mobile station and the Access point. We will always reach to this 802.11 data frame for analysis typically to verify and analyze over the air if the protocols and data from higher layers within the frame body is getting through to the wire. These frames transport data packets from higher layers, such as web pages, printer control data, etc., within the body of the frame.

  wlan.fc.type_subtype == 0x20

On a packet analyzer we observe the contents of the frame body within 802.11 data frames for interesting traffic in question.

References

- [www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets](http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets)
- [https://supportforums.cisco.com/docs/DOC-13664](https://supportforums.cisco.com/docs/DOC-13664)

802.11 Sniffer Capture Analysis - WPA/WPA2 with PSK or EAP

**WPA-PSK(TKIP)**

1. Beacon frames are transmitted periodically to announce presence of wireless network and contain all information about it (data rates, channels, security ciphers, key management etc):

2. Probe request, is sent by STA to obtain information from AP:
3. Probe response, AP responds with a probe response frame, containing capability information, supported data rates, etc., when after it receives a probe request frame from STA:

4. 802.11 authentication is a process whereby the access point either accepts or rejects the identity of a radio NIC. The NIC begins the process by sending an authentication frame containing its identity to the access point. With open system authentication (the default), the radio NIC sends only one authentication frame, and the access point responds with an authentication frame as a response indicating acceptance (or rejection):

   a. Dot11 authentication request:

   b. Dot11 authentication response:

5. 802.11 association enables the access point to allocate resources for and synchronize with a radio NIC. A NIC begins the association process by sending an association request to an access point. This frame carries information about the NIC (e.g., supported data rates) and the SSID of the network it wishes to associate with.

   a. Dot11 association request:

   After receiving the association request, the access point considers associating with the NIC, and (if accepted) reserves memory space and establishes an association ID for the NIC.

   b. Dot11 association response:

6. 4-way handshake, during this phase PTK is created, PSK is used as PMK to construct those values:

   a. AP sends 802.1x authentication frame with ANonce, STA now has all information to construct PTK:

   b. STA responds with 802.1x authentication frame with SNonce and MIC:

   c. AP constructs 802.1x frame with new MIC and GTK with sequence number. This sequence number will be used in the next multicast or broadcast frame, so that the receiving STA can perform basic replay detection:

   d. STA sends ACK:

   From that point all data is sent encrypted.

**WPA2-PSK(AES/TKIP)**

The process if fairly the same as in previous section, I'll highlight only information that is different.
1. WPA2 AP management frame include RSN element that included unicast cipher suite, AKM information and GTK cipher suite (if both AES and TKIP are selected, then less stronger encryption method will be used for GTK).

2. During 4-way handshake frames contain version information for WPA2 in “Type” fields.

Note: you can decrypt WEP/WPA-PSK/WPA2-PSK encrypted wireless traffic if 4-way handshake key exchange frames are included in trace and PSK is known.

In order to encrypt wireless traffic in wireshark open Preferences-> Protocols->IEEE 802.11 and provide PSK information and select “Enable decryption option”.

To decrypt WPA/WPA2 encrypted traffic specify Key in format:

“wpa-psk:PSK:SSID”

Note: In order to filter out WLAN traffic from specific STA in wireshark you could use “WLAN Statistic” option.

In order to filter traffic from specific STA go to “Statistics -> WLAN Traffic”, from the list of SSIDs select corresponding SSID STA is associated with, and apply filter based on the STA.

How to decrypt WPA2 AES data on Over the Air Packet Captures with Wireshark

Requirements:

1. Capture to be on .pcap format.
2. Frames are to be presented in 802.11 format.
3. Know the SSID name and PSK for the WLAN from which Over the Air Capture has been collected.
4. Key: Capture the 4 EAPOL 4 way handshake.

The most accurate process to do this is to start the capture and then de-authenticate the client in order to catch the process from zero, meaning that the 4 way EAPOL handshake will be included.

If frames are within another format, like PEEKREMOTE it will be required to decode them, please section above on how to Decode PEEKREMOTE frames.

Process

Once capture has been opened in Wireshark go to “Edit” – “Preferences” Menu.
Go to “Protocols” menu and look for “IEEE 802.11”

From IEEE 802.11 section check the “Enable Decryption” check box and click on “Edit…” button
next to “Decryption Keys” label.

Once in the “Edit” menu click on New button on the left side of the window.

From the key type choose “wpa-psk”.

In order to obtain the key it is important to know the exact name of SSID and PSK for which decrypt process is being conducted.

Have these two values and go to the next website to generate the key based on these two elements.

- **WPA PSK (Raw Key) Generator**

Type in the SSID name and the PSK on the specified fields, string being typed into the fields must be exact as define for SSID and for PSK.

Once values have been defined, click on “Generate PSK”, this will generate the key, copy it and go back to Wireshark.
Paste the key that was generated into the “Key” field, click “OK”.

Then click “Apply” at the “Preferences” screen. Capture will begin to be decoded.

Once decoded it will be possible to see contents of 802.11 packets that were previously ciphered.
WPA/WPA2 Enterprise

1) WPA(TKIP)/WPA2(AES) with dot1x (PEAP)

This process follows the same steps like previous except for the AKM method and deriving PTK/GTK and AP advertised attributes in 802.11 management frames.

a. In this example AP advertises WPA(TKIP)/WPA2(AES) with dot1x authentication, both RSN and WPA tag attributes for AKM contain WPA value, whether in case of PSK authentication this field contains “PSK”. Also in this example TKIP is used for WPA and AES is used for WPA2.

b. STA selects one of authentication methods and cipher suites advertised by AP. In this case WPA2 with AES was selected, that can be seen in RSN IE parameters.

c. After successful dot1x association dot1x authentication takes place, during this process we can see what EAP method is used by STA for authentication and certificate(s) exchange information between supplicant and AAA server.

d. After successful dot1x authentication PMK is trasmitted to AP in “Access-Accept” message from AAA server and the same PMK is derived on the client, next 4-way handshake takes place and PTK and GTK establishment.

Radius exchange between WLC and AAA server:

General flow diagram:

WPA(TKIP)/WPA2(AES) with dot1x (EAP-TLS)

Difference for this type of authentication compared to the previous one is that client provides its certificate in “Client Hello” message and mutual authentication is performed between client and AAA server based on certificates.

EAP exchange between STA and WLC:

Radius exchange between WLC and AAA server:

General flow diagram:
2) WPA(TKIP)/WPA2(AES) with dot1x (FAST)

Only dot1x authentication stage is a bit different comparing to the previous example. After successful dot11 association dot1x authentication takes place, AP sends dot1x identity request to the STA and STA provides identity response, the response depends on what PAC provisioning is been in use (in-band PAC provisioning (phase 0) or out-of-band PAC provisioning). In case of in-band provisioning PAC is sent to the client from AAA server, once client has PAC it goes to EAP-FAST phase1 from this point TLS tunnel establishment starts (phase 1)

After TLS tunnel is established inner authentication method (phase 2) starts inside encrypted tunnel. On successful authentication PMK is send in “Access-Accept” message to AP from AAA server, the same key is derived based on dot1x exchange on STA. This key (PMK) is used to calculate PTK during 4-way handshake that will be used to secure communication between STA and AP.

General flow diagram:

802.11 Sniffer Capture Analysis – Multicast

Introduction

Multicast Sniffing

Solution

The controller performs multicasting in two modes:

- **Unicast** mode—In this mode, the controller unicasts every multicast packet to every AP associated to the controller. This mode is inefficient but might be required on networks that do not support multicasting.
- **Multicast** mode—In this mode, the controller sends multicast packets to an LWAPP multicast group. This method reduces overhead on the controller processor and shifts the work of packet replication to your network, which is much more efficient than the unicast method.
- You can enable multicast mode using the controller GUI or CLI.

IGMP Snooping on WLC

In controller software release 4.2, IGMP snooping is introduced to better direct multicast packets. When this feature is enabled, the controller gathers IGMP reports from the clients, processes the reports, creates unique multicast group IDs (MGIDs) from the IGMP reports after checking the Layer 3 multicast address and the VLAN number, and sends the IGMP reports to the infrastructure switch. The controller sends these reports with the source address as the interface address on which it received the reports from the clients.

The controller then updates the access point MGID table on the AP with the client MAC address. When the controller receives multicast traffic for a particular multicast group, it forwards it to all the APs. However, only those APs that have active clients listening or subscribed to that multicast group send multicast traffic on that particular WLAN. IP packets are forwarded with an MGID that
is unique for an ingress VLAN and the destination multicast group. Layer 2 multicast packets are forwarded with an MGID that is unique for the ingress interface.

**Note:** IGMP snooping is not supported on the 2000 series controllers, the 2100 series controllers, or the Cisco Wireless LAN Controller Network Module for Cisco Integrated Services Routers.

**Guidelines for Using Multicast Mode**

Use these guidelines when you enable multicast mode on your network:

The Cisco Unified Wireless Network solution uses some IP address ranges for specific purposes. Keep these ranges in mind when you configure a multicast group: Although not recommended, any multicast address can be assigned to the LWAPP multicast group; this includes the reserved link local multicast addresses used by OSPF, EIGRP, PIM, HSRP, and other multicast protocols.

Cisco recommends that multicast addresses be assigned from the administratively scoped block 239/8. IANA has reserved the range of 239.0.0.0-239.255.255.255 as administratively scoped addresses for use in private multicast domains. See the note for additional restrictions. These addresses are similar in nature to the reserved private IP unicast ranges, such as 10.0.0.0/8, defined in RFC 1918. Network administrators are free to use the multicast addresses in this range inside of their domain without fear of conflict with others elsewhere in the Internet. This administrative or private address space must be used within the enterprise and its leave or entry blocked from the autonomous domain (AS).

**Note:** Do not use the 239.0.0.X address range or the 239.128.0.X address range. Addresses in these ranges overlap with the link local MAC addresses and flood out all switch ports, even with IGMP snooping turned on.

Cisco recommends that enterprise network administrators further subdivide this address range into smaller geographical administrative scopes within the enterprise network to limit the "scope" of particular multicast applications. This prevents high-rate multicast traffic from leaving a campus (where bandwidth is plentiful) and congesting the WAN links. It also allows for efficient filtering of the high bandwidth multicast from reaching the controller and the wireless network.

When you enable multicast mode on the controller, you must configure an LWAPP multicast group address on the controller. APs subscribe to the LWAPP multicast group using Internet Group Management Protocol (IGMP).

- Cisco 1100, 1130, 1200, 1230, and 1240 APs use IGMP versions 1, 2, and 3. However, Cisco 1000 Series APs use only IGMP v1 to join the multicast group.
- Multicast mode works only in Layer 3 LWAPP mode.
- APs in monitor mode, sniffer mode, or rogue detector mode do not join the LWAPP multicast group address.
- When you use controllers that run version 4.1 or earlier, you can use the same multicast address on all the controllers. If you use controllers that run version 4.2 or later, the LWAPP multicast group configured on the controllers must be different for each controller used on the network.
- If you use controllers with version 4.1 or earlier, the multicast mode does not work across intersubnet mobility events, such as guest tunneling, site-specific VLANs, or interface override that uses RADIUS. The multicast mode does work in these subnet mobility events when you disable the Layer 2 IGMP snooping/CGMP features on the wired
LAN.

In later versions, that is, 4.2 or later, the multicast mode does not operate across intersubnet mobility events, such as guest tunneling. It does, however, operate with interface overrides that use RADIUS (but only when IGMP snooping is enabled) and with site-specific VLANs (access point group VLANs).

- The controller drops any multicast packets sent to the UDP port numbers 12222, 12223, and 12224. Make sure the multicast applications on your network do not use those port numbers.
- Multicast traffic is transmitted at 6 Mbps in an 802.11a network. Therefore, if several WLANs attempt to transmit at 1.5 Mbps, packet loss occurs. This breaks the multicast session.

**Configuring Multicast (Using Multicast-Multicast Mode)**

Select Multicast - Multicast and configure your group, each WLC in your mobility group should use a unique address.

Enable multicast routing on the L3 device and enable PIM on the following VLANs. Management, AP-Manger, VLAN on which the AP are in and as well as the VLAN where the clients that will receive the multicast stream. Example:

VLAN 40 is the WLC management, VLAN 40 is for AP, and VLAN 50 is where my clients are. So under all of these SVI I need to issue the multicast commands.

Issue all Multicast show command to verify, example: show ip mroute, show ip igmp groups to validate that the group for the AP is built properly.

We can also enable IGMP Snooping on the WLC. The WLC will hold its own snooping table for the IGMP messages that it receives, so that it knows who is requesting the stream.

**On Wireless LAN Controller**

Enable Global Multicast on the WLC and Enable Multicast – Multicast mode on the WLC.
Once the client sends the multicast join, we will see the below on the WLC MGID

Remember this address needs to be not in use by any actual multicast applications on the network.
Multicast configuration on Wired network

Configure Multicast routing Globally and then enable PIM on each interface.

6504-WLCBUG#sh run | i multicast

ip multicast-routing

6504-WLCBUG#sh run int vla 50

Building configuration...

Current configuration : 119 bytes

interface Vlan50

description // WLAN DHCP pool VLAN //

ip address 172.16.1.1 255.255.255.0

ip pim dense-mode
Building configuration...

Current configuration : 121 bytes

interface Vlan40

description // Management Vlan //

ip address 10.105.135.136 255.255.255.128

ip pim dense-mode

dr

Interface Vlan40

6504-WLCBUG#sh ip pim interface vlan 40

Address Interface Ver/ Nbr Query DR DR Mode Count Intvl Prior

10.105.135.136 Vlan40 v2/D 0 30 1 10.105.135.136

6504-WLCBUG#sh ip pim interface vlan 50

Address Interface Ver/ Nbr Query DR DR Mode Count Intvl Prior

172.16.1.1 Vlan50 v2/D 0 30 1 172.16.1.1

6504-WLCBUG#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.255.255.255), 4d17h/00:02:03, RP 0.0.0.0, flags: DC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Vlan40, Forward/Dense, 4d17h/00:00:00

(*, 239.254.0.3), 2w1d/00:02:07, RP 0.0.0.0, flags: DC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Vlan40, Forward/Dense, 3d10h/00:00:00

(*, 224.0.1.40), 2w1d/00:02:13, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Vlan11, Forward/Dense, 2w1d/00:00:00

Packet Captures

Topology

Wired PC ----------- 6500 Switch -------- WISM ------- AP ))) ))) ((( Wireless Client

Vlan 50      Vlan 40      Vlan 40      Vlan 40      Vlan 50

MCAST Traffic Generator Tool

is used on the Wired PC to Generate Multicast Stream – Continuous UDP packets.
Wired Wireshark packet capture on the MCAST generator

Windows Netmon Capture on the Mcast packet generator
MCAST Receiver Tool is used on the Wireless Client to Receive the Multicast traffic from the Source (Wired PC).

Wireshark Captures on the Wireless interface of the Wireless client

Netmon Capture on the Wireless interface of the Wireless Client
802.11 Sniffer Capture Analysis – Web Authentication

Introduction

WEB AUTHENTICATION Sniffer on Cisco WLC Troubleshooting

Web Authentication Process

Web authentication is a Layer 3 security feature that causes the controller to not allow IP traffic, except DHCP-related packets/ DNS-related packets, from a particular client until that client has correctly supplied a valid username and password with an exception of traffic allowed through Pre-Auth ACL. Web authentication is the only security policy that allows the client to get an IP address before Authentication. It is a simple Authentication method without the need for a supplicant or client utility. Web authentication can be done either locally on a WLC or over a RADIUS server. Web authentication is typically used by customers who want to deploy a guest-access network.

Web authentication starts when the controller intercepts the first TCP HTTP (port 80) GET packet from the client. In order for the client’s web browser to get this far, the client must first obtain an IP address, and do a translation of the URL to IP address (DNS resolution) for the web browser. This lets the web browser know which IP address to send the HTTP GET.

When web authentication is configured on the WLAN, the controller blocks all traffic (until the authentication process is completed) from the client, except for DHCP and DNS traffic. When the client sends the first HTTP GET to TCP port 80, the controller redirects the client to https:1.1.1.1/login.html for processing. This process eventually brings up the login web page.

- You open the web browser and type in a URL, for example, http://www.google.com. The client sends out a DNS request for this URL to get the IP for the destination. WLC bypasses the DNS request to the DNS server and DNS server responds back with a DNS reply, which contains the IP address of the destination www.google.com, which in turn is forwarded to the wireless clients.
- The client then tries to open a TCP connection with the destination IP address. It sends out a TCP SYN packet destined to the IP address of www.google.com.
- The WLC has rules configured for the client and hence can act as a proxy for www.google.com. It sends back a TCP SYN-ACK packet to the client with source as the IP address of www.google.com. The client sends back a TCP ACK packet in order to complete the three-way TCP handshake and the TCP connection is fully established.
- The client sends an HTTP GET packet destined to www.google.com. The WLC intercepts this packet,
sends it for redirection handling. The HTTP application gateway prepares a HTML body and sends it back as the reply to the HTTP GET requested by the client. This HTML makes the client to go to the default webpage URL of the WLC, for example, <Virtual-Server-IP>/login.html.

- Client closes the TCP connection with the IP address, for example www.google.com.
- Now the client wants to go to http://1.1.1.1/login.html and so it tries to open a TCP connection with the virtual IP address of the WLC. It sends a TCP SYN packet for 1.1.1.1 to the WLC.
- The WLC responds back with a TCP SYN-ACK and the client sends back a TCP ACK to the WLC in order to complete the handshake.
- Client sends a HTTP GET for /login.html destined to 1.1.1.1 in order to request for the login page.
- This request is allowed up to the Web Server of the WLC, and the server responds back with the default login page. The client receives the login page on the browser window where the user can go ahead and log in.

Configuration Webauth

Lets Go ahead and configure.

TOPOLOGY

A Wireless Client is connected to the AP which is registered to the WLC which is connected to the switch, which is connected to the Router where the DNS, Routing, L3 connectivity is configured.

Router to Act as a DNS

```
3825#sh run | i host
hostname 3825
ip host www.google.com 200.200.200.1
ip host www.yahoo.com 200.200.200.2
ip host www.facebook.com 200.200.200.3
3825#sh run | i dns
dns-server 16.16.16.1
ip dns server
3825#sh run | b dhcp
ip dhcp excluded-address 16.16.16.1 16.16.16.5
ip dhcp pool webauth-sniffer network 16.16.16.0 255.255.255.0
default-router 16.16.16.1
dns-server 16.16.16.1
```

Configuration on the WLC

- Go to WLAN and then NEW and enter the details
- Configure the WLAN for NO L2 Auth
- Configure the WLAN for L3 auth with WEBAUTH
- After config, you should see like this
- Go to Security TAB and create Local net users
- Enter the Clients credentials
- Go to Security >> Webauth TAB and the chose the Web auth type == Internal / External Redirect / Custom
- Connect the client
- After getting the IP address, open the browser and type in the web address..
- The Clients gets redirected to the Web auth page where in we enter the username and password
- After successful log in
- Client getting Redirected to the Redirect page

Here is the Packet Capture When the client tries to connect

**Client does:**
The client's IP address is 16.16.16.7. The client resolved the URL to the web server it was accessing 200.200.200.1. As you can see, the client did the three way handshake to start up the TCP connection and then sent an HTTP GET packet starting with packet 576. The controller is intercepting the packets and replying with code 200. The code 200 packet has a redirect URL in it:

**Client gets the HTTPS Login page**

**Client Accepts the Certificate**

The client then starts the HTTPS connection to the redirect URL which sends it to the 1.1.1.1, which is the virtual IP address of the controller. The client has to validate the server certificate or ignore it in order to bring up the SSL tunnel. Here the Client tries to access Facebook.com after successful auth and his TCP session starts without any problem.

**Here is the client Debug when the the client tried connecting**

```
Cisco Controller) > (Cisco Controller) >show debug MAC address
............
00:21:5c:8c:c7:61 Debug Flags Enabled: dhcp packet enabled. dot11 mobile enabled. dot11 state enabled. dot1x events enabled. dot1x states enabled. pem events enabled. pem state enabled. CCKM client debug enabled. webauth redirect enabled.

May 18 13:43:50.568: 00:21:5c:8c:c7:61 Adding mobile on LWAPP AP a8:b1:d4:c4:35:b0(0) *apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 Association received from mobile on AP a8:b1:d4:c4:35:b0

*apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 STA - rates (8): 130 132 139 150 12 18 24 36 0 0 0 0 0 0 0 0 *apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 STA - rates (12): 130 132 139 150 12 18 24 36 48 72 96 108 0 0 0 0 *apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 0.0.0.0 START (0) Initializing policy *apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 0.0.0.0 DHCP Not required on AP a8:b1:d4:c4:35:b0 vapId 1 apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 0.0.0.0 L2AUTHCOMPLETE (4) last state AUTHCHECK (2) Change state to DHCP_REQD (7) last state DHCP_REQD (7) *apfMsConnTask_0: May 18 13:43:50.568: 00:21:5c:8c:c7:61 Scheduling deletion of
```
Mobile Station: (callerId: 49) in 1800 seconds *apfMsConnTask_0: May 18 13:43:50.569: 00:21:5c:8c:c7:61 Sending Assoc Response to station on BSSID a8:b1:d4:c4:35:b0 (status 0) ApVapId 1 Slot 0 *apfMsConnTask_0: May 18 13:43:50.569: 00:21:5c:8c:c7:61 apfProcessAssocReq (apf_80211.c:5272) Changing state for mobile 00:21:5c:8c:c7:61 on AP a8:b1:d4:c4:35:b0 from Associated to Associated *apfReceiveTask: May 18 13:43:50.570: 00:21:5c:8c:c7:61 0.0.0.0 DHCPREQD (7) State Update from Mobility-Incomplete to Mobility-Complete, mobility role=Local, client state=APF_MS_STATE_ASSOCIATED *apfReceiveTask: May 18 13:43:50.570: 00:21:5c:8c:c7:61 0.0.0.0 0.0.0.0 DHCPREQD (7) Fast Path rule type = Airespace AP - Learn IP address on AP a8:b1:d4:c4:35:b0, slot 0, interface = 1, QOS = 0 ACL Id = 255, Jumbo Fr *apfReceiveTask: May 18 13:43:50.570: 00:21:5c:8c:c7:61 0.0.0.0 DHCPREQD (7) Successfully plumbed mobile rule (ACL ID 255) *pemReceiveTask: May 18 13:43:50.689: 00:21:5c:8c:c7:61 DHCP received op BOOTREQUEST (1) (len 310,vlan 0, port 1, encap 0xec03) *DHCP Socket Task: May 18 13:43:50.689: 00:21:5c:8c:c7:61 DHCP processing DHCP DISCOVER (1) *DHCP Socket Task: May 18 13:43:50.689: 00:21:5c:8c:c7:61 DHCP received op BOOTREQUEST (1) (len 310,vlan 0, port 1, encap 0xec03) *DHCP Socket Task: May 18 13:43:50.689: 00:21:5c:8c:c7:61 DHCP successfully bridged packet to STA
DHCP client associated to interface webauth-sniffer which can support client subnet.


Reference:

Here is a reference to the Wikipedia article on the 802.11 specifications: [IEEE 802.11 Standards](https://en.wikipedia.org/wiki/IEEE_802.11)