

# Understanding Debug Client on Wireless LAN Controllers (WLCs)

Document ID: 100260

---

## Introduction

### Prerequisites

- Requirements
- Components Used
- Conventions

### Debug Client

### Debug Client Variations

- Mobility
- EAP Authentication Troubleshooting

### Client Connection

### Controller Processes

- Policy Enforcement Module (PEM)
- Client Traffic Forwarding
- Access Point Functions (APF)
- 802.1x Authentication (Dot1x)

### Debug Client Analysis

### Troubleshooting Examples

- Wrong Client Cipher Configuration
- Wrong Preshared Key

### Related Information

---

## Introduction

This document provides detailed information about the **debug client** command output on wireless LAN controllers.

This document covers these topics:

- How a wireless client is handled
- Troubleshooting basic association and authentication issues

The output to be analyzed covers the scenario for a WPA pre-shared key (WPA-PSK) network.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- How to configure the wireless LAN controller (WLC) and Lightweight Access Point (LAP) for basic operation
- Lightweight Access Point Protocol (LWAPP) and wireless security methods
- How the 802.11 authentication and association processes work

## Components Used

The information in this document is based on these software and hardware versions:

- Cisco 2000/2100/4400 Series WLC that runs firmware 4.1 or 4.2
- LWAPP-based access points

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

## Debug Client

The command **debug client** <MACADDRESS> is a macro that enables eight debug commands, plus a filter on the MAC address provided, so only messages that contain the specified MAC address are shown. The eight debug commands show the most important details on client association and authentication. The filter helps with situations where there are multiple wireless clients. Situations such as when too much output is generated or the controller is overloaded when debugging is enabled without the filter.

The information collected covers important details about client association and authentication (with two exceptions mentioned later in this document).

The commands that are enabled are shown in this output:

```
(Cisco Controller) >show debug

MAC address ..... 00:00:00:00:00:00

Debug Flags Enabled:
  dhcp packet enabled.
  dot11 mobile enabled.
  dot11 state enabled.
  dot1x events enabled.
  dot1x states enabled.
  pem events enabled.
  pem state enabled.
```

These commands cover address negotiation, 802.11 client state machine, 802.1x authentication, Policy Enforcement Module (PEM), and address negotiation (DHCP).

## Debug Client Variations

For most scenarios, the **debug client** <MACAddress> command is enough to get the information needed. However, here are two important situations where additional debugging is needed:

- Mobility (client roaming between controllers)
- EAP Authentication Troubleshooting

## Mobility

In this situation, mobility debugs need to be enabled after the **debug client** <MACAddress> command has been introduced in order to gain additional information on the mobility protocol interaction between controllers.

**Note:** Details on this output will be covered in future documents.

In order to enable mobility debugs, use the **debug client** <MACAddress>, and then use the **debug mobility handoff enable** command:

```
(Cisco Controller) >debug client 00:00:00:00:00:00
(Cisco Controller) >debug mobility handoff enable
(Cisco Controller) >show debug

MAC address ..... 00:00:00:00:00:00

Debug Flags Enabled:
  dhcp packet enabled.
  dot11 mobile enabled.
  dot11 state enabled
  dot1x events enabled.
  dot1x states enabled.
  mobility handoff enabled.
  pem events enabled.
  pem state enabled.
```

## EAP Authentication Troubleshooting

In order to troubleshoot the interaction between the WLC and the authentication server (external RADIUS or internal EAP server), use the command **debug AAA all enable**, which shows the required details. This command should be used after the **debug client** <MACAddress> command and can be combined with other debug commands as needed (for example, **handoff**).

```
(Cisco Controller) >debug client 00:00:00:00:00:00
(Cisco Controller) >debug aaa all enable
(Cisco Controller) >show debug
MAC address ..... 00:00:00:00:00:00
Debug Flags Enabled:
  aaa detail enabled.
  aaa events enabled.
  aaa packet enabled.
  aaa packet enabled.
  aaa ldap enabled.
  aaa local-auth db enabled.
  aaa local-auth eap framework errors enabled.
  aaa local-auth eap framework events enabled.
  aaa local-auth eap framework packets enabled.
  aaa local-auth eap framework state machine enabled.
  aaa local-auth eap method errors enabled.
  aaa local-auth eap method events enabled.
  aaa local-auth eap method packets enabled.
  aaa local-auth eap method state machine enabled.
  aaa local-auth shim enabled.
  aaa tacacs enabled.
  dhcp packet enabled.
  dot11 mobile enabled.
  dot11 state enabled
  dot1x events enabled
```

```
dot1x states enabled.  
mobility handoff enabled.  
pem events enabled.  
pem state enabled.
```

## Client Connection

For the purposes of this document, *client connection* is the process for a wireless client to pass through these steps:

### 802.11 Section

1. Probing, to find a valid AP to associate.
2. Authentication: Can be Open (null) or Shared. Normally, Open is selected.
3. Association: Requesting data services to the AP.

### L2 Policies Section

1. None; PSK or EAP authentication takes place depending on configuration.
2. Key negotiation, if an encryption method is selected.

### L3 Policies Section

1. Address learning.
2. Web authentication, if selected.

**Note:** These steps represent a subset or summary of the full process. This document describes a simplified scenario that covers 802.11 and L2 policies and uses WPA-PSK, plus address learning. No external AAA or L3 policies for authentication are used.

## Controller Processes

In each section, the controller uses separated processes in order to keep track of the state of the client at each moment. The processes interact between them to ensure that the client is added to the connection table (per the security policies configured). In order to understand the client connection steps to the controller, here is a short summary of the most relevant processes:

- **Policy Enforcement Module (PEM)** Controls the client state and forces it through each of the security policies on the WLAN configuration.
- **Access Point Functions (APF)** Basically, the 802.11 state machine.
- **Dot1x** Implements the state machine for 802.1x, PSK authentication, and key handling for the wireless clients.
- **Mobility** Tracks interaction with other controllers on the same mobility group.
- **Data Transformation Layer (DTL)** Sits between the software components and the network hardware acceleration (NPU); controls the ARP information.

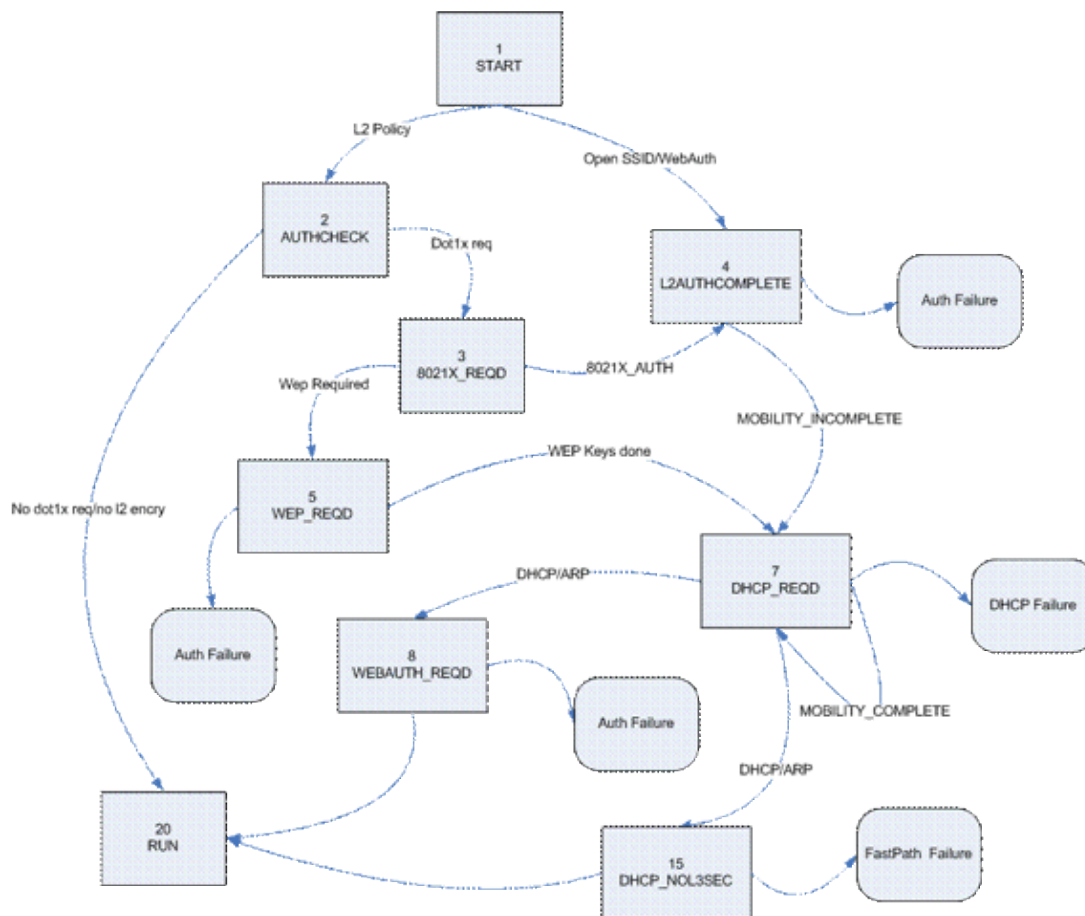
## Policy Enforcement Module (PEM)

Based on the WLAN configuration, the client passes through a series of steps. PEM ensures this is done in order for it to comply with the required L2 and L3 security policies.

Here is a subset of the PEM states relevant for the analysis of a client debug:

- **START** Initial status for new client entry.
- **AUTHCHECK** WLAN has an L2 authentication policy to enforce.
- **8021X\_REQD** Client must complete 802.1x authentication.
- **L2AUTHCOMPLETE** Client has successfully finished the L2 policy. The process can now proceed to L3 policies (address learning, Web auth, etc). Controller sends here the mobility announcement to learn L3 information from other controllers if this is a roaming client in the same mobility group.
- **WEP\_REQD** Client must complete WEP authentication.
- **DHCP\_REQD** Controller needs to learn the L3 address from client, which is done either by ARP request, DHCP request or renew, or by information learned from other controller in the mobility group. If DHCP Required is marked on the WLAN, only DHCP or mobility information are used.
- **WEBAUTH\_REQD** Client must complete Web authentication. (L3 policy)
- **RUN** Client has successfully completed the required L2 and L3 policies and can now transmit traffic to the network.

This figure shows a simplified PEM state machine with the client transitions until it reaches the RUN state, where the client can now send traffic to the network:



**Note:** This figure does not cover all possible transitions and states. Some intermediate steps have been removed for clarity.

## Client Traffic Forwarding

Between the START state and before the final RUN state, the client traffic is not forwarded to the network, but is passed to the main CPU on the controller for analysis. The information that is forwarded depends on the state and the policies in place; for example, if 802.1x is enabled, EAPOL traffic is forwarded to the CPU. Another example is if Web Auth is used, then HTTP and DNS is allowed and intercepted by the CPU to do the web redirection and obtain client authentication credentials.

When the client reaches the RUN state, the client information is sent to the NPU in order to enable FastPath switching, which does a cable-rate forwarding of the user traffic to the client VLAN and frees the central CPU of user data forwarding tasks.

The traffic that is forwarded depends on the client type that is applied to the NPU. This table describes the most relevant types:

Type	Description
1	Normal client traffic forwarding.
9	IP learning state. One packet from this client is sent to CPU in order to learn the IP address used.
2	ACL pass-through. Used when the WLAN is an ACL configured to inform the NPU.

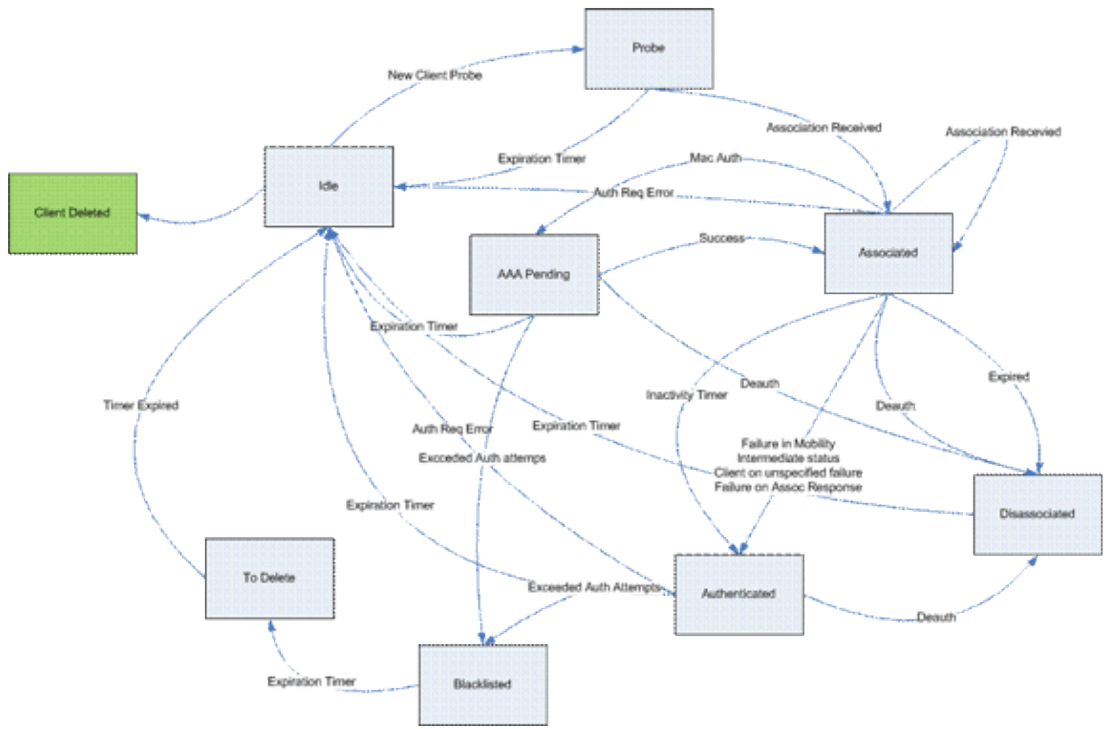
## Access Point Functions (APF)

This process handles the state of the client through the 802.11 machine state and interacts with mobility code in order to validate the different roaming scenarios. This document does not cover the mobility details or its states.

The following table shows the more relevant client states that are entered in during a client association to the controller:

Name	Description
Idle	New client or temporary state on some situations.
AAA Pending	Client waiting for MAC address authentication.
Authenticated	Open authentication successful or intermediate state in some situations.
Associated	Client successfully passed MAC auth and open auth processes.
Disassociated	Client sent disassociation/deauthentication, or association timer expired.
To Delete	Client marked to be deleted (normally after exclusion timer expired).
Probe	Probe request received for new client.
Excluded/Blacklisted	Client has been marked as excluded. Normally related to WPS policies.
Invalid	Error on client state.

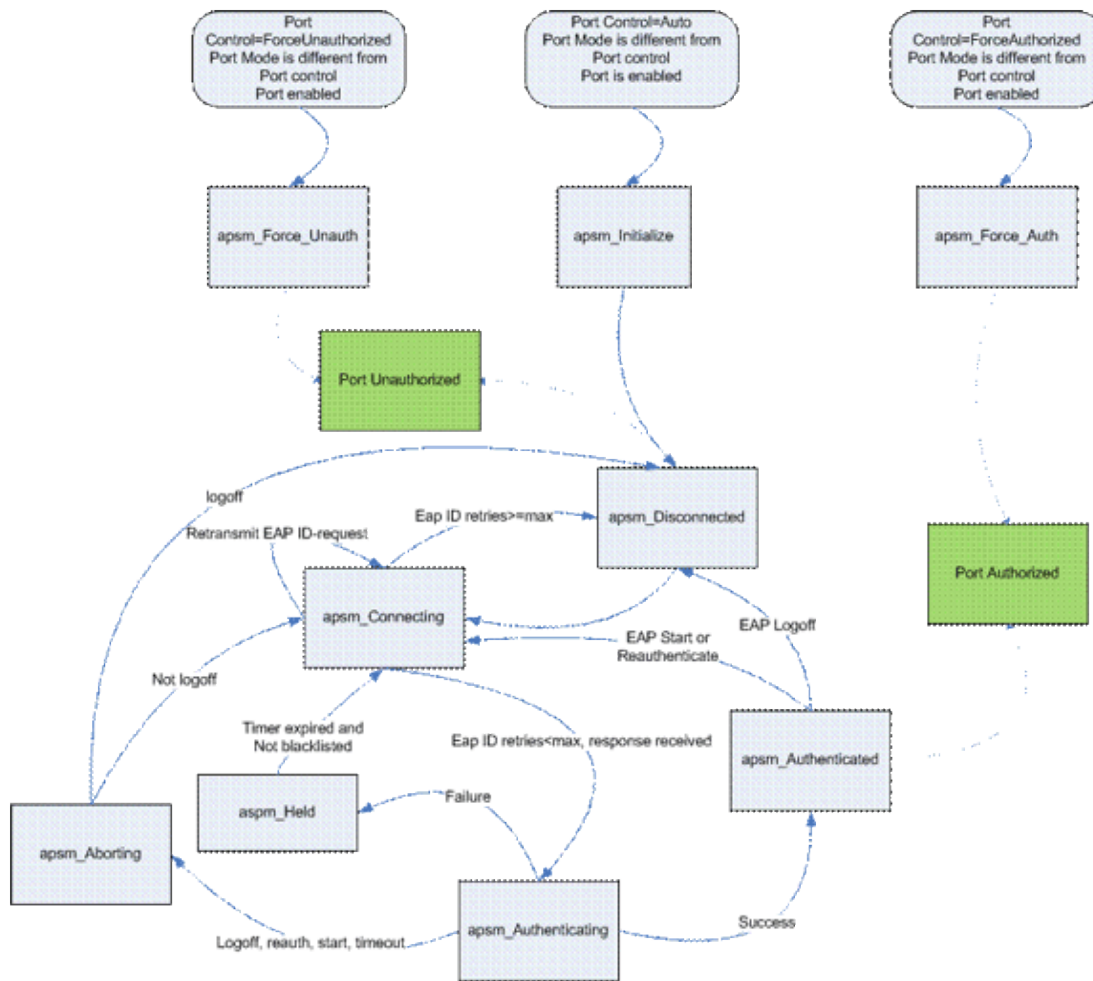
This figure represents a state machine transition and shows only the most relevant states and transitions:



## 802.1x Authentication (Dot1x)

The Dot1x process is responsible for 802.1x authentication and key management for the client. This means that, even on WLANs that do not have an EAP policy requiring 802.1x, Dot1x participates to handle the key creation and negotiation with client and also for the cached key handling (PMK or CCKM).

This state machine shows the full 802.1x transitions:



## Debug Client Analysis

### APF Process

```
Wed Oct 31 10:46:13 2007: 00:1b:77:42:07:69 Adding mobile on LWAPP AP
00:1c:0j:ca:5f:c0(0)
```

```
!--- A new station is received. After validating type, it is added to the
!--- AP that received it. This can happen both on processing association
!--- request or probe requests
```

```
Wed Oct 31 10:46:13 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 23) in 5 seconds
```

```
!--- Sets an expiration timer for this entry in case it does not progress
!--- beyond probe status. 5 Seconds corresponds to Probe Timeout. This message
!--- might appear with other time values since, during client processing,
!--- other functions might set different timeouts depending on state.
```

```
Wed Oct 31 10:46:13 2007: 00:1b:77:42:07:69 apfProcessProbeReq
(apf_80211.c:4057) Changing state for mobile 00:1b:77:42:07:69 on AP
00:1c:0j:ca:5f:c0 from Idle to Probe
```

```
!--- APF state machine is updated.
```

Wed Oct 31 10:46:13 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 24) in 5 seconds

*!--- New Probe request update sent AP about client. IMPORTANT:  
!--- Access points do not forward all probe requests to the controller; they  
!--- summarize per time interval (by default 500 msec). This information is  
!--- used later by location and load balancing processes.*

Wed Oct 31 10:46:14 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 24) in 5 seconds

*!--- New Probe request update sent AP about client.*

Wed Oct 31 10:46:14 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 24) in 5 seconds

*!--- New Probe request update sent AP about client.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 24) in 5 seconds

*!--- New Probe request update sent AP about client.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Association received from mobile on AP 00:1c:0j:ca:5f:c0

*!--- Access point reports an association request from the client.  
!--- When the process reaches this point, the client is not excluded and not  
!--- in mobility intermediate state*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 STA - rates (8): 140 18 152  
36 176 72 96 108 0 0 0 0 0 0 0

*!--- Controller saves the client supported rates into its connection table.  
!--- Units are values of 500 kbps, basic (mandatory) rates have the MSb set.  
!--- The above would be 6mbps basic, 9, 12 basic, 18, 24 basic, 36, 48, 54*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Processing WPA IE type 221, length 24 for mobile 00:1b:77:42:07:69

*!--- Controller validates the 802.11i security information element.*

#### **PEM Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Deleted mobile LWAPP rule on AP [00:1c:0j:ca:5f:c0]

*!--- As the client requests new association, APF requests to PEM to delete the  
!--- client state and remove any traffic forwarding rules that it could have.*

#### **APF Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Updated location for station old AP 00:00:00:00:00:00-0, new AP 00:1c:0j:ca:5f:c0-1

*!--- APF updates where this client is located. For example, this client is  
!--- a new addition; therefore, no value exists for the old location.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Initializing  
policy

*!--- PEM notifies that this is a new user. Security policies are checked  
!--- for enforcement.*

#### **PEM Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Change state  
to AUTHCHECK (2) last state AUTHCHECK (2)

*!--- PEM marks as authentication check needed.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 AUTHCHECK (2) Change  
state to 8021X\_REQD (3) last state 8021X\_REQD

*!--- After the WLAN configuration is checked, the client will need either  
!--- 802.1x or PSK authentication*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 8021X\_REQD (3) Plumbed  
mobile LWAPP rule on AP 00:1c:0j:ca:5f:c0

*!--- PEM notifies the LWAPP component to add the new client on the AP with  
!--- a list of negotiated capabilities, rates, Qos, etc.*

#### **APF Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 apfPemAddUser2 (apf\_policy.c:209)  
Changing state for mobile 00:1b:77:42:07:69 on AP 00:1c:0j:ca:5f:c0 from  
Probe to Associated

*!--- APF notifies that client has been moved successfully into associated  
!--- state.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Stopping deletion of Mobile  
Station: (callerId: 48)

*!--- The expiration timer for client is removed, as now the session timeout  
!--- is taking place. This is also part of the above notification  
!--- (internal code callerId: 48).*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Sending Assoc Response to  
station on BSSID 00:1c:0j:ca:5f:c0 (status 0)

*!--- APF builds and sends the association response to client.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 apfProcessAssocReq  
(apf\_80211.c:3838) Changing state for mobile 00:1b:77:42:07:69 on AP  
00:1c:0j:ca:5f:c0 from Associated to Associated

*!--- The association response was sent successfully; now APF keeps the  
!--- client in associated state and sets the association timestamp on this point.*

### **Dot1x Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Creating a new PMK Cache Entry  
for station 00:1b:77:42:07:69 (RSN 0)

*!--- APF calls Dot1x to allocate a new PMK cached entry for the client.  
!--- RSN is disabled (zero value).*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Initiating WPA PSK to mobile  
00:1b:77:42:07:69

*!--- Dot1x signals a new WPA or WPA2 PSK exchange with mobile.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 dot1x - moving mobile  
00:1b:77:42:07:69 into  
Force Auth state

*!--- As no EAPOL authentication takes place, the client port is marked as  
!--- forced Auth. Dot1x performs key negotiation with PSK clients only.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Skipping EAP-Success to mobile  
00:1b:77:42:07:69

*!--- For PSK, CCKM or RSN, the EAP success is not sent to client, as there  
!--- was no EAPOL authentication taking place.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Sending EAPOL-Key Message to  
mobile  
00:1b:77:42:07:69

state INITPMK (message 1), replay counter 00.00.00.00.00.00.00.00

*!--- Dot1x starts the exchange to arrive into PTK. PMK is known, as this  
!--- is PSK auth. First message is ANonce.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile  
00:1b:77:42:07:69

*!--- Message received from client.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Received EAPOL-key in PKT\_START  
state (message 2) from mobile 00:1b:77:42:07:69

*!--- This signals the start of the validation of the second message  
!--- from client (SNonce+MIC). No errors are shown, so process continues.  
!--- Potential errors at this point could be: deflection attack (ACK bit  
!--- not set on key), MIC errors, invalid key type, invalid key length, etc.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Stopping retransmission timer  
for mobile 00:1b:77:42:07:69

*!--- Dot1x got an answer for message 1, so retransmission timeout is stopped.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Sending EAPOL-Key Message to mobile 00:1b:77:42:07:69

state PTKINITNEGOTIATING (message 3), replay counter 00.00.00.00.00.00.01

*!--- Derive PTK; send GTK + MIC.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile 00:1b:77:42:07:69

*!--- Message received from client.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 Received EAPOL-key in PTKINITNEGOTIATING state (message 4) from mobile 00:1b:77:42:07:69

*!--- This signals the start of validation of message 4 (MIC), which  
!--- means client installed the keys. Potential errors after this message  
!--- are MIC validation errors, invalid key types, etc.*

### **PEM Process**

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 8021X\_REQD (3) Change state to L2AUTHCOMPLETE (4) last state L2AUTHCOMPLETE (4)

*!--- PEM receives notification and signals the state machine to change to L2  
!--- authentication completed.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 L2AUTHCOMPLETE (4) Plumbed mobile LWAPP rule on AP 00:1c:0j:ca:5f:c0

*!--- PEM pushes client status and keys to AP through LWAPP component.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 L2AUTHCOMPLETE (4) Change state to DHCP\_REQD (7) last state DHCP\_REQD (7)

*!--- PEM sets the client on address learning status.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7) pemAdvanceState2 4238, Adding TMP rule

*!--- PEM signals NPU to allow DHCP/ARP traffic to be inspected by controller  
!--- for the address learning.*

Wed Oct 31 10:46:15 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7) Adding Fast Path rule

type = Airespace AP - Learn IP address

on AP 00:1c:0j:ca:5f:c0, slot 1, interface = 1, QOS = 0

ACL Id = 255, Jumbo Frames = NO, 802.1P = 0, DSCP = 0, TokenID = 5006

*!--- Entry is built for client and prepared to be forwarded to NPU.  
!--- Type is 9 (see the table in the Client Traffic Forwarding section of  
!--- this document) to allow controller to learn the IP address.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7)  
Successfully plumbed mobile rule (ACL ID 255)

*!--- A new rule is successfully sent to internal queue to add the client  
!--- to the NPU.*

#### **Dot1x Process**

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Stopping retransmission timer  
for mobile 00:1b:77:42:07:69

*!--- Dot1x received message from client.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Sending EAPOL-Key Message to  
mobile 00:1b:77:42:07:69

state PTKINITDONE (message 5 - group), replay counter  
00.00.00.00.00.00.00.02

*!--- Group key update prepared for client.*

#### **PEM Process**

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 Added NPU entry of type 9

*!--- NPU reports that entry of type 9 is added (learning address state).  
!--- See the table in the Client Traffic Forwarding section of this document.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Sent an XID frame

*!--- No address known yet, so the controller sends only XID frame  
!--- (destination broadcast, source client address, control 0xAF).*

#### **Dot1x Process**

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Sent EAPOL-Key M5 for mobile  
00:1b:77:42:07:69

*!--- Key update sent.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile  
00:1b:77:42:07:69

*!--- Key received.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Received EAPOL-key in  
REKEYNEGOTIATING state (message 6) from mobile 00:1b:77:42:07:69

*!--- Successfully received group key update.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Stopping retransmission timer  
for mobile 00:1b:77:42:07:69

*!--- Group key timeout is removed.*

#### **DHCP Process**

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 DHCP received op BOOTREQUEST  
(1) (len 308, port 1, encap 0xec03)

*!--- First DHCP message received from client.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 DHCP dropping packet due to  
ongoing mobility handshake exchange, (siaddr 0.0.0.0, mobility  
state = 'apfMsMmQueryRequested')

#### **PEM Process**

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7) mobility  
role update request from Unassociated to Local

Peer = 0.0.0.0, Old Anchor = 0.0.0.0, New Anchor = 192.168.100.11

*!--- NPU is notified that this controller is the local anchor, so to  
!--- terminate any previous mobility tunnel. As this is a new client,  
!--- old address is empty.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7) State  
Update from Mobility-Incomplete to Mobility-Complete, mobility  
role=Local

*!--- Role change was successful.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7)  
pemAdvanceState2 3934, Adding TMP rule

*!--- Adding temporary rule to NPU for address learning now with new mobility  
!--- role as local controller.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7)  
Replacing Fast Path rule

type = Airespace AP - Learn IP address

on AP 00:1c:0j:ca:5f:c0, slot 1, interface = 1, QOS = 0

ACL Id = 255, Jumbo Frames = NO, 802.1P = 0, DSCP = 0, TokenID = 5006

*!--- Entry is built.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 DHCP\_REQD (7)  
Successfully plumbed mobile rule (ACL ID 255)

*!--- A new rule is successfully sent to internal queue to add the  
!--- client to the NPU.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 0.0.0.0 Added NPU entry of type 9

*!--- Client is on address learning state; see the table in the  
!--- Client Traffic Forwarding section of this document. Now mobility  
!--- has finished.*

Wed Oct 31 10:46:19 2007: 00:1b:77:42:07:69 Sent an XID frame

*!--- No address known yet, so controller sends only XID frame (destination  
!--- broadcast, source client address, control 0xAF).*

### **DHCP Process**

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP received op BOOTREQUEST  
(1) (len 308, port 1, encap 0xec03)

*!--- DHCP request from client.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP selecting relay 1 -  
control block settings:

dhcpServer: 0.0.0.0, dhcpNetmask: 0.0.0.0,

dhcpGateway: 0.0.0.0, dhcpRelay: 0.0.0.0 VLAN: 0

*!--- Based on the WLAN configuration, the controller selects the identity to  
!--- use to relay the DHCP messages.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP selected relay 1 -  
192.168.100.254 (local address 192.168.100.11, gateway 192.168.100.254,  
VLAN 100, port 1)

*!--- Interface selected.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
transmitting DHCP DISCOVER (1)

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
op: BOOTREQUEST, htype: Ethernet, hlen: 6, hops: 1

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
xid: 0xd3d3b6e9 (3553867497), secs: 1024, flags: 0

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
chaddr: 00:1b:77:42:07:69

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
ciaddr: 0.0.0.0, yiaddr: 0.0.0.0

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
siaddr: 0.0.0.0, giaddr: 192.168.100.11

*!--- Debug parsing of the frame sent. The most important fields are included.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP sending REQUEST to  
192.168.100.254 (len 350, port 1, vlan 100)

*!--- DHCP request forwarded.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP selecting relay 2 -  
control block settings:

dhcpServer: 0.0.0.0, dhcpNetmask: 0.0.0.0,

dhcpGateway: 0.0.0.0, dhcpRelay: 192.168.100.11 VLAN: 100

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP selected relay 2 ? NONE

*!--- No secondary server configured, so no additional DHCP request are  
!--- prepared (configuration dependant).*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP received op BOOTREPLY (2)  
(len 308, port 1, encap 0xec00)

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP setting server from OFFER  
(server 192.168.100.254, yiaddr 192.168.100.105)

*!--- DHCP received for a known server. Controller discards any offer not on  
!--- the DHCP server list for the WLAN/Interface.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP sending REPLY to STA  
(len 416, port 1, vlan 100)

*!--- After building the DHCP reply for client, it is sent to AP for forwarding.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP transmitting DHCP OFFER (2)

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
op: BOOTREPLY, htype: Ethernet, hlen: 6, hops: 0

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
xid: 0xd3d3b6e9 (3553867497), secs: 0, flags: 0

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
chaddr: 00:1b:77:42:07:69

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
ciaddr: 0.0.0.0, yiaddr: 192.168.100.105

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
siaddr: 0.0.0.0, giaddr: 0.0.0.0

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP  
server id: 1.1.1.1 rcvd server id: 192.168.100.254

*!--- Debug parsing of the frame sent. The most important fields are included.*

Wed Oct 31 10:46:21 2007: 00:1b:77:42:07:69 DHCP received op BOOTREQUEST (1)  
(len 316, port 1, encap 0xec03)

*!--- Client answers*

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP selecting relay 1 -  
control block settings:

```
                dhcpServer: 192.168.100.254, dhcpNetmask: 0.0.0.0,
                dhcpGateway: 0.0.0.0, dhcpRelay: 192.168.100.11  VLAN: 100

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP selected relay 1 -
    192.168.100.254 (local address 192.168.100.11, gateway 192.168.100.254,
    VLAN 100, port 1)

!--- DHCP relay selected per WLAN config

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP transmitting DHCP REQUEST (3)

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    op: BOOTREQUEST, htype: Ethernet, hlen: 6, hops: 1

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    xid: 0xd3d3b6e9 (3553867497), secs: 1024, flags: 0

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    chaddr: 00:1b:77:42:07:69

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    ciaddr: 0.0.0.0, yiaddr: 0.0.0.0

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    siaddr: 0.0.0.0, giaddr: 192.168.100.11

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    requested ip: 192.168.100.105

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP
    server id: 192.168.100.254  rcvd server id: 1.1.1.1

!--- Debug parsing of the frame sent. The most important fields are included.

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP sending REQUEST to
    192.168.100.254 (len 358, port 1, vlan 100)

!--- Request sent to server.

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP selecting relay 2 -
    control block settings:

                dhcpServer: 192.168.100.254, dhcpNetmask: 0.0.0.0,
                dhcpGateway: 0.0.0.0, dhcpRelay: 192.168.100.11  VLAN: 100

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP selected relay 2 ? NONE

!--- No other DHCP server configured.

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP received op BOOTREPLY
    (2) (len 308, port 1, encap 0xec00)

!--- Server sends a DHCP reply, most probably an ACK (see below).

PEM Process

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 192.168.100.105 DHCP_REQD
```

(7) Change state to RUN (20) last state RUN (20)

*!--- DHCP negotiation successful, address is now known, and client  
!--- is moved to RUN status.*

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 192.168.100.105 RUN (20)  
Reached PLUMBFASPATH: from line 4699

*!--- No L3 security; client entry is sent to NPU.*

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 192.168.100.105 RUN (20)  
Replacing Fast Path rule

type = Airespace AP Client

on AP 00:1c:0j:ca:5f:c0, slot 1, interface = 1, QOS = 0

ACL Id = 255, Jumbo Frames = NO, 802.1P = 0, DSCP = 0, TokenID = 5006

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 192.168.100.105 RUN (20)  
Successfully plumbed mobile rule (ACL ID 255)

#### **DHCP Process**

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 Assigning Address  
192.168.100.105 to mobile

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP sending REPLY to STA  
(len 416, port 1, vlan 100)

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP transmitting DHCP ACK (5)

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
op: BOOTREPLY, htype: Ethernet, hlen: 6, hops: 0

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
xid: 0xd3d3b6e9 (3553867497), secs: 0, flags: 0

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
chaddr: 00:1b:77:42:07:69

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
ciaddr: 0.0.0.0, yiaddr: 192.168.100.105

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
siaddr: 0.0.0.0, giaddr: 0.0.0.0

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 DHCP  
server id: 1.1.1.1 rcvd server id: 192.168.100.254

#### **PEM Process**

Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 192.168.100.105 Added NPU  
entry of type 1

*!--- Client is now successfully associated to controller.  
!--- Type is 1; see the table in the Client Traffic Forwarding  
!--- section of this document.*

```
Wed Oct 31 10:46:25 2007: 00:1b:77:42:07:69 Sending a gratuitous ARP for
192.168.100.105, VLAN Id 100
```

*!--- As address is known, gratuitous ARP is sent to notify.*

## Troubleshooting Examples

### Wrong Client Cipher Configuration

This example shows a client with different capabilities to the AP. The client is probing for the SSID, but as the probe request shows some parameters not supported, the client never proceeds to authentication/association phases. In particular, the problem introduced was a mismatch between the client using WPA, and the AP advertising only WPA2 support:

```
Wed Oct 31 10:51:37 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 23) in 5 seconds
Wed Oct 31 10:51:37 2007: 00:1b:77:42:07:69 apfProcessProbeReq
(apf_80211.c:4057) Changing state for mobile 00:1b:77:42:07:69 on AP
00:1c:b0:ea:5f:c0 from Idle to Probe
```

*!--- Controller adds the new client, moving into probing status*

```
Wed Oct 31 10:51:37 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:38 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:38 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
```

*!--- AP is reporting probe activity every 500 ms as configured*

```
Wed Oct 31 10:51:41 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:41 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:41 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:41 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:44 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:44 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:44 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:44 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:51:49 2007: 00:1b:77:42:07:69 apfMsExpireCallback (apf_ms.c:433)
Expiring Mobile!
Wed Oct 31 10:51:49 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Deleted mobile
LWAPP rule on AP [00:1c:b0:ea:5f:c0]
Wed Oct 31 10:51:49 2007: 00:1b:77:42:07:69 Deleting mobile on AP
00:1c:b0:ea:5f:c0(0)
```

*!--- After 5 seconds of inactivity, client is deleted, never moved into
authentication or association phases.*

## Wrong Preshared Key

This shows client trying to authenticate by WPA-PSK to the infrastructure, but failing due to mismatch of the preshared key between client and controller, resulting on the eventual blacklisting of the client:

```
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Adding mobile on LWAPP AP
00:1c:b0:ea:5f:c0(0)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 23) in 5 seconds
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 apfProcessProbeReq (apf_80211.c:
4057) Changing state for mobile 00:1b:77:42:07:69 on AP 00:1c:b0:ea:5f:c0
from Idle to Probe
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile
Station: (callerId: 24) in 5 seconds
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Association received from mobile
on AP 00:1c:b0:ea:5f:c0
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 STA - rates (8): 130 132 139 150
12 18 24 36 0 0 0 0 0 0 0
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 STA - rates (12): 130 132 139 150
12 18 24 36 48 72 96 108 0 0 0
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Processing WPA IE type 221,
length 24 for mobile 00:1b:77:42:07:69
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 0.0.0.0 START (0)
Initializing policy
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Change state to
AUTHCHECK (2) last state AUTHCHECK (2)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 0.0.0.0 AUTHCHECK (2) Change
state to 8021X_REQD (3) last state 8021X_REQD (3)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 0.0.0.0 8021X_REQD (3) Plumbed
mobile LWAPP rule on AP 00:1c:b0:ea:5f:c0
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 apfPemAddUser2 (apf_policy.c:209)
Changing state for mobile 00:1b:77:42:07:69 on AP 00:1c:b0:ea:5f:c0 from
Probe to Associated
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Stopping deletion of Mobile
Station: (callerId: 48)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Sending Assoc Response to station
on BSSID 00:1c:b0:ea:5f:c0 (status 0)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 apfProcessAssocReq (apf_80211.c:
3838) Changing state for mobile 00:1b:77:42:07:69 on AP 00:1c:b0:ea:5f:c0
from Associated to Associated
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Creating a new PMK Cache Entry
for station 00:1b:77:42:07:69 (RSN 0)
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Initiating WPA PSK to mobile
00:1b:77:42:07:69
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 dot1x - moving mobile
00:1b:77:42:07:69 into Force Auth state
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Skipping EAP-Success to mobile
00:1b:77:42:07:69
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Sending EAPOL-Key Message to
mobile 00:1b:77:42:07:69
state INITPMK (message 1), replay counter 00.00.00.00.00.00.00.00
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile
00:1b:77:42:07:69
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Received EAPOL-key in PKT_START
state (message 2) from mobile 00:1b:77:42:07:69
Wed Oct 31 10:55:55 2007: 00:1b:77:42:07:69 Received EAPOL-key M2 with
invalid MIC from mobile 00:1b:77:42:07:69
Wed Oct 31 10:55:56 2007: 00:1b:77:42:07:69 802.1x 'timeoutEvt' Timer expired
for station 00:1b:77:42:07:69
Wed Oct 31 10:55:56 2007: 00:1b:77:42:07:69 Retransmit 1 of EAPOL-Key M1
(length 99) for mobile 00:1b:77:42:07:69
Wed Oct 31 10:55:56 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile
00:1b:77:42:07:69
Wed Oct 31 10:55:56 2007: 00:1b:77:42:07:69 Received EAPOL-key in PKT_START
state (message 2) from mobile 00:1b:77:42:07:69
```

Wed Oct 31 10:55:56 2007: 00:1b:77:42:07:69 Received EAPOL-key M2 with invalid MIC from mobile 00:1b:77:42:07:69

*!--- MIC error due to wrong preshared key*

Wed Oct 31 10:55:57 2007: 00:1b:77:42:07:69 802.1x 'timeoutEvt' Timer expired for station 00:1b:77:42:07:69

Wed Oct 31 10:55:57 2007: 00:1b:77:42:07:69 Retransmit 2 of EAPOL-Key M1 (length 99) for mobile 00:1b:77:42:07:69

Wed Oct 31 10:55:57 2007: 00:1b:77:42:07:69 Received EAPOL-Key from mobile 00:1b:77:42:07:69

Wed Oct 31 10:55:57 2007: 00:1b:77:42:07:69 Received EAPOL-key in PKT\_START state (message 2) from mobile 00:1b:77:42:07:69

Wed Oct 31 10:55:57 2007: 00:1b:77:42:07:69 Received EAPOL-key M2 with invalid MIC from mobile 00:1b:77:42:07:69

Wed Oct 31 10:55:58 2007: 00:1b:77:42:07:69 802.1x 'timeoutEvt' Timer expired for station 00:1b:77:42:07:69

Wed Oct 31 10:55:58 2007: 00:1b:77:42:07:69 Retransmit failure for EAPOL-Key M1 to mobile 00:1b:77:42:07:69, retransmit count 3, msch deauth count 0

Wed Oct 31 10:55:58 2007: 00:1b:77:42:07:69 Sent Deauthenticate to mobile on BSSID 00:1c:b0:ea:5f:c0 slot 0(caller 1x\_ptsm.c:462)

*!--- Client is deauthenticated, after three retries*

*!--- The process is repeated three times, until client is blacklisted*

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 Blacklisting (if enabled) mobile 00:1b:77:42:07:69

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 apfBlacklistMobileStationEntry2 (apf\_ms.c:3560) Changing state for mobile 00:1b:77:42:07:69 on AP 00:1c:b0:ea:5f:c0 from Associated to Exclusion-list (1)

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 44) in 10 seconds

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 0.0.0.0 8021X\_REQD (3) Change state to START (0) last state 8021X\_REQD (3)

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 0.0.0.0 START (0) Reached FAILURE: from line 3522

Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 Scheduling deletion of Mobile Station: (callerId: 9) in 10 seconds

---

## Related Information

- [Lightweight Access Point FAQ](#)
- [Wireless LAN Controller \(WLC\) Troubleshoot FAQ](#)
- [Cisco Wireless LAN Controller Module Q&A](#)
- [Cisco Wireless LAN Controllers Q&A](#)
- [Radio Resource Management under Unified Wireless Networks](#)
- [Wireless LAN \(WLAN\) Technology Support](#)
- [Technical Support & Documentation – Cisco Systems](#)

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Oct 13, 2009

Document ID: 100260

---