



トラブルシューティングおよびベスト プラクティス

このマニュアルでは、特定の機能を実装する際に必要となる可能性のあるその他のトラブルシューティングとベスト プラクティスを示し、説明します。

このマニュアルで説明する機能は次のとおりです。

- 「Cisco Compatible Extensions バージョン 5 クライアント デバイスのトラブルシューティング」 (P.A-1)
- 「WLAN 上の Web 認証セキュリティ」 (P.A-2)

Cisco Compatible Extensions バージョン 5 クライアント デバイスのトラブルシューティング

Cisco Compatible Extension クライアントとの通信の問題をトラブルシューティングするために、診断チャンネルとクライアント レポートの 2 つの機能が設計されています。



(注) これらの機能は、Cisco Compatible Extensions バージョン 5 クライアント デバイスだけでサポートされています。Cisco Compatible Extensions バージョン 5 クライアント デバイス以外での使用、および以前のバージョンが稼動しているクライアントでの使用はサポートされていません。

診断チャンネル

診断チャンネル機能によって、WLAN とのクライアント通信に関する問題をトラブルシューティングすることができます。困難を抱えたクライアントによって起動する場合、診断チャンネルは WLAN です。これは、クライアントのパスに置かれた通信への障害物が最も少なく、最も堅牢な通信方法が提供されるように設定されます。クライアントが経験した通信の困難の原因を特定する試行において、定義済みのテストのセットをクライアントとアクセス ポイントに受けさせることができます。



(注) コントローラごとに 1 つの WLAN しか診断チャンネルを有効にできず、この WLAN のセキュリティはすべて無効です。

診断チャネルの設定

診断チャネルを設定する手順は、次のとおりです。

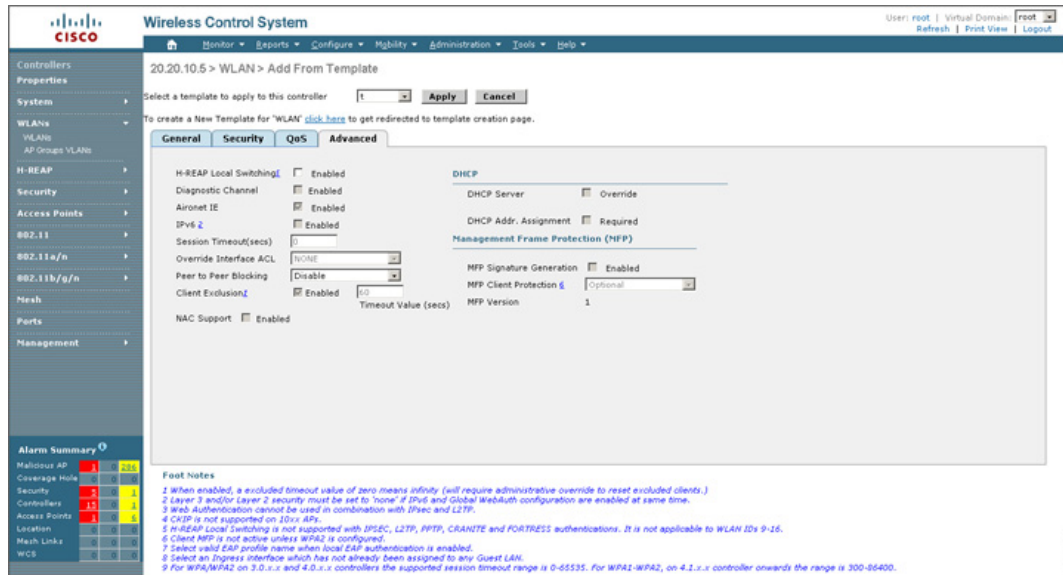
- ステップ 1 [Configure] > [Controllers] の順に選択します。
- ステップ 2 IP アドレスをクリックして、特定のコントローラを選択します。
- ステップ 3 左側のサイドバーのメニューから、[WLAN] > [WLANs] の順に選択します。
- ステップ 4 [Select a command] ドロップダウンメニューから [Add WLAN] を選択して新しい WLAN を作成するか、または既存の WLAN のプロファイル名をクリックします。



(注) 新しい WLAN を作成して診断テストを実行することをお勧めします。

- ステップ 5 [WLANs] ページが表示されたら、[Advanced] タブをクリックします (図 A-1 を参照)。

図 A-1 [WLANs Advanced] タブ



- ステップ 6 この WLAN 上で診断チャネルを有効にする場合は、[Diagnostic Channel] チェックボックスをオンにします。そうでない場合には、このチェックボックスをデフォルト値のオフのままにします。
- ステップ 7 [Save] をクリックして、変更内容を確定します。

WLAN 上の Web 認証セキュリティ

この項では、WLAN に Web 認証セキュリティを実装する場合に役立つトラブルシューティングとベスト プラクティスの手順を説明します。

Web 認証は、WLAN 上のユーザへの Web ベースの認証を可能にするレイヤ 3 のセキュリティ機能です。主にゲスト ネットワークのシナリオで使用されますが、これに限定されるわけではありません。

WLAN が Web 認証セキュリティで設定されると、ユーザはレイヤ 2 認証（静的 WEP、WPA+PSK、MAC フィルタリングなど）の通過後にログイン ページにリダイレクトされます。ログイン ページは、ローカル デバイスまたは外部 Web サーバに格納され、ロゴ、タイトルなどのカスタマイズが許可されるように変更できます。

WLAN が Web 認証 WLAN で設定された後は、要求された Web サイトに、無線クライアントによって HTTP *get* 要求が送信されます。コントローラ ファイアウォールによって、指定された URL の DNS レゾリューションが許可されます。レゾリューション後に、コントローラが無線クライアントからの HTTP パケットに割り込んで、ログイン ページにリダイレクトします。クレデンシャルがログイン ページに入力および送信されると、ローカル データベースに対して認証されます。ユーザがローカル データベースで検出されない場合、設定された RADIUS サーバに接触します。



(注) クライアントと認証エージェント間には、PAP および CHAP 認証が使用されます。RADIUS サーバがこれらのプロトコルをいずれもサポートしていて、Web 認証が許されていることを確認してください。

認証が成功すると、トラフィックを通過させることができます。認証の試行が 3 回失敗すると、クライアントは除外されます。除外されたクライアントは、除外タイムアウト制限を超えるまでアソシエートできません。除外タイムアウト制限は、積極的な負荷分散で設定されます。積極的な負荷分散は、モバイルクライアントとアソシエートされたアクセス ポイントの間で負荷をアクティブに分散させます。

Web 認証 WLAN は、事前認証の Access Control List (ACL; アクセス コントロール リスト) でも設定されます。この ACL は通常の ACL 同様に設定されますが、クライアントが認証に先立って必要とするリソースへのアクセスを許可します。管理者は、インターフェイス セクションを使用して認証後に ACL をクライアントに適用する必要があります。

Web 認証 WLAN は、セッションのタイムアウト値でも設定できます。この値によって、クライアントが再度デバイスの認証を行うのに必要な時間が定義されます。値が 0 に設定される場合、これは無限を意味するので、クライアントはログアウト オプションが使用されるまで再度認証されることはありません。http://<VirtualIP>/logout.html でログアウト URL にアクセスできます。



(注) ログアウト ウィンドウを表示するには、クライアント上のポップアップ ブロッカーをすべてオフにします。

Web 認証は、レイヤ 3 セキュリティ下の異なるモードに設定することができます。最も一般的に使用される Web 認証のモードは、次のとおりです。

- Internal Web : http://<virtual IP /DNS name >/login.html を使用した、内部ページへのリダイレクション。カスタマイズ可能です。
- External Web : 外部 URL へのリダイレクション。

デバッグ コマンド

次のデバッグ コマンドが使用できます。

デバッグ戦略

ゲスト トンネリングなしで WLAN に設定された Web 認証に対して、次の戦略を使用します。

ステップ 1 無線 MAC アドレスを使用し書き込むモバイル クライアントを特定します。MS Windows ベースのすべてのシステムにコマンドプロンプト > **ipconfig /all** を使用します。

ステップ 2 モバイル クライアントの無線を無効にします。

ステップ 3 高速 (115200) 用シリアル コンソール セットまたはコントローラの管理ポートへの SSH セッション経由で次のデバッグ コマンドを有効にします。

```
debug client <client-mac-address>
debug pm ssh-tcp enable
debug pm ssh-appgw enable
debug pm rules enable
debug pm config enable

show client detail <client-mac-address>

debug pem event enable
debug pem state enable
```

ステップ 4 無線を有効にし、クライアントをアソシエートさせます。クライアントがアソシエートされてから、**show client detail <client-mac-address>** を入力します。

```
Client Username ..... N/A
AP MAC Address..... 00:0b:85:09:96:10
Client State..... Associated
Wireless LAN Id..... 1
BSSID..... 00:0b:85:09:96:1f
Channel..... 11
IP Address..... 10.50.234.3
Association Id..... 1
Authentication Algorithm..... Open System
Reason Code..... 0
Status Code..... 0
Session Timeout..... 0
Client CCX version..... 3
Mirroring..... Disabled
QoS Level..... Silver
Diff Serv Code Point (DSCP)..... disabled
802.1P Priority Tag..... disabled
WMM Support..... Disabled
Mobility State..... Local
Internal Mobility State..... apfMsMmInitial
Mobility Move Count..... 0
--More-- or (q)uit
Security Policy Completed..... No
Policy Manager State..... WEBAUTH_REQD =====**
Policy Manager Rule Created..... Yes
NPU Fast Fast Notified..... Yes
Last Policy Manager State..... WEBAUTH_REQD
Client Entry Create Time..... 67733 seconds
Policy Type..... N/A
Encryption Cipher..... None
Management Frame Protection..... No
EAP Type..... Unknown
Interface..... management
VLAN..... 0
Client Capabilities:
  CF Pollable..... Not implemented
  CF Poll Request..... Not implemented
  Short Preamble..... Implemented
  PBCC..... Not implemented
  Channel Agility..... Not implemented
  Listen Interval..... 0
```

```

Client Statistics:
  Number of Bytes Received..... 188595
  Number of Bytes Sent..... 19229
  Number of Packets Received..... 3074
--More-- or (q)uit
  Number of Packets Sent..... 76
  Number of Policy Errors..... 0
  Radio Signal Strength Indicator..... -41 dBm
  Signal to Noise Ratio..... 59 dB
Nearby AP Statistics:
  TxExcessiveRetries: 0
  TxRetries: 0
  RtsSuccessCnt: 0
  RtsFailCnt: 0
  TxFiltered: 0
  TxRateProfile: [0,0,0,0,0,0,0,0,0,0,0,0]
  ap:09:96:10(slot 1) .....
antenna0: 48 seconds ago -45 dBm..... antenna1: 123 seconds ago -128 dBm

```

ステップ 5 クライアントの PEM 状態が WEBAUTH_REQD であることを確認します。クライアントのブラウザページを開いて、次のメッセージを探します。

```

Wed Mar 7 17:59:15 2007: ***** sshpmAddWebRedirectRules: POLICY SEMAPHORE LOCKED
*****
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: mobile station addr is 10.50.234.3
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: RuleID for ms 10.50.234.3 is 44
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: using HTTP-S for web auth (addr:
10.50.234.15).
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: inbound local http rule created for ms
10.50.234.3 local 1.1.1.1.
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: inbound http redirect rule created.
Wed Mar 7 17:59:15 2007: sshpmRuleIndexInsert: adding rule for RuleID 44
Wed Mar 7 17:59:15 2007: sshpmRuleIndexInsert: computed raw hash index 02ad3271 for rule
id 0000002c
Wed Mar 7 17:59:15 2007: sshpmRuleIndexInsert: computed adjusted index 00000c32 for rule
id 0000002c
Wed Mar 7 17:59:15 2007: sshpmAddWebRedirectRules: committing rules for ms 10.50.234.3
Wed Mar 7 17:59:15 2007: ***** sshpmPolicyCommitCallback: POLICY SEMAPHORE
UNLOCKED - [unconditionally] *****
Wed Mar 7 17:59:15 2007: sshpmPolicyCommitCallback: called; ContextPtr: 0x2c; Success: 1
Wed Mar 7 17:59:15 2007: ***** sshpmPolicyCommitCallback: POLICY SEMAPHORE
UNLOCKED - [unconditionally] *****
Wed Mar 7 18:02:32 2007: SshPmAppgw/pm_appgw.c:1234/ssh_pm_appgw_request: New application
gateway request for `alg-http@ssh.com': 10.50.234.3.1153 > 10.50.234.1.80 (nat:
10.50.234.1.80) tcp ft=0x00000000 tt=0x00000000
Wed Mar 7 18:02:32 2007: SshPmAppgw/pm_appgw.c:1239/ssh_pm_appgw_request: Packet
attributes: trigger_rule=0x4ecb, tunnel_id=0x0, trd_index=0xddffffff,
prev_trd_index=0xddffffff
Wed Mar 7 18:02:32 2007: SshPmAppgw/pm_appgw.c:1240/ssh_pm_appgw_request: Packet:
Wed Mar 7 18:02:32 2007: 00000000: 4500 0030 0308 4000 8006 0f57 0a32 ea03
E..0..@....W.2..
Wed Mar 7 18:02:32 2007: 00000010: 0a32 ea01 0481 0050 2f42 e3a4 0000 0000
.2.....P/B.....
Wed Mar 7 18:02:32 2007: 00000020: 7002 4000 42fe 0000 0204 05b4 0101 0402
p.@.B.....
Wed Mar 7 18:02:32 2007: SshPmStAppgw/pm_st_appgw.c:403/ssh_pm_st_appgw_start: Calling
redirection callback
Wed Mar 7 18:02:32 2007: SshPmAppgw/pm_appgw.c:155/ssh_appgw_redirect: Application
gateway redirect: 10.50.234.1.80 -> 10.50.234.1.80
Wed Mar 7 18:02:32 2007: SshPmStAppgw/pm_st_appgw.c:445/ssh_pm_st_appgw_mappings:
Creating application gateway mappings: 10.50.234.3.1153 > 10.50.234.1.80 (10.50.234.1.80)
Wed Mar 7 18:02:32 2007: SshPmStAppgw/pm_st_appgw.c:102/ssh_pm_appgw_mappings_cb: appgw
connection cached: init flow_index=5967 resp flow_index=5964 event_cnt=718

```

```

Wed Mar 7 18:02:32 2007: SshPmStAppgw/pm_st_appgw.c:493/ssh_pm_st_appgw_mappings_done:
NAT on initiator side
Wed Mar 7 18:02:32 2007:
SshPmStAppgw/pm_st_appgw.c:583/ssh_pm_st_appgw_tcp_responder_stream_done:
ssh_pm_st_appgw_tcp_responder_stream_done: conn->context.responder_stream=0x0
Wed Mar 7 18:02:32 2007:
SshPmStAppgw/pm_st_appgw.c:624/ssh_pm_st_appgw_tcp_responder_stream_done: Opening
initiator stream 10.50.234.1:61611 > 10.76.108.121:2024
Wed Mar 7 18:02:32 2007: SshPmStAppgw/pm_st_appgw.c:154/ssh_pm_appgw_i_flow_enabled:
Initiator flow mode has now been set.
Wed Mar 7 18:02:32 2007: SshPmAppgw/pm_appgw.c:507/ssh_appgw_tcp_listener_callback: New
initiator stream: src=10.50.234.1:61611, dst=10.76.108.121:2024
Wed Mar 7 18:02:32 2007:
SshPmStAppgw/pm_st_appgw.c:646/ssh_pm_st_appgw_tcp_open_initiator_stream: Initiator stream
opened
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:531/ssh_appgw_http_conn_cb: New TCP
HTTP connection 10.50.234.3.1153 > 10.50.234.1.80
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:535/ssh_appgw_http_conn_cb: Responder
sees initiator as `10.50.234.15.1153'
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:539/ssh_appgw_http_conn_cb: Initiator
sees responder as `10.50.234.1.80'
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (i) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:32 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
-1 bytes (offset 0 data 0)
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (r) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:32 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:132/ssh_appgw_http_st_wait_input:
appgw_http.c.132: io->src is NULL
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
-1 bytes (offset 0 data 0)
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (i) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:32 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
-1 bytes (offset 0 data 0)
Wed Mar 7 18:02:32 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (r) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:32 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:36 2007: SshAppgwHttp/appgw_http.c:132/ssh_appgw_http_st_wait_input:
appgw_http.c.132: io->src is NULL
Wed Mar 7 18:02:36 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
-1 bytes (offset 0 data 0)
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (i) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
283 bytes (offset 0 data 0)
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 283
bytes:
Wed Mar 7 18:02:41 2007: 00000000: 4745 5420 2f20 4854 5450 2f31 2e31 0d0a GET /
HTTP/1.1..
Wed Mar 7 18:02:41 2007: 00000010: 4163 6365 7074 3a20 696d 6167 652f 6769 Accept:
image/gi
Wed Mar 7 18:02:41 2007: 00000020: 662c 2069 6d61 6765 2f78 2d78 6269 746d f,
image/x-xbitm

```

```
Wed Mar 7 18:02:41 2007: 00000030: 6170 2c20 696d 6167 652f 6a70 6567 2c20 ap,
image/jpeg,
Wed Mar 7 18:02:41 2007: 00000040: 696d 6167 652f 706a 7065 672c 2061 7070 image/pjpeg,
app
Wed Mar 7 18:02:41 2007: 00000050: 6c69 6361 7469 6f6e 2f78 2d73 686f 636b
lication/x-shock
Wed Mar 7 18:02:41 2007: 00000060: 7761 7665 2d66 6c61 7368 2c20 2a2f 2a0d wave-flash,
*/*.
Wed Mar 7 18:02:41 2007: 00000070: 0a41 6363 6570 742d 4c61 6e67 7561 6765
.Accept-Language
Wed Mar 7 18:02:41 2007: 00000080: 3a20 656e 2d75 730d 0a41 6363 6570 742d :
en-us..Accept-
Wed Mar 7 18:02:41 2007: 00000090: 456e 636f 6469 6e67 3a20 677a 6970 2c20 Encoding:
gzip,
Wed Mar 7 18:02:41 2007: 000000a0: 6465 666c 6174 650d 0a55 7365 722d 4167
deflate..User-Ag
Wed Mar 7 18:02:41 2007: 000000b0: 656e 743a 204d 6f7a 696c 6c61 2f34 2e30 ent:
Mozilla/4.0
Wed Mar 7 18:02:41 2007: 000000c0: 2028 636f 6d70 6174 6962 6c65 3b20 4d53 (compatible;
MS
Wed Mar 7 18:02:41 2007: 000000d0: 4945 2036 2e30 3b20 5769 6e64 6f77 7320 IE 6.0;
Windows
Wed Mar 7 18:02:41 2007: 000000e0: 4e54 2035 2e31 3b20 5356 3129 0d0a 486f NT 5.1;
SV1)..Ho
Wed Mar 7 18:02:41 2007: 000000f0: 7374 3a20 3130 2e35 302e 3233 342e 310d st:
10.50.234.1.
Wed Mar 7 18:02:41 2007: 00000100: 0a43 6f6e 6e65 6374 696f 6e3a 204b 6565 .Connection:
Keep-Alive
Wed Mar 7 18:02:41 2007: 00000110: 702d 416c 6976 650d 0a0d 0a p=Alive....
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:985/ssh_appgw_parse_request_line: parsing request
line GET / HTTP/1.1
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:1018/ssh_appgw_parse_request_line: internal http
version 3
Wed Mar 7 18:02:41 2007: SshAppgwHttpState/appgw_http_state.c:1155/ssh_appgw_add_method:
caching method 2 for reply 0
Wed Mar 7 18:02:41 2007: SshAppgwHttpState/appgw_http_state.c:1604/ssh_appgw_check_msg:
examining request using service id 34
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:594/ssh_appgw_http_get_dst_host: destination host:
10.50.234.1
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:1474/ssh_appgw_inject_reply: injecting 404 reply as
msg 0
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:284/ssh_appgw_http_st_write_data:
entering state st_write_data
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (i) reading_hdr 1 nmsgs 1
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:136/ssh_appgw_http_st_wait_input: read
-1 bytes (offset 0 data 0)
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (r) reading_hdr 1 nmsgs 0
Wed Mar 7 18:02:41 2007:
SshAppgwHttpState/appgw_http_state.c:1851/ssh_appgw_http_is_inject: next inject is msg# 0
current msg# 0
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:207/ssh_appgw_http_st_inject: entering
state st_inject (r): msgs 0
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:259/ssh_appgw_http_st_inject: closing
connection after inject
Wed Mar 7 18:02:41 2007: SshAppgwHttp/appgw_http.c:400/ssh_appgw_http_st_terminate:
entering state st_terminate (r): teardown 0 terminate i: 1 r: 1
```

```

Wed Mar  7 18:02:45 2007: SshAppgwHttp/appgw_http.c:99/ssh_appgw_http_st_wait_input:
entering state st_wait_input: (i) reading_hdr 1 nmsgs 1
Wed Mar  7 18:02:45 2007:
SshAppgwHttpState/appgw_http_state.c:2077/ssh_appgw_http_handle_state: handling: 0 bytes:
Wed Mar  7 18:02:45 2007: SshAppgwHttp/appgw_http.c:400/ssh_appgw_http_st_terminate:
entering state st_terminate (i): teardown 0 terminate i: 1 r: 1
Wed Mar  7 18:02:45 2007:
SshAppgwHttp/appgw_http.c:732/ssh_appgw_http_connection_terminate: service HTTP-REDIR: TCP
HTTP connection 10.50.234.3.1153 > 10.50.234.1.80 terminated
Wed Mar  7 18:02:45 2007: SshPmStAppgw/pm_st_appgw.c:1094/ssh_pm_st_appgw_terminate:
terminating appgw instance

```

”

- ステップ 6** HTTP GET メッセージが表示されない場合、HTTP パケットがコントローラに到達していません。クライアントがリダイレクションを完了してから、ログインを入力して送信します。
- ステップ 7** NPUdevshell hapiMmcDebugScbInfoShow へのクライアントのエントリ（「クライアント MAC アドレス」）を確認します。PEM 状態が WEBAUTH_REQD to RUN から動かない場合、クレデンシャルの問題が存在します。ローカルまたは RADIUS データベース内（設定場所に関わらず）のクレデンシャルを確認します。
- ステップ 8** クライアントに RUN 状態が表示される場合、クライアントからゲートウェイへの確認を行い、トラブルシューティングが通過中かどうか確認します。

ベスト プラクティス

クライアントがログイン ページにリダイレクトされず、ネットワーク内の DNS 解決を避けたい場合、**http://<controller-mgmt-ip>** を入力します。リダイレクションが発生する場合、この問題はネットワーク関連ではありません。

config network web-auth-port <Port> を入力して、コントローラに標準の HTTP ポート (80) 以外のポートを定義します。コントローラは、ポートが割り込みに設定されていても、セキュア HTTP または HTTPS(443) に割り込みません。