Informazioni sul client di debug sui controller LAN wireless (WLC)

Sommario

Introduzione

Questo documento descrive in dettaglio le **debug client** sui controller WLC (Wireless LAN Controller).

Prerequisiti

Requisiti

Questo documento tratta i seguenti argomenti:

- Come viene gestito un client wireless
- Come risolvere i problemi relativi alle associazioni di base e all'autenticazione

L'output da analizzare copre lo scenario per una rete WPA a chiave precondivisa (WPA-PSK).

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Configurazione del WLC e del Lightweight Access Point (LAP) per un funzionamento di base
- Metodi LWAPP (Lightweight Access Point Protocol) e di sicurezza wireless
- Funzionamento dei processi di autenticazione e associazione 802.11

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- WLC Cisco AireOS (8540, 5520, vWLC) con firmware 8.5 o 8.10.
- Access point CAPWAP.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Client di debug

Il comando **debug client**
è una macro che abilita otto comandi di debug, più un filtro sull'indirizzo MAC fornito, in modo che vengano visualizzati solo i messaggi che contengono l'indirizzo MAC specificato. Gli otto comandi di debug mostrano i dettagli più importanti sull'associazione e l'autenticazione dei client. Il filtro è utile in situazioni in cui sono presenti più client wireless. Situazioni quali la generazione di un output eccessivo o il sovraccarico del controller quando il debug è abilitato senza il filtro.

Le informazioni raccolte riguardano dettagli importanti relativi all'associazione e all'autenticazione dei client (con due eccezioni menzionate più avanti in questo documento).

I comandi attivati sono mostrati in questo output:

<#root>

```
(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.
```

Questi comandi riguardano la negoziazione degli indirizzi, il computer dello stato client 802.11, l'autenticazione 802.1x, il modulo di imposizione dei criteri (PEM) e la negoziazione degli indirizzi (DHCP).

Varianti client di debug

Per la maggior parte degli scenari, **debug client** è sufficiente per ottenere le informazioni necessarie. Tuttavia, sono due le situazioni importanti in cui è necessario eseguire un ulteriore debug:

- Mobilità (roaming client tra controller)
- Risoluzione dei problemi di autenticazione EAP

Mobilità

In questo caso, i debug sulla mobilità devono essere attivati dopo **debug client** è stato introdotto per ottenere ulteriori informazioni sull'interazione del protocollo di mobilità tra i controller.

Nota: i dettagli su questo output sono illustrati in altri documenti.

Per abilitare i debug relativi alla mobilità, utilizzare il **debug client** e quindi utilizzare il comando **debug mobility handoff enable** comando:

<#root>

```
(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.
```

Risoluzione dei problemi di autenticazione EAP

Per risolvere i problemi di interazione tra il WLC e il server di autenticazione (server RADIUS esterno o EAP interno), utilizzare il comando **debug AAA all enable** , in cui vengono visualizzati i dettagli richiesti. Questo comando viene utilizzato dopo **debug client** e può essere combinato con altri comandi di debug se necessario (ad esempio, **handoff**).

<#root>

```
(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.
```

Connessione client

Ai fini del presente documento, per *connessione client* si intende il processo attraverso il quale un client wireless effettua le seguenti operazioni:

Sezione 802.11

1. Probe, per trovare un punto di accesso valido da associare.
2. Autenticazione: può essere aperta (null) o condivisa. Normalmente, l'opzione Apri è selezionata.
3. Associazione: richiedere i servizi dati all'access point.

Sezione Criteri L2

1. Nessuna; l'autenticazione PSK o EAP viene eseguita in base alla configurazione.

2. Negoziazione chiave, se è selezionato un metodo di crittografia.

Sezione Criteri L3

1. Informazioni sull'indirizzo.
2. autenticazione Web, se selezionata.

> **Nota**: questi passi rappresentano un sottoinsieme o un riepilogo dell'intero processo. Questo documento descrive uno scenario semplificato che copre le policy 802.11 e L2 e utilizza WPA-PSK, oltre ad indirizzare l'apprendimento. Non vengono utilizzati criteri AAA o L3 esterni per l'autenticazione.

Processi controller

In ogni sezione, il controller utilizza processi separati per tenere traccia dello stato del client in ogni momento. I processi interagiscono tra loro per garantire che il client venga aggiunto alla tabella di connessione (in base ai criteri di sicurezza configurati). Per comprendere le fasi di connessione del client al controller, di seguito è riportato un breve riepilogo dei processi più rilevanti:

- **Policy Enforcement Module (PEM)**: controlla lo stato del client e lo forza tramite ciascuno dei criteri di sicurezza sulla configurazione WLAN.
- **Access Point Functions (APF)** - In pratica, la macchina a stati 802.11.
- **Dot1x**: implementa la macchina a stati per l'autenticazione 802.1x, PSK e l'handle della chiave per i client wireless.
- **Mobilità**: tiene traccia dell'interazione con altri controller dello stesso gruppo di mobilità.
- **DTL (Data Transformation Layer)** - Posizionato tra i componenti software e la NPU (Network Hardware Acceleration). Controlla le informazioni ARP.

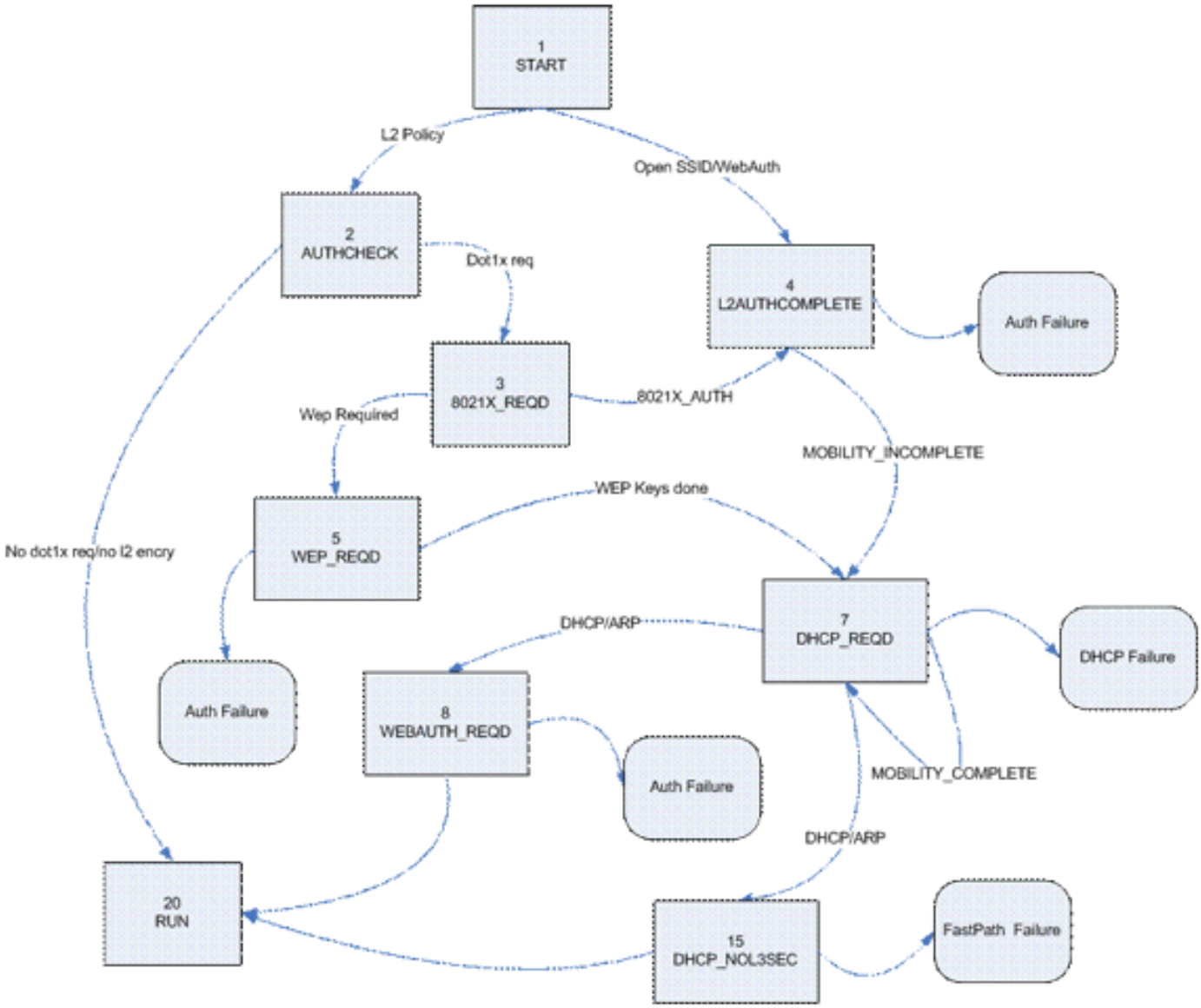
PEM (Policy Enforcement Module)

In base alla configurazione WLAN, il client esegue una serie di passaggi. PEM garantisce che questa operazione venga eseguita in modo da garantire la conformità ai criteri di protezione L2 e L3 richiesti.

Di seguito è riportato un sottoinsieme degli stati PEM rilevanti per l'analisi del debug di un client:

- **START**: stato iniziale della nuova voce del client.
- **AUTHCHECK**: la WLAN dispone di un criterio di autenticazione L2 da applicare.
- **8021X_REQD**: il client deve completare l'autenticazione 802.1x.
- **L2AUTHCOMPLETE**: il client ha completato il criterio L2. Il processo può ora passare alle regole L3 (apprendimento degli indirizzi, autenticazione Web, ecc.). Il controller invia qui l'annuncio di mobilità per apprendere informazioni L3 da altri controller se si tratta di un roaming client nello stesso gruppo di mobilità.
- **WEP_REQD** - Il client deve completare l'autenticazione WEP.
- **DHCP_REQD**: il controller deve imparare l'indirizzo L3 dal client, operazione che viene eseguita tramite richiesta ARP, richiesta DHCP o rinnovo o tramite informazioni ottenute da un altro controller nel gruppo di mobilità. Se DHCP Required è contrassegnato sulla WLAN, vengono utilizzate solo le informazioni DHCP o sulla mobilità.
- **WEBAUTH_REQD**: il client deve completare l'autenticazione Web. (criterio L3)
- **RUN**: il client ha completato i criteri L2 e L3 richiesti e può ora trasmettere il traffico alla rete.

Nell'immagine è illustrata una macchina a stati PEM semplificata con le transizioni client fino al raggiungimento dello stato RUN, dove il client può ora inviare il traffico alla rete:



Nota: questa figura non copre tutte le transizioni e gli stati possibili. Alcuni passaggi intermedi sono stati rimossi per maggiore chiarezza.

Inoltro traffico client

Tra lo stato START e prima dello stato RUN finale, il traffico del client non viene inoltrato alla rete, ma viene passato alla CPU principale sul controller per l'analisi. Le informazioni che vengono inoltrate dipendono dallo stato e dai criteri in uso; ad esempio, se è abilitato 802.1x, il traffico EAPOL viene inoltrato alla CPU. Se si utilizza l'autenticazione Web, vengono inoltre consentiti i protocolli HTTP e DNS, intercettati dalla CPU per eseguire il reindirizzamento Web e ottenere le credenziali di autenticazione del client.

Quando il client raggiunge lo stato RUN, le informazioni client vengono inviate alla NPU per abilitare la commutazione FastPath, che esegue l'inoltro del traffico utente alla VLAN client alla velocità di trasmissione e libera la CPU centrale dalle attività di inoltro dei dati utente.

Il traffico inoltrato dipende dal tipo di client applicato alla NPU. In questa tabella vengono descritti i tipi più rilevanti.

Tipo	Descrizione

1	Inoltro del traffico client normale.
9	Stato di apprendimento IP. Un pacchetto di questo client viene inviato alla CPU per conoscere l'indirizzo IP usato.
2	Pass-through ACL. Utilizzato quando la WLAN è un ACL configurato per informare la NPU.

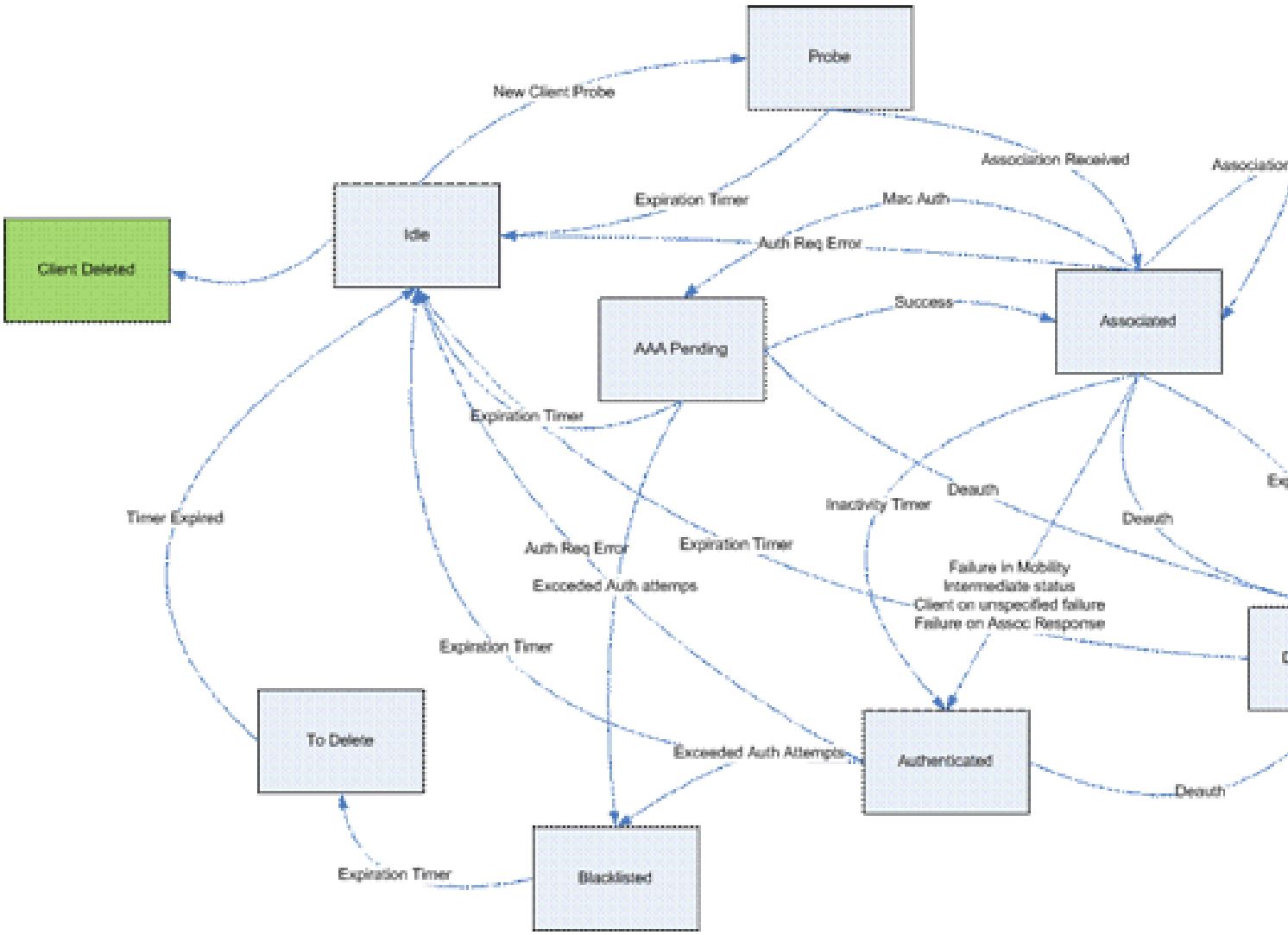
Funzioni Access Point (APF)

Questo processo gestisce lo stato del client attraverso lo stato del computer 802.11 e interagisce con il codice di mobilità per convalidare i diversi scenari di roaming. Il presente documento non riguarda i dettagli della mobilità né i suoi stati.

Nella tabella seguente vengono illustrati gli stati client più rilevanti che possono verificarsi quando un client è associato al controller:

Nome	Descrizione
Inattivo	Nuovo client o stato temporaneo in alcune situazioni.
Ciondolo AAA	Il client attende l'autenticazione dell'indirizzo MAC.
Autenticato	Autenticazione aperta riuscita o stato intermedio in alcune situazioni.
Associato	Il client ha superato i processi di autenticazione MAC e di apertura.
Non associato	Il client ha inviato la disassociazione/deautenticazione o il timer di associazione è scaduto.
Per eliminare	Client contrassegnato per l'eliminazione (in genere dopo la scadenza del timer di esclusione).
Sonda	Richiesta probe ricevuta per il nuovo client.
Escluso/Bloccato elencato	Il client è stato contrassegnato come escluso. Normalmente correlato ai criteri WPS.
Non valido	Errore nello stato del client.

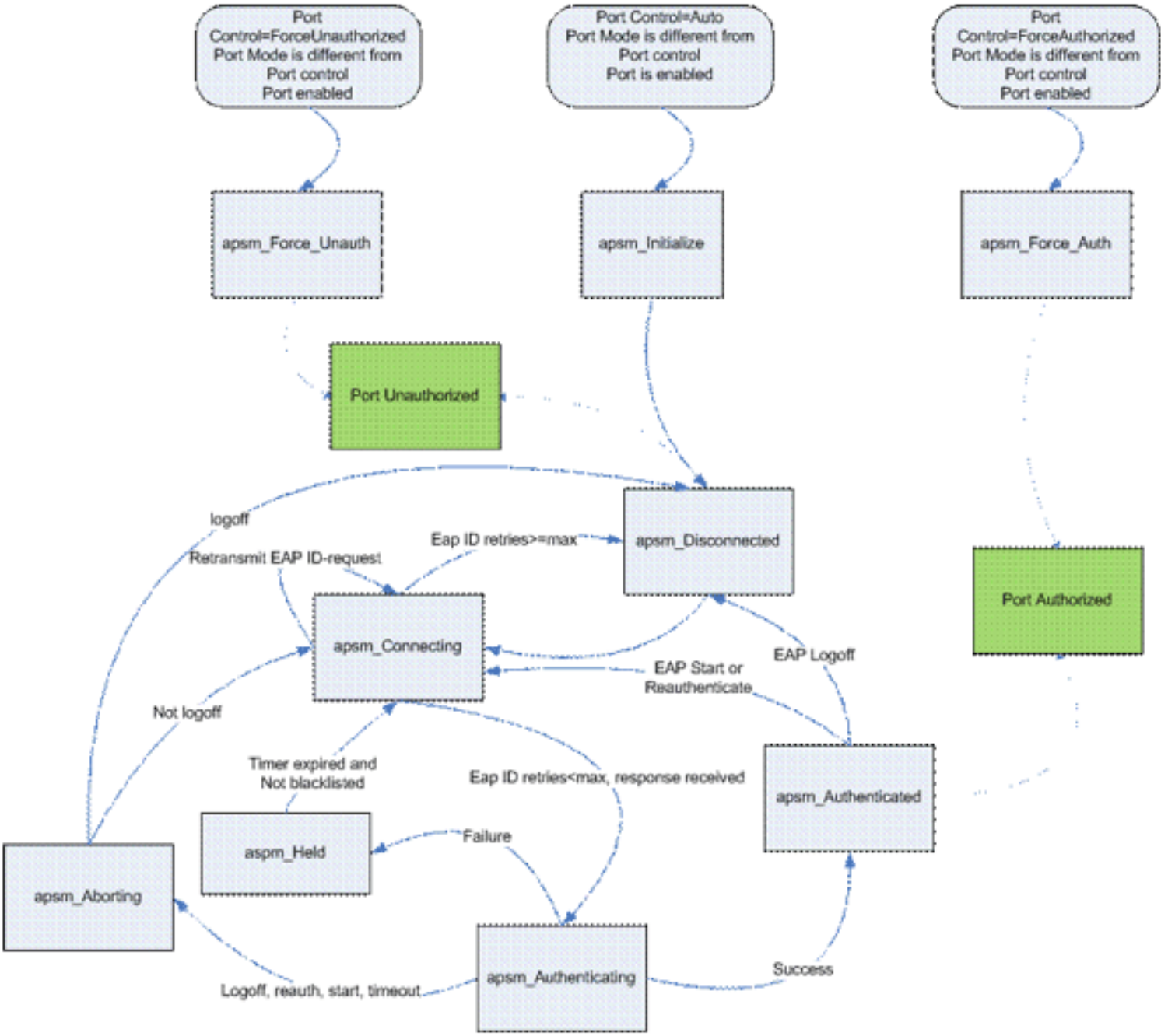
Questa immagine rappresenta una transizione di macchina a stati e mostra solo gli stati e le transizioni più rilevanti:



Autenticazione 802.1x (Dot1x)

Il processo Dot1x è responsabile dell'autenticazione 802.1x e della gestione delle chiavi per il client. Ciò significa che, anche sulle WLAN che non dispongono di una policy EAP che richiede 802.1x, il dot1x partecipa alla creazione e alla negoziazione delle chiavi con il client e anche per la gestione delle chiavi memorizzate nella cache (PMK o CCKM).

Questa macchina a stati mostra le transizioni 802.1x complete:



Debug analisi client

In questa sezione viene mostrato l'intero processo nei log quando un client si connette a una WLAN.

<#root>

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 that depend 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 0

!--- Controller saves the client supported rates into its connection table.
!--- Units are values of 500 kbps, basic (mandatory) rates have the Most Significant bit (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
    00.00.00.00.00.00.00.01

!--- 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: x.x.x.x  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: x.x.x.x

!--- 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 PLUMBFASTPATH: 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: x.x.x.x  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.
```

Risoluzione dei problemi relativi agli esempi

Configurazione della crittografia del client errata

Nell'esempio viene mostrato un client con funzionalità diverse rispetto all'access point. Il client richiede l'SSID, ma poiché la richiesta di verifica mostra alcuni parametri non supportati, non passa mai alle fasi di autenticazione/associazione.

In particolare, è stata introdotta una mancata corrispondenza tra il client che utilizza WPA e l'AP che annuncia solo il supporto WPA2:

```
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.
```

Chiave già condivisa errata

Ciò mostra che il client tenta di eseguire l'autenticazione da parte di WPA-PSK all'infrastruttura, ma l'operazione non riesce a causa di una mancata corrispondenza della chiave già condivisa tra il client e il controller, che determina l'eventuale aggiunta del client all'elenco di esclusione (blocco):

```
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 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 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, mscb 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 block listed


Wed Oct 31 10:56:10 2007: 00:1b:77:42:07:69 Block listing (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
```

Informazioni correlate

- **Supporto tecnico e download Cisco**

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).