HiveOS\_Capwap over HTTP\_TestCase

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 05/10/2010 | Yun Feng | Initial version |
| 0.2 | 2011-10-19 | Zhao haihui | Add a case Ft\_Capwap\_discovery\_8 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Content

Glossary and Abbreviations

# Introduction

On current customer side, most of network manage will deny 12222 port, but HTTP port will be allowed by default. So we will avoid this issue, then design this feature to suit the requirement.

Function Spec: <http://saturn.aerohive.com/view.php?fDocumentId=2779>

Unit Test Case: <http://saturn.aerohive.com/view.php?fDocumentId=2838>

CLI Review Form: <http://saturn.aerohive.com/view.php?fDocumentId=2815>

# Test Objectives

2.1 Test function which is capwap setup a HTTP connection with HM when enable capwap over http

2.2 Test function for name and port of proxy server

2.3 Test function for account of proxy server

2.4 Test function if capwap can not get HM IP, HTTP connection will be failed

2.5 Test setup HTTP failed status if configure a wrong HM IP

2.6 Test download/upload file successfully via jerry’s tool

2.7 Test capwap can get HM ip by dhcp

2.8 Test capwap switch betwwen UDP and HTTP

2.9 Test restartability for capwap over HTTP

2.9 Test regression UDP case for capwap over TCP

# Test Acceptance Criterion from Development

* Approved – Functional Specifications
* Approved – Unit Test Plans

# Product Pass Criterion

# Test Bed/Topo Design

* Top1

HiveAP connect with HM without proxy



* Top2

HiveAP connect with HM over proxy and proxy is gateway



* Top3

HiveAP connect with HM over proxy and proxy is not gateway



# TestCase

## Key Scenarios

## Function Test Case

### Check capwap over HTTP basic funcation

#### Capwapoverhttp\_Function\_1

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_1 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology1(not use proxy) | | |
| Description | Test function which is capwap setup a HTTP connection with HM when enable capwap over http | | |
| Pre-condition | Reset config  HM1 can be reachable  AP configure:  Capwap client server name HM’s IP  Capwap client server port 80 (CLI bug 11504)  Capwap client transport HTTP | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And sent a delta configure succefully | | |
| Test result | 1. Save image successful 2. Upload delta config successful 3. Upload complete config successful(bug 11503) 4. Upload bootstrap config successful 5. SSH client successful 6. Packet capture successful 7. Get tech data successful | | |
| Comment | Not use proxy | | |

#### Capwapoverhttp\_Function\_2

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_2 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Topology1(not use proxy) | | |
| Description | Test function which is capwap setup a HTTP connection with HM when configure wrong port | | |
| Pre-condition | Reset config  HM1 can be reachable  AP configure:  Capwap client server name HM’s IP  Capwap client server port 8080  Capwap client transfer-mode HTTP | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can not connect with HM. And AP haven’t abnormal error | | |
| Test result | AP could not connect to HM | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_3

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_3 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology2 (Use transparent proxy to test(default gateway is proxy’s ip address) | | |
| Description | Test function which is capwap setup a HTTP connection with HM when enable capwap over http | | |
| Pre-condition | Reset config  HM1 can be reachable  Proxy server use squid but can not configure proxy server account athentication  AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 1. Save image ------bug 11537 2. Upload delta config successful 3. Upload complete config------bug 11537 4. Upload bootstrap config------bug 11537 5. SSH client-----------------------not support 6. Packet capture failed 7. Get tech data failed | | |
| Comment | Use transparent proxy to test(default gateway is proxy’s ip address) | | |

#### Capwapoverhttp\_Function\_4

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_4 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology2 (Use transparent proxy to test(default gateway is proxy’s ip address) | | |
| Description | Test function for name and port of proxy server | | |
| Pre-condition | Reset config  HM1 can be reachable via proxy    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 1. Update image successful 2. Upload delta config successful 3. Upload complete config successful 4. Upload bootstrap successful 5. SSH client-------not support 6. Packet capture successful 7. Get tech data successful | | |
| Comment | 1. HM as log server not be supported 2. HM as NTP server not be supported(bug 11509) | | |

#### Capwapoverhttp\_Function\_5

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_5 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology3(use proxy but not transparent proxy) | | |
| Description | Test function for name and port of proxy server | | |
| Pre-condition | Reset config  HM1 can be reachable via proxy    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 1. Save image successful  2. Upload delta config successful(bug 11517)  3. Upload complete config successful(bug 11503 fixed, verified)  4. Upload bootstrap config  5. SSH client ------------------------not support  6. Packet capture successful  7. Get tech data successful | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_6

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_6 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test function for name and port of proxy server(custom port) | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server custom port(change 3128 to 3500) | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 1. Update image successful 2. Upload delta config successful 3. Upload complete config successful 4. Upload bootstrap successful 5. SSH client------not support 6. Packet capture successful 7. Get tech data successful | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_7

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_7 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test function for account of proxy server | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 1. Update image successful 2. Upload delta config successful 3. Upload complete config successful 4. Upload bootstrap successful 5. SSH client successful 6. Packet capture successful 7. Get tech data successful   2011-03-19 22:46:37 debug [capwap\_htc\_basic]: HTTP proxy content length:1048576  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: creating a new tunnel  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: Generate the authentication :guest:aerohive  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: Ready connect with remote HTTP tunnel server  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: register proxy port :3128  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: tunnel\_out\_connect: ready setup the out connection  2011-03-19 22:46:37 debug [capwap\_htc\_basic]: CAPWAP\_TCP: setup a TCP connection successfully  2011-03-19 22:46:38 debug [capwap\_htc\_basic]: Get response from http tunnel server  2011-03-19 22:46:38 debug [capwap\_htc\_basic]: tunnel\_out\_connect: output connected  2011-03-19 22:46:38 debug [capwap\_htc\_basic]: tunnel\_write\_request: TUNNEL\_OPEN (length=1)  2011-03-19 22:46:38 debug [capwap\_htc\_basic]: CAPWAP\_TCP: setup a TCP connection successfully  2011-03-19 22:46:39 debug [capwap\_htc\_basic]: Get response from http tunnel server  2011-03-19 22:46:39 debug [capwap\_htc\_basic]: tunnel\_in\_connect: input connected  2011-03-19 22:46:39 debug [capwap\_htc\_basic]: Connect with remote HTTP tunnel server done | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_8

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_8 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test function for account of proxy server (length of user-name and password) | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string>  Minumal and maxmual length of user-name and password | | |
| Test procedure | Send a delta configure via HM | | |
| Expect result | Capwap client can connect with HM. And AP haven’t abnormal error | | |
| Test result | 2011-03-19 23:32:27 debug [capwap\_htc\_basic]: HTTP proxy content length:1048576  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: creating a new tunnel  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: Generate the authentication :testtesttesttesttesttesttesttest:testtesttesttesttesttesttesttest  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: Ready connect with remote HTTP tunnel server  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: register proxy port :3128  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: tunnel\_out\_connect: ready setup the out connection  2011-03-19 23:32:27 debug [capwap\_htc\_basic]: CAPWAP\_TCP: setup a TCP connection successfully  2011-03-19 23:32:28 debug [capwap\_htc\_basic]: Get response from http tunnel server  2011-03-19 23:32:28 debug [capwap\_htc\_basic]: tunnel\_out\_connect: output connected  2011-03-19 23:32:28 debug [capwap\_htc\_basic]: tunnel\_write\_request: TUNNEL\_OPEN (length=1)  2011-03-19 23:32:28 debug [capwap\_htc\_basic]: CAPWAP\_TCP: setup a TCP connection successfully  2011-03-19 23:32:29 debug [capwap\_htc\_basic]: Get response from http tunnel server  2011-03-19 23:32:29 debug [capwap\_htc\_basic]: tunnel\_in\_connect: input connected  2011-03-19 23:32:29 debug [capwap\_htc\_basic]: Connect with remote HTTP tunnel server done | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_9

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_9 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test function if capwap can not get HM IP, HTTP connection will be failed | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure |  | | |
| Expect result | Capwap client can not connect with HM. And AP haven’t abnormal error | | |
| Test result | AP could not connect to HM------bug 11546 | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_10

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_10 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test setup HTTP failed status if configure a wrong HM IP | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure | Configure a wrong HM IP | | |
| Expect result | Capwap client can not connect with HM. And AP haven’t abnormal error | | |
| Test result | AP could not connect to HM | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_11

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_11 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test download/upload file | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure | Use jerry’s tools | | |
| Expect result | Download/upload file successfully | | |
| Test result | 1. Upload complete config from HM to AP successful(download) 2. Upload bootstrap successful(download) 3. Get tech data from AP successful(upload) 4. Packet capture successful(upload) | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_12

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_12 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test capwap can get HM ip by dhcp | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure | AP can get HM IP by DHCP  Send a delta configure | | |
| Expect result | AP can connect HM successfully. Delta configure can be successfully | | |
| Test result | 1. Update image successful 2. Upload delta config successful 3. Upload complete config successful 4. Upload bootstrap successful 5. SSH client successful 6. Packet capture successful 7. Get tech data successful | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_13

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_13 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology1 | | |
| Description | Test capwap switch betwwen UDP and HTTP | | |
| Pre-condition | Reset config  HM1 can be reachable | | |
| Test procedure | Make a tools which can enable capwap over http and disable capwap over http, will be switch when capwap has run status | | |
| Expect result | AP can connect HM successfully. Delta configure can be successfully | | |
| Test result | Pass | | |
| Comment | AP和HM之间没有代理  Default AP use UDP to connect with HM, then deny UDP, “no capwap client enable/capwap client enable”, AP will use TCP connect to HM, then permit UDP, “no capwap client enable/capwap client enable”, AP will use UDP again | | |

#### Capwapoverhttp\_Function\_14

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_14 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Topology2 | | |
| Description | Test restartability for capwap over HTTP | | |
| Pre-condition | Reset config  HM1 can be reachable    AP configure:  Capwap client server name HM’s IP  Capwap client server port 80  Capwap client transfer-mode HTTP  Capwap clinet HTTP proxy name serverIP port server port  Capwap client HTTP proxy user <string> password <string> | | |
| Test procedure | Enter into shell and killall -9 capwap | | |
| Expect result | AP can recover and haven’t abnormal error | | |
| Test result | Pass | | |
| Comment |  | | |

#### Capwapoverhttp\_Function\_15

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | CapwapoverHTTP\_Function\_15 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Topology1 | | |
| Description | Set capwap transport to http and set server port to 80, make AP or BR connent to HM, when upload complete config or delta config, the command should not be removed | | |
| Pre-condition | Set capwap client server name  Set capwap transport to http  Set capwap server port to 80 | | |
| Test procedure | 1. Show capwap client to check capwap with run status and transport mode is “HTTP on TCP” 2. Select AP or BR upload delta config or complete config, check if these two commands will be removed | | |
| Expect result | 1. AP connect with HM and transport mode is “HTTP on TCP” 2. The command that transport and server port setting not be removed | | |
| Test result | Bug 18919 &18954 | | |
| Comment |  | | |

# Rgression test case

#### Case ID Ft\_Capwap\_discovery\_1

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_1 |
| Priority | Accept |
| Topology to use | Topology2 |
| Description | When HiveAPs are in newly discovered state on HM, what is the capwap state on HiveAPs. If capwap traffic is ok. |
| Pre-condition | HiveAPs and HM are in capwap communication. |
| Test procedure | 1. Check the HiveAPs capwap state. 2. Check capwap state on HM in newly discovered page. |
| Expect result | HiveAPs: run state.  HM: connected. |
| Test result | Pass |

#### Case ID Ft\_Capwap\_discovery\_2

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_2 |
| Priority | Accept |
| Topology to use | Topology2 |
| Description | When HiveAPs are in newly discovered state on HM, what is the capwap state on HiveAPs. If capwap traffic is cut off. |
| Pre-condition | After HM has been found, cut down capwap communication between HiveAPs and HM. |
| Test procedure | 1. Shut down network traffic. 2. Check the HiveAPs capwap state. 3. Check capwap state on HM in newly discovered page. |
| Expect result | HiveAPs: discovery  HM: disconnected |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_3

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_3 |
| Priority | Accept |
| Topology to use | Topology2 |
| Description | When HiveAPs are in managed state on HM, what is the capwap state on HiveAPs. If capwap traffic is ok. |
| Pre-condition | HiveAPs are accepted as managed APs.  HiveAPs and HM are in capwap communication. |
| Test procedure | 1. Check the HiveAPs capwap state. 2. Check capwap state on HM in newly discovered page. |
| Expect result | HiveAPs: run state.  HM: connected. |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_4

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_4 |
| Priority | Accept |
| Topology to use | Topology2 |
| Description | When HiveAPs are in managed state on HM, what is the capwap state on HiveAPs. If capwap traffic is cut off. |
| Pre-condition | HiveAPs are accepted as managed APs.  HiveAPs and HM are in capwap communication. |
| Test procedure | 1. Check the HiveAPs capwap state. 2. Check capwap state on HM in newly discovered page. |
| Expect result | HiveAPs: discovery  HM: disconnected |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_5

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_5 |
| Priority | Middle |
| Topology to use | Topology2 |
| Description | When HiveAPs are in banned state on HM, what is the capwap state on HiveAPs. If capwap traffic is ok. |
| Pre-condition | HiveAPs are banned as banned APs on HM.  HiveAPs and HM are in capwap communication. |
| Test procedure | 1. Check the HiveAPs capwap state. 2. Check capwap state on HM in newly discovered page.   3. Check whether the HiveAPs and HM still exchange echo packets. |
| Expect result | HiveAPs: run state.  HM: connected.  HiveAPs and HM still exchange echo packets. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_6

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_6 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | When HiveAPs are in banned state on HM, delete it, what is the capwap state on HiveAPs. If capwap traffic is ok. |
| Pre-condition | HiveAPs are banned as banned APs on HM.  HiveAPs and HM are in capwap communication. |
| Test procedure | 1. Delete APs in banned on HM. 2. Check the HiveAPs capwap state. 3. Check capwap state on HM in newly discovered page.   4. Check whether the HiveAPs and HM still exchange echo packets. |
| Expect result | HiveAPs: will timeout, and resend discovery to reach run state.  HM: ignore echo packet, when receive new discovery, enter newly discovery state. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_7

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_7 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Config hivemanager IP on hivemanager and then push the configuration to the device and then reboot the device to see what happens. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_8

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_8 |
| Priority | Accept |
| Topology to use | HM---SW---(ETH0)BR100 |
| Description | Capwap can works normal via eth0 |
| Pre-condition | Configure HM in DHCP server  Open \_debug capwap basic  Debug console |
| Test procedure | 1. Show interface eth0 dhcp client to check if can obtain HM from DHCP server via eth0 2. Show capwap client to check if BR can connect to HM via eth0 |
| Expect result | 1. Can obtain HM from DHCP server via eth0 2. BR can connect to HM via eth0 |
| Test result |  |

## HiveAPs in factory default, DHCP server enabled.

## Case ID Ft\_Capwap\_discovery\_9

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_9 |
| Priority | High |
| Topology to use | Topology1 |
| Description | In the same broadcast domain, HiveManager can be discovered no matter how DHCP server is configured or not. |
| Pre-condition | HiveAP in factory default (or no configurations for HiveManager (hivemanager <ip>) and DHCP custom options)  HiveAP and HiveManager are in the same broadcast domain |
| Test procedure | 1. Boot up all HiveAPs. 2. Check the capwap state on device. 3. Check the state of all HiveAPs on HiveManager. 4. Use sniffer to check the packets. |
| Expect result | HiveAPs and HiveManger will enter run state. |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_10

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_10 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when HiveAPs get the option 225. DHCP server is Cisco. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain.  DHCP Server has custom option for HiveManager using option 225 |
| Test procedure | 1. Using Cisco DHCP server to config HiveManager IP address. 2. Boot up all HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_11

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_11 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when HiveAPs get the option 225. DHCP server is Linux. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain.  DHCP Server has custom option for HiveManager using option 225 |
| Test procedure | 1. Using Linux DHCP server to config HiveManager IP address. 2. Boot up all HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_12

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_12 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when HiveAPs get the option 225. DHCP server is Netscreen. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain.  DHCP Server has custom option for HiveManager using option 225 |
| Test procedure | 1. Using Netscreen DHCP server to config HiveManager IP address. 2. Boot up all devices. 3. Check the capwap state on device. 4. Check the state of all devices on HiveManager. |
| Expect result | HiveManager can be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_13

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_13 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Set up two DNS entries, one like “hivemanager\_test”, passed to client via DHCP server, and “hivemanager” (the default) with different address to see which is used. By default, the name returned by DHCP server should be used instead of the default hivemanager. It also should verify that there is no CAPWAP broadcast discovery message. The discovery message should be unicast. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_14

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_14 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when DNS name is configured on DHCP server using Linux’s. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain.  DHCP Server has custom option for HiveManager using option 225 |
| Test procedure | 1. Using Linux DHCP server to config HiveManager name. 2. Config DNS server. 3. Boot up all HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_15

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_15 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when DNS name is configured on DHCP server using Netscreen’s. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain.  DHCP Server has custom option for HiveManager using option 225 |
| Test procedure | 1. Using Netscreen DHCP server to config HiveManager name. 2. Config DNS server. 3. Boot up all HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_16

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_16 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager can not be discovered when DHCP server does not config 225 option. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain. |
| Test procedure | 1. Using any DHCP server to config but does not use option 225. 2. Boot up all HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can not be discovered because of the option not 225. After HiveAPs can not discover HM in different subnet, they will broadcast. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_17

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_17 |
| Priority | High |
| Topology to use | Topology2 |
| Description | HiveManager can be discovered when DNS server has “hivemanager” entry. |
| Pre-condition | HiveAP in factory default settings.  HiveAP and HiveManager are in the different broadcast domain. |
| Test procedure | 1. Config DNS to add “hivemanager” entry. 2. Boot up all HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by unicast because of the default name “hivemanager” configured on DNS server. |
| Test result |  |

## HiveAPs in non-factory default, DHCP server disabled

## Case ID Ft\_Capwap\_discovery\_18

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_18 |
| Priority | High |
| Topology to use | Topology1 |
| Description | HiveManager can be discovered by broadcasting in the same broadcast domain. |
| Pre-condition | DHCP server disabled.  HiveManager in the same broadcast domain.  HiveAP doesn’t have “hivemanager <ip addr>” config. |
| Test procedure | 1. Boot up all HiveAPs. 2. Config IP on HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by broadcasting. |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_19

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_19 |
| Priority | High |
| Topology to use | Topology1 |
| Description | HiveManager can be discovered by unicasting, when the HiveManager IP is manually configured. |
| Pre-condition | DHCP server disabled.  HiveManager in the same broadcast domain.  HiveAP has “hivemanager <ip addr>” config. |
| Test procedure | 1. Config HiveAP IP on HiveAPs. 2. Config hivemanager IP on HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by unicasting. |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_20

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_20 |
| Priority | Low |
|  |  |
| Topology to use | Topology1 |
| Description | HiveManager can be discovered by unicasting, when the HiveManager name is manually configured. |
| Pre-condition | DHCP server disabled.  HiveManager in the same broadcast domain.  HiveAP has “hivemanager name” config. |
| Test procedure | 1. Config manually HM name entry on DNS server. 2. Config HiveAP IP on HiveAPs. 3. Config hivemanager name on HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by unicasting because of the manually configured name configured on DNS server. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_21

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_21 |
| Priority | Low |
| Topology to use | Topology1 |
| Description | HiveManager can not be discovered with wrong HM IP manually set, after using unicast failed, HiveAP will still unicast to find HM. |
| Pre-condition | DHCP server disabled.  HiveManager in the same broadcast domain.  HiveAP has “hivemanager <ip addr>” config but it’s wrong. |
| Test procedure | 1. Config HiveAP IP on HiveAPs. 2. Config hivemanager IP on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can not be discovered, because the IP of HM is wrong. Every time HiveAPs will use HM IP for new capwap process. |
| Test result | Pass |

## Case ID Ft\_Capwap\_discovery\_22

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_22 |
| Priority | Low |
| Topology to use | Topology1 |
| Description | HiveManager can still be discovered with wrong name, but name on DNS server look up failed. |
| Pre-condition | DHCP server disabled.  HiveManager in the same broadcast domain.  HiveAP has “hivemanager name” config but it’s wrong. |
| Test procedure | 1. Config HiveAP IP on HiveAPs. 2. Config hivemanager name on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered, but name on DNS server look up failed. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_23

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_23 |
| Priority | Middle |
| Topology to use | Topology4 |
| Description | Only one HiveManager can be discovered when there are two HiveMangers exist in the same broadcast domain. |
| Pre-condition | DHCP server disabled.  Two HiveManagers are in the same broadcast domain.  No HiveManager IP or name config on all hiveaps. |
| Test procedure | 1. Config HiveAP IP on HiveAPs. 2. Config hivemanager name on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | Only one HiveManager can be discovered by broadcast. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_24

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_24 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HiveManager in different broadcast domain can not be discovered, when there is no HiveManager config IP or name on HiveAPs. |
| Pre-condition | DHCP server disabled.  HiveManager in different broadcast domain  HiveAP doesn’t have “hivemanager <ip addr> or name” config |
| Test procedure | 1. Config HiveAP IP on HiveAPs and save config. 2. Boot up the HiveAPs. 3. Check the capwap state on device. 4. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can not be discovered by broadcast. Because they are in different broadcast domain. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_25

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_25 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | When there is HiveManager config IP on HiveAPs, HiveManager in different broadcast domain can be discovered, |
| Pre-condition | DHCP server disabled.  HiveManager in different broadcast domain  HiveAP has “hivemanager <ip addr>” config |
| Test procedure | 1. Config HiveAP IP on HiveAPs 2. Config HiveManager IP on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by unicast. Because HiveAPs have leaned the IP of the HiveManager. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_26

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_26 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | When there is HiveManager config name on HiveAPs, HiveManager in different broadcast domain can be discovered, |
| Pre-condition | DHCP server disabled.  HiveManager in different broadcast domain  HiveAP has “hivemanager name” config |
| Test procedure | 1. Config HiveAP IP on HiveAPs 2. Config HiveManager IP on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can be discovered by unicast. Because HiveAPs have leaned the IP of the HiveManager. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_27

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_27 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | When there is wrong HiveManager config IP on HiveAPs, HiveManager can not be discovered. |
| Pre-condition | DHCP server disabled.  HiveManager in different broadcast domain  HiveAP has the wrong “hivemanager IP” config |
| Test procedure | 1. Config HiveAP IP on HiveAPs 2. Config HiveManager IP on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can not be discovered. Because HiveAPs have leaned wrong IP of the HiveManager, it should go as unicast -> (after failed ) still unicast -> still failed(because they every time HiveAPs will use wrong IP to start new capwap process) |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_28

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_28 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | When there is wrong HiveManager config name on HiveAPs, HiveManager in different broadcast domain can not be discovered, |
| Pre-condition | DHCP server disabled.  HiveManager in different broadcast domain  HiveAP has the wrong “hivemanager name” config |
| Test procedure | 1. Config HiveAP IP on HiveAPs 2. Config HiveManager name on HiveAPs and save config. 3. Boot up the HiveAPs. 4. Check the capwap state on device. 5. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveManager can not be discovered. Because HiveAPs have leaned wrong name of the HiveManager, it should go as resolve the name -> (after failed ) broadcast -> still failed(because they are in different broadcast domain) |
| Test result |  |

## HiveAPs in non-factory default, DHCP server enabled

## Case ID Ft\_Capwap\_discovery\_29

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_29 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | HM can be discovered when manually configure custom option for HM on HiveAPs. |
| Pre-condition | DHCP server enabled with the right HM IP.  HiveManager in different broadcast domain |
| Test procedure | 1. Config HM IP for custom option 229 on DHCP server 2. Config HiveAP IP on HiveAPs 3. Config HiveManager IP with option 229 on HiveAPs and save config 4. Boot up the HiveAPs 5. Check the capwap state on device. 6. Check the state of all HiveAPs on HiveManager. |
| Expect result | HiveAPs get the HM IP from option 229, they will enter run state. |
| Test result | Pass |

## HiveManger IP affect order test case

## Case ID Ft\_Capwap\_discovery\_30

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_30 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Config HiveManager IP in CLI-wizard, after reboot, HiveManager should be discovered by HiveAPs. |
| Pre-condition | 1. Default factory setting on HiveAPs. 2. DHCP server disabled. |
| Test procedure | 1. Reset HiveAPs to default factory 2. Config HiveManager in CLI-wizard 3. After complete CLI-wizard, reboot HiveAPs 4. Check capwap state on HiveAPs 5. Check capwap state on HiveManager |
| Expect result | Capwap state on HiveAPs should be run.  Capwap state on HiveManager should be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_31

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_31 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Maually HiveManager IP should have higher priority than that option got from DHCP server. |
| Pre-condition | 1. Default factory setting on HiveAPs. 2. DHCP server enabled with option 225 HiveManager IP A configured. |
| Test procedure | 1. Boot up HiveAPs 2. Config HiveManager IP/name B in CLI 3. Check capwap state on HiveAPs 4. Check capwap state on HiveManager 5. Interface mgt0 dhcp client to get option from dhcp server again |
| Expect result | HiveAPs should use IP B which is manually set, to send discovery to find HiveManager.  Capwap state on HiveAPs should be run.  Capwap state on HiveManager should be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_32

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_32 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Maually DHCP custom option should have higher priority than default option 225 got from DHCP server. |
| Pre-condition | 1. Default factory setting on HiveAPs. 2. DHCP server enabled with option 225 HiveManager IP A configured. 3. DHCP server enabled with option 223 HiveManager IP B configured. |
| Test procedure | 1. Boot up HiveAPs 2. Config option 223 for HM 3. Interface mgt0 dhcp client to get option from dhcp server 4. Check capwap state on HiveAPs 5. Check capwap state on HiveManager |
| Expect result | HiveAPs should use IP B which is manually set, to send discovery to find HiveManager.  Capwap state on HiveAPs should be run.  Capwap state on HiveManager should be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_33

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_33 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | Maually DHCP custom option should have higher priority than default option 225 got from DHCP server. |
| Pre-condition | 1. Default factory setting on HiveAPs. 2. DHCP server enabled with option 225 HiveManager IP A configured. 3. DHCP server enabled with option 223 HiveManager IP B configured. |
| Test procedure | 1. Boot up HiveAPs 2. Config option 223 for HM 3. Interface mgt0 dhcp client to get option from dhcp server 4. Check capwap state on HiveAPs 5. Check capwap state on HiveManager |
| Expect result | HiveAPs should use IP B which is manually set, to send discovery to find HiveManager.  Capwap state on HiveAPs should be run.  Capwap state on HiveManager should be discovered. |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_34

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_34 |
| Priority | Low |
| Topology to use | Topology2 |
| Description | The default hivemanager via DNS should have higher priority than broadcast. |
| Pre-condition | 1. Default factory setting on HiveAPs. 2. DHCP server enabled with option 225 HiveManager name “hivemanager” is configured. |
| Test procedure | 1. Boot up HiveAPs 2. Check capwap state on HiveAPs 3. Check capwap state on HiveManager |
| Expect result | HiveAPs should use resolved IP of the name “hivemanager” which is manually set, to send discovery to find HiveManager.  Capwap state on HiveAPs should be run.  Capwap state on HiveManager should be discovered. |
| Test result |  |

## Different IP type testing on capwap discovery

### Case ID *Ft\_Capwap\_discovery\_35*

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_35 |
| Priority | Low |
| Description | Verify default setting of capwap (ip in [0，10][11，64]，[63，127]，[128] ) |
| Pre-condition | Devicp have ip and can broadcast to hivemanager |
| Test procedure | Reboot device and don’t set hivemanager ip from cli and dhcp. |
| Expect result | Device can talk with hivemanager by broadcast and can entry run state. |
| Test result | Ok |

### Case ID *Ft\_Capwap\_discovery\_36*

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_36 |
| Priority | Low |
| Description | Modify hivemanager ip by cli (ip in [0，10][11，64]，[63，127]，[128]) |
| Pre-condition | There has a new hivemanager with ip |
| Test procedure | Reset hivemanager ip by cli. |
| Expect result | Device can talk with hivemanager by broadcast and can entry run state. |
| Test result | ok |

## Negative test case, based on capwap client state machine test

## Case ID Ft\_Capwap\_discovery\_Negative\_1

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_1 |
| Priority | Low |
| Topology to use |  |
| Description | When capwap is running, change mgt0 IP. |
| Pre-condition |  |
| Test procedure |  |
| Expect result | Pass: after a while, AP connect to HM |
| Test result | After about 3-4 min, New IP can talk CAPWAP with HM, because APs have to wait for server timeout, does we should do some enhancement? |

## Case ID Ft\_Capwap\_discovery\_Negative\_2

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_2 |
| Priority | Low |
| Topology to use |  |
| Description | When capwap is running, change HM IP. (to different IP or the same IP) |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result | If we quickly switch from other IP-> HM IP. After about 3-4 min, New IP can talk CAPWAP with HM, because APs have to wait for server timeout, does we should do some enhancement? |

## Case ID Ft\_Capwap\_discovery\_Negative\_3

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_3 |
| Priority | Low |
| Topology to use |  |
| Description | When capwap is running, no capwap client on HiveAPs then restart capwap client. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result | If we quickly switch from no-> enable. After about 3-4 min, New IP can talk CAPWAP with HM, because APs have to wait for server timeout, does we should do some enhancement? |

## Case ID Ft\_Capwap\_discovery\_Negative\_4

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_4 |
| Priority | Low |
| Topology to use |  |
| Description | Enabling CAPWAP client where there is no server out there, see what happens |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result | If there is mgt0 IP, HiveAPs will use this IP unicast to find HM again and again, from discovery -> sulking -> discovery.  If there is no mgt0 IP, HiveAPs will broadcast again and again. |

## Case ID Ft\_Capwap\_discovery\_Negative\_5

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_5 |
| Priority | Low |
| Topology to use |  |
| Description | When the capwap is in each state, unplug the Ethernet cable to the hive manager in order to test the state machine. Then plug on the cable how it handle. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_6

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_6 |
| Priority | Low |
| Topology to use |  |
| Description | When the capwap is in run state, unplug the cable to the hive manager to see what happens. Then plug on the cable how it handle. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_7

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_7 |
| Priority | Low |
| Topology to use |  |
| Description | Put the client on the denied AP category in the hive manager and enable the capwap client see what happens |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_8

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_8 |
| Priority | Low |
| Topology to use |  |
| Description | Device lost hivemanager ip, how capwap handle. |
| Pre-condition |  |
| Test procedure |  |
| Expect result |  |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_9

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_9 |
| Priority | Low |
| Description | Show capwap parameter |
| Pre-condition | Default setting |
| Test procedure | *When capwap run in all state, run show capwap client cli command* |
| Expect result | *Device will display all kind of parameter of capwap* |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_10

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_10 |
| Priority | Low |
| Description | Configure all kind of capwap parameter |
| Pre-condition | Default setting |
| Test procedure | *Configure all kind of capwap parameter and show capwap client* |
| Expect result | Capwap will show right capwap parameter |
| Test result |  |

## Case ID Ft\_Capwap\_discovery\_Negative\_11

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_discovery\_Negative\_11 |
| Priority | Low |
| Description | Show all kind of state duration time |
| Pre-condition | Default setting |
| Test procedure | *Capwap start and show capwap client* |
| Expect result | All state duration time is right |
| Test result |  |

## HiveOS shall send HM the SSID/BSSID mapping when a SSID bind to BSSID

### Case ID Ft\_Capwap\_Trap\_1

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_Trap\_1 |
| Priority | High |
| Description | When SSID bind to BSSID, CAPWAP will send event message to HM |
| Pre-condition | Device have ip and can broadcast to hivemanager |
| Test procedure | 1. bind a ssid to access interface. Turn on debug   \_debug dcd basic  \_debug capwap trap  On AP debug:  AH-0e53c0#interface wifi0 ssid 1  2009-11-30 19:24:25 debug [dcd\_basic]: create ifname wifi0.2 success!  2009-11-30 19:24:25 debug [dcd\_basic]: ssid bind/unbind trap constructed, total\_len = 64  2009-11-30 19:24:25 debug [dcd\_basic]: dump of ssid\_bind\_unbind\_trap:  2009-11-30 19:24:25 debug [dcd\_basic]: Trap Type: 103  2009-11-30 19:24:25 debug [dcd\_basic]: Data length: 61  2009-11-30 19:24:25 debug [dcd\_basic]: Object name length: 7  2009-11-30 19:24:25 debug [dcd\_basic]: Object name: wifi0.2  2009-11-30 19:24:25 debug [dcd\_basic]: Describe length: 39  2009-11-30 19:24:25 debug [dcd\_basic]: Describe: ssid (1) bind to bssid (0019:770e:53d1)  2009-11-30 19:24:25 debug [dcd\_basic]: Ifindex: 14  2009-11-30 19:24:25 debug [dcd\_basic]: Bssid: 0019:770e:53d1  2009-11-30 19:24:25 debug [dcd\_basic]: Ssid name length: 1  2009-11-30 19:24:25 debug [dcd\_basic]: Ssid name: 1  2009-11-30 19:24:25 debug [dcd\_basic]: State: bind  2009-11-30 19:24:25 debug [dcd\_basic]: ssid bind/unbind trap sent success  2009-11-30 19:24:25 debug [dcd\_basic]: set essid to interface wifi0.2 success!  2009-11-30 19:24:25 debug [dcd\_basic]: wifi0.2: Set device ssid (1) to fe success  2009-11-30 19:24:25 debug [dcd\_basic]: ssid profile is null  2009-11-30 19:24:25 debug [dcd\_basic]: reference count = 2  2009-11-30 19:24:25 debug [dcd\_basic]: DHCP server already disabled on ssid 1 interface wifi0.2  2009-11-30 19:24:25 debug [dcd\_basic]: Set 0 walled gardens to FE on ifname wifi0.2  2009-11-30 19:24:25 debug [dcd\_basic]: Send event (dcd respond auth) ssid\_name=1; seq\_num=0; more=0; default\_user\_group=0; notice\_flag=3;  2009-11-30 19:24:25 debug [dcd\_basic]: ifindex=14; ifname=wifi0.2; if\_state=0; if\_mode=3; strict=0; extend\_flag=3; additiona\_method\_flag=0  2009-11-30 19:24:25 debug [dcd\_basic]: suite\_name=1; method=16; rekey\_period=0; gmk\_rekey\_period= 0; default\_wep\_key\_idx=0; ptk-rekey-period=0;  2009-11-30 19:24:25 debug [dcd\_basic]: ptk\_timeout=0; ptk\_retry=0; gtk\_timeout=0; gtk\_retry=0  2009-11-30 19:24:25 debug [dcd\_basic]: key\_type[0]=0; key\_vlaue[0]=;  2009-11-30 19:24:25 debug [dcd\_basic]: key\_type[1]=0; key\_vlaue[1]=;  2009-11-30 19:24:25 debug [dcd\_basic]: key\_type[2]=0; key\_vlaue[2]=;  2009-11-30 19:24:25 debug [dcd\_basic]: key\_type[3]=0; key\_vlaue[3]=;  2009-11-30 19:24:25 debug [dcd\_basic]: eap\_reauth\_period=0 preauth\_flag=0 preauth\_ifname=; CWP\_registered\_uid = 0; CWP\_timeout = 43200; CWP\_popup=disabled  2009-11-30 19:24:25 debug [dcd\_basic]: aaa: num of server=0, auth retry interval=600, acct retryinterval=20  2009-11-30 19:24:25 debug [dcd\_basic]: No.=0 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:24:25 debug [dcd\_basic]: No.=1 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:24:25 debug [dcd\_basic]: No.=2 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:24:25 debug [dcd\_basic]: No.=3 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:24:25 debug [dcd\_basic]: DCD Send RADIUS DOS event: if\_name(wifi0.2), screen\_method(radius-attack),enable(0), threshold(5), action(alarm), action\_time(10)  2009-11-30 19:24:25 debug [capwap\_trap]: Get trp information, total len:64, trap type:5  2009-11-30 19:24:25 debug [capwap\_trap]: CAPWAP receive other type trap!  2009-11-30 19:24:25 debug [capwap\_trap]: Send capwap trap sequence number:110, total len:72  2009-11-30 19:24:25 debug [capwap\_trap]: print capwap send trap buffer:  2009-11-30 19:24:25 debug 7e7ff060 00 00 00 00 00 00 00 6e 67 00 3d 07 77 69 66 69 .......n g.=.wifi  2009-11-30 19:24:25 debug 7e7ff070 30 2e 32 27 73 73 69 64 20 28 31 29 20 62 69 6e 0.2'ssid (1) bin  2009-11-30 19:24:25 debug 7e7ff080 64 20 74 6f 20 62 73 73 69 64 20 28 30 30 31 39 d to bss id (0019  2009-11-30 19:24:25 debug 7e7ff090 3a 37 37 30 65 3a 35 33 64 31 29 00 00 00 0e 00 :770e:53 d1).....  2009-11-30 19:24:25 debug 7e7ff0a0 19 77 0e 53 d1 01 31 01 .w.S..1.  2009-11-30 19:24:25 debug [capwap\_trap]: Get trp information, total len:364, trap type:1  2009-11-30 19:24:25 debug [capwap\_trap]: CAPWAP receive state change trap!  2009-11-30 19:24:25 debug [capwap\_trap]: trap type is:3  2009-11-30 19:24:25 debug [capwap\_trap]: trap object name:wifi0, length:5  2009-11-30 19:24:25 debug [capwap\_trapAH-0e53c0#  2>check the trap on HiveManager  On HM, refresh current AP:    On HM data base:  su postgres  psql hm  select \* from HM\_LATESTXIF order by id desc;  id | apmac | apname | apserialnumber | bssid | ifadminstatus | ifconfmode | ifindex | ifmode | ifname | ifoperstatus | ifpromiscuous | iftype | ssidname | time | time\_zone | owner  -------+--------------+-----------+----------------+--------------+---------------+------------+---------+--------+---------+--------------+---------------+--------+----------------+---------------+-----------+-------  26995 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D1 | 1 | 0 | 13 | 1 | wifi0.2 | 1 | 2 | 1 | 1 | 1259608658836 | GMT | 6  26994 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E0 | 1 | 0 | 10 | 1 | wifi1.1 | 1 | 2 | 1 | yfeng\_location | 1259608658836 | GMT | 6  26993 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D0 | 1 | 0 | 8 | 1 | wifi0.1 | 1 | 2 | 1 | yfeng\_location | 1259608658836 | GMT | 6  26992 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53C0 | 1 | 0 | 7 | 0 | mgt0 | 1 | 2 | 0 | N/A | 1259608658836 | GMT | 6  26991 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E0 | 1 | 1 | 5 | 1 | wifi1 | 1 | 2 | 0 | N/A | 1259608658836 | GMT | 6  26990 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D0 | 1 | 1 | 3 | 1 | wifi0 | 1 | 2 | 0 | N/A |
| Expect result | 1>\_debug capwap trap or \_debug dcd basic to check the trap has been sent by capwap  2>enter into “Monitor”-> “Events” to see HM get a event which ssid bind to BSSID   |  |  |  |  |  | | --- | --- | --- | --- | --- | | AH-0e53c0 | [0019770E53C0](javascript:%20eventDetails('/hm/events.action',%20759)) | 11-30-2009 11:24:25 | ssid (1) bind to bssid (0019:770e:53d1) | wifi0.2 | |
| Test result | PASS  Note: now, HM can not sync the AP BSSD’s mac real time, will sync the value after one hour.(bug 10698) |

### Case ID Ft\_Capwap\_Trap\_2

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_Trap\_2 |
| Priority | High |
| Description | When SSID unbind to BSSID, CAPWAP will send event message to HM |
| Pre-condition | Device have ip and can broadcast to hivemanager |
| Test procedure | 3>unbind a ssid to access interface. Turn on debug  4>check the unbind trap on HiveManager |
| Expect result | 3>\_ debug capwap trap or \_debug dcd basic to check the trap has been sent by capwap  AP debug:  AH-0e53c0# no interface wifi0 ssid 1  2009-11-30 19:23:07 debug [dcd\_basic]: DHCP server already disabled on ssid 1 interface wifi0.2  2009-11-30 19:23:07 debug [dcd\_basic]: Set 0 walled gardens to FE on ifname wifi0.2  2009-11-30 19:23:07 debug [dcd\_basic]: Send event (dcd respond auth) ssid\_name=1; seq\_num=0; more=0; default\_user\_group=0; notice\_flag=4;  2009-11-30 19:23:07 debug [dcd\_basic]: ifindex=13; ifname=wifi0.2; if\_state=0; if\_mode=3; strict=0; extend\_flag=3; additiona\_method\_flag=0  2009-11-30 19:23:07 debug [dcd\_basic]: suite\_name=1; method=16; rekey\_period=0; gmk\_rekey\_period= 0; default\_wep\_key\_idx=0; ptk-rekey-period=0;  2009-11-30 19:23:07 debug [dcd\_basic]: ptk\_timeout=0; ptk\_retry=0; gtk\_timeout=0; gtk\_retry=0  2009-11-30 19:23:07 debug [dcd\_basic]: key\_type[0]=0; key\_vlaue[0]=;  2009-11-30 19:23:07 debug [dcd\_basic]: key\_type[1]=0; key\_vlaue[1]=;  2009-11-30 19:23:07 debug [dcd\_basic]: key\_type[2]=0; key\_vlaue[2]=;  2009-11-30 19:23:07 debug [dcd\_basic]: key\_type[3]=0; key\_vlaue[3]=;  2009-11-30 19:23:07 debug [dcd\_basic]: eap\_reauth\_period=0 preauth\_flag=0 preauth\_ifname=; CWP\_registered\_uid = 0; CWP\_timeout = 43200; CWP\_popup=disabled  2009-11-30 19:23:07 debug [dcd\_basic]: aaa: num of server=0, auth retry interval=600, acct retryinterval=20  2009-11-30 19:23:07 debug [dcd\_basic]: No.=0 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:23:07 debug [dcd\_basic]: No.=1 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:23:07 debug [dcd\_basic]: No.=2 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:23:07 debug [dcd\_basic]: No.=3 server=; auth port=1812; acct port=0; sec str=  2009-11-30 19:23:07 debug [dcd\_basic]: DCD Send RADIUS DOS event: if\_name(wifi0.2), screen\_method(radius-attack),enable(0), threshold(5), action(alarm), action\_time(10)  2009-11-30 19:23:07 debug [dcd\_basic]: ssid bind/unbind trap constructed, total\_len = 68  2009-11-30 19:23:07 debug [dcd\_basic]: dump of ssid\_bind\_unbind\_trap:  2009-11-30 19:23:07 debug [dcd\_basic]: Trap Type: 103  2009-11-30 19:23:07 debug [dcd\_basic]: Data length: 65  2009-11-30 19:23:07 debug [dcd\_basic]: Object name length: 7  2009-11-30 19:23:07 debug [dcd\_basic]: Object name: wifi0.2  2009-11-30 19:23:07 debug [dcd\_basic]: Describe length: 43  2009-11-30 19:23:07 debug [dcd\_basic]: Describe: ssid (1) unbind from bssid (0019:770e:53d1)  2009-11-30 19:23:07 debug [dcd\_basic]: Ifindex: 13  2009-11-30 19:23:07 debug [dcd\_basic]: Bssid: 0019:770e:53d1  2009-11-30 19:23:07 debug [dcd\_basic]: Ssid name length: 1  2009-11-30 19:23:07 debug [dcd\_basic]: Ssid name: 1  2009-11-30 19:23:07 debug [dcd\_basic]: State: unbind  2009-11-30 19:23:07 debug [dcd\_basic]: ssid bind/unbind trap sent success  2009-11-30 19:23:07 debug [dcd\_basic]: ssid profile is null  2009-11-30 19:23:07 debug [dcd\_basic]: hive name: yfeng\_location\_hive  2009-11-30 19:23:07 debug [dcd\_basic]: not backhaul interface  2009-11-30 19:23:07 debug [dcd\_basic]: reference count = 1  2009-11-30 19:23:08 debug [capwap\_trap]: Get trp information, total len:68, trap type:5  2009-11-30 19:23:08 debug [capwap\_trap]: CAPWAP receive other type trap!  2009-11-30 19:23:08 debug [capwap\_trap]: Send capwap trap sequence number:108, total len:76  2009-11-30 19:23:08 debug [capwap\_trap]: print capwap send trap buffer:  2009-11-30 19:23:08 debug 7e7ff060 00 00 00 00 00 00 00 6c 67 00 41 07 77 69 66 69 .......l g.A.wifi  2009-11-30 19:23:08 debug 7e7ff070 30 2e 32 2b 73 73 69 64 20 28 31 29 20 75 6e 62 0.2+ssid (1) unb  2009-11-30 19:23:08 debug 7e7ff080 69 6e 64 20 66 72 6f 6d 20 62 73 73 69 64 20 28 ind from bssid (  2009-11-30 19:23:08 debug 7e7ff090 30 30 31 39 3a 37 37 30 65 3a 35 33 64 31 29 00 0019:770 e:53d1).  2009-11-30 19:23:08 debug 7e7ff0a0 00 00 0d 00 19 77 0e 53 d1 01 31 00 .....w.S ..1.  2009-11-30 19:23:08 debug [capwap\_trap]: Get trp information, total len:364, trap type:1  2009-11-30 19:23:08 debug [capwap\_trap]: CAPWAP receive state change trap!  2009-11-30 19:23:08 debug [capwap\_trap]: trap type is:3  2009-11-30 19:23:08 debug [capwap\_trap]: trap object name:wifi0.2, length:7  2009-11-30 19:23:08 debug [capwap\_trap]: trap description:access interface wifi0.2 is down now, length:36  2009-11-30 19:23:08 debug [capwap\_trap]: trap code:0  2009-11-30 19:23:08 debug [capwap\_trap]: fill trap header len:52  2009-11-30 19:23:08 debug [capwap\_trap]: trap privious state:1  2009-11-30 19:23:08 debug [capwap\_trap]: trap current state:2  2009-11-30 19:23:08 debug [capwap\_trap]: total trap len:54  2009-11-30 19:23:08 debug [capwap\_trap]: Send capwap trap sequence number:109, total len:62  2009-11-30 19:23:08 debug [capwap\_trap]: printf capwap send trap buffer:  2009-11-30 19:23:08 debug 7e7ff058 00 00 00 00 00 00 00 6d 03 00 33 07 77 69 66 69 .......m ..3.wifi  2009-11-30 19:23:08 debug 7e7ff068 30 2e 32 24 61 63 63 65 73 73 20 69 6e 74 65 72 0.2$acce ss inter  2009-11-30 19:23:08 debug 7e7ff078 66 61 63 65 20 77 69 66 69 30 2e 32 20 69 73 20 face wif i0.2 is  2009-11-30 1AH-0e53c0#2009-11-30 19:23:17 debug [dcd\_basic]: Receive event (statistics send event to dcd).  2009-11-30 19:23:17 debug [dcd\_basic]: Recieve event (capwap query interface map info request).  4> enter into “Monitor”-> “Events” to see HM get a event which ssid unbind to BSSID   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | AH-0e53c0 | [0019770E53C0](javascript:%20eventDetails('/hm/events.action',%20757)) | 11-30-2009 11:23:08 | ssid (1) unbind from bssid (0019:770e:53d1) | wifi0.2 |   2>check the trap on HiveManager  On HM, refresh current AP:    On HM data base:  su postgres  psql hm  select \* from HM\_LATESTXIF order by id desc;  hm=#      select \* from HM\_LATESTXIF order by id desc;  id | apmac | apname | apserialnumber | bssid | ifadminstatus | ifconfmode | ifindex | ifmode | ifname | ifoperstatus | ifpromiscuous | iftype | ssidname | time | time\_zone | owner  -------+--------------+-----------+----------------+--------------+---------------+------------+---------+--------+---------+--------------+---------------+--------+----------------+---------------+-----------+-------  26914 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BE0 | 1 | 0 | 13 | 1 | wifi1.1 | 1 | 2 | 1 | yfeng\_location | 1259607600559 | GMT | 6  26913 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BD0 | 1 | 0 | 11 | 1 | wifi0.1 | 1 | 2 | 1 | yfeng\_location | 1259607600559 | GMT | 6  26912 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BC3 | 1 | 0 | 10 | 2 | agg0 | 2 | 2 | 0 | N/A | 1259607600559 | GMT | 6  26911 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BC2 | 1 | 0 | 9 | 2 | red0 | 2 | 2 | 0 | N/A | 1259607600559 | GMT | 6  26910 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BC0 | 1 | 0 | 8 | 0 | mgt0 | 1 | 2 | 0 | N/A | 1259607600559 | GMT | 6  26909 | 001977031BC0 | AH-031bc0 | 34008102200058 | 001977031BE0 | 1 | 1 | 6 | 1 | wifi1 | 1 | 2 | 0 | N/A |
| Test result | PASS |

### Case ID Ft\_Capwap\_Trap\_3

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_Trap\_3 |
| Priority | High |
| Description | When 16 SSID bind to BSSID, CAPWAP will send 16 event message to HM |
| Pre-condition | Device have ip and can broadcast to hivemanager |
| Test procedure | 1>bind 16 ssid to access interface. Turn on debug  2>check the trap on HiveManager |
| Expect result | 1>\_debug capwap trap or \_debug dcd basic to check the trap has been sent by capwap  2>enter into “Monitor”-> “Events” to see HM get a event which 16 ssid bind to BSSID  select \* from HM\_LATESTXIF order by id desc;  id | apmac | apname | apserialnumber | bssid | ifadminstatus | ifconfmode | ifindex | ifmode | ifname | ifoperstatus | ifpromiscuous | iftype | ssidname | time | time\_zone | owner  -------+--------------+-----------+----------------+--------------+---------------+------------+---------+--------+---------+--------------+---------------+--------+----------------+---------------+-----------+-------  28502 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E7 | 1 | 0 | 28 | 1 | wifi1.8 | 1 | 2 | 1 | 1 | 1259613450893 | GMT | 6  28501 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E6 | 1 | 0 | 27 | 1 | wifi1.7 | 1 | 2 | 1 | 5 | 1259613450893 | GMT | 6  28500 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E5 | 1 | 0 | 26 | 1 | wifi1.6 | 1 | 2 | 1 | 6 | 1259613450893 | GMT | 6  28499 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E4 | 1 | 0 | 25 | 1 | wifi1.5 | 1 | 2 | 1 | 4 | 1259613450893 | GMT | 6  28498 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E3 | 1 | 0 | 24 | 1 | wifi1.4 | 1 | 2 | 1 | 2 | 1259613450893 | GMT | 6  28497 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E2 | 1 | 0 | 23 | 1 | wifi1.3 | 1 | 2 | 1 | 3 | 1259613450893 | GMT | 6  28496 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D7 | 1 | 0 | 22 | 1 | wifi0.8 | 1 | 2 | 1 | 1 | 1259613450893 | GMT | 6  28495 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D6 | 1 | 0 | 21 | 1 | wifi0.7 | 1 | 2 | 1 | 5 | 1259613450893 | GMT | 6  28494 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D5 | 1 | 0 | 20 | 1 | wifi0.6 | 1 | 2 | 1 | 6 | 1259613450893 | GMT | 6  28493 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D4 | 1 | 0 | 19 | 1 | wifi0.5 | 1 | 2 | 1 | 4 | 1259613450893 | GMT | 6  28492 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D3 | 1 | 0 | 18 | 1 | wifi0.4 | 1 | 2 | 1 | 2 | 1259613450893 | GMT | 6  28491 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D2 | 1 | 0 | 17 | 1 | wifi0.3 | 1 | 2 | 1 | 3 | 1259613450893 | GMT | 6  28490 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E1 | 1 | 0 | 16 | 1 | wifi1.2 | 1 | 2 | 1 | ssid0 | 1259613450893 | GMT | 6  28489 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D1 | 1 | 0 | 15 | 1 | wifi0.2 | 1 | 2 | 1 | ssid0 | 1259613450893 | GMT | 6  28488 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E0 | 1 | 0 | 10 | 1 | wifi1.1 | 1 | 2 | 1 | yfeng\_location | 1259613450893 | GMT | 6  28487 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D0 | 1 | 0 | 8 | 1 | wifi0.1 | 1 | 2 | 1 | yfeng\_locati |
| Test result | PASS |

### Case ID Ft\_Capwap\_Trap\_4

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_Trap\_4 |
| Priority | High |
| Description | When 16 SSID unbind to BSSID, CAPWAP will send 16 event message to HM |
| Pre-condition | Device have ip and can broadcast to hivemanager |
| Test procedure | 3>unbind 16 ssid to access interface via delta config. Turn on debug  4>Check the unbind trap on HiveManager |
| Expect result | 3>\_ debug capwap trap or \_debug dcd basic to check the trap has been sent by capwap  4> Check the unbind trap on HiveManager |
| Test result | PASS |

### Case ID Ft\_Capwap\_Trap\_5

|  |  |
| --- | --- |
| Case ID | Ft\_Capwap\_Trap\_5 |
| Priority | High |
| Description | When HM query statistic vif table, CAPWAP will send ssid’s BSSID MAC to HM |
| Pre-condition | Device have ip and can broadcast to hivemanager |
| Test procedure | 1>bind a ssid to access interface  2> query statistic vif table on HiveManager, check the trap on HiveManager |
| Expect result | 1>on vif talbe, can see correct BSSID MAC on HiveManager   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Radio Details | | | | | | | | Radio | Type | Mode | Channel | Tx Power | Noise Floor | SSID | | wifi0 | 11ng | Access | 1 | 10 dBm | -106 dBm | [yfeng\_location](javaScript:%20showSsidDetails('yfeng_location',%206);),  [ssid0](javaScript:%20showSsidDetails('ssid0',%206);), | | wifi1 | 11na | Access | 36 | 10 dBm | -120 dBm | [yfeng\_location](javaScript:%20showSsidDetails('yfeng_location',%206);),  [ssid0](javaScript:%20showSsidDetails('ssid0',%206);), |   2>on HM data base:  hm=#        select \* from HM\_LATESTXIF order by id desc;  id | apmac | apname | apserialnumber | bssid | ifadminstatus | ifconfmode | ifindex | ifmode | ifname | ifoperstatus | ifpromiscuous | iftype | ssidname | time | time\_zone | owner  -------+--------------+-----------+----------------+--------------+---------------+------------+---------+--------+---------+--------------+---------------+--------+----------------+---------------+-----------+-------  28281 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E1 | 1 | 0 | 16 | 1 | wifi1.2 | 1 | 2 | 1 | ssid0 | 1259613186528 | GMT | 6  28280 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D1 | 1 | 0 | 15 | 1 | wifi0.2 | 1 | 2 | 1 | ssid0 | 1259613186528 | GMT | 6  28279 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E0 | 1 | 0 | 10 | 1 | wifi1.1 | 1 | 2 | 1 | yfeng\_location | 1259613186528 | GMT | 6  28278 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D0 | 1 | 0 | 8 | 1 | wifi0.1 | 1 | 2 | 1 | yfeng\_location | 1259613186528 | GMT | 6  28277 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53C0 | 1 | 0 | 7 | 0 | mgt0 | 1 | 2 | 0 | N/A | 1259613186528 | GMT | 6  28276 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53E0 | 1 | 1 | 5 | 1 | wifi1 | 1 | 2 | 0 | N/A | 1259613186528 | GMT | 6  28275 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53D0 | 1 | 1 | 3 | 1 | wifi0 | 1 | 2 | 0 | N/A | 1259613186528 | GMT | 6  28274 | 0019770E53C0 | AH-0e53c0 | 12009060800031 | 0019770E53C0 | 1 | 0 | 2 | 2 | eth0 | 1 | 2 | 0 | N/A |
| Test result | PASS |

## Stress Test Case

## Duration Test Case

## Performance Test Case

## Scalability Test Case

## Compatibility Test Case

## CLI Management (Automation Status: Yes/No)

<Just list all cli that this feature has one by one>

< memory leak case for these CLI.Normally, the leak is happened when we do some commands repeatedly.  Like create an object, then delete that object, it should release all the memory it allocated. But this is not true for all the cases. If you create/delete an object several times(but how many times?) and the memory just going down and never recovered, it maybe a memory leak(again, how can we decide it is really a memory leak?).>

## GUI Management-HiveManager

<List HM test case or test log>

## GUI Management-HiveUI

<List HiveUI test case or test log>