Internet Engineering Task Force (IETF)
Request for Comments: 5833
Category: Informational
ISSN: 2070-1721

Y. Shi, Ed.
Hangzhou H3C Tech. Co., Ltd.
D. Perkins, Ed.
C. Elliott, Ed.

Y. Zhang, Ed.
Fortinet, Inc.

May 2010

Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Base MIB

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular, it describes the managed objects for modeling the Control And Provisioning of Wireless Access Points (CAPWAP) Protocol. This MIB module is presented as a basis for future work on the SNMP management of the CAPWAP protocol.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5833.

Table of Contents

1. Introduction

The CAPWAP Protocol [RFC5415] defines a standard, interoperable protocol, which enables an Access Controller (AC) to manage a collection of Wireless Termination Points (WTPs).

This document defines a MIB module that can be used to manage the CAPWAP implementations. This MIB module covers both configuration and WTP status-monitoring aspects of CAPWAP, and provides a way to reuse MIB modules for any wireless technology. It presented as a basis for future work on a SNMP management of the CAPWAP protocol.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579], and STD 58, RFC 2580 [RFC2580].

3. Terminology

This document uses terminology from the CAPWAP Protocol specification [RFC5415] and the Architecture Taxonomy for CAPWAP [RFC4118].

Access Controller (AC): The network entity that provides WTP access to the network infrastructure in the data plane, control plane, management plane, or a combination therein.

Wireless Termination Point (WTP): The physical or network entity that contains an radio frequency (RF) antenna and wireless physical layer (PHY) to transmit and receive station traffic for wireless access networks.

Control And Provisioning of Wireless Access Points (CAPWAP): It is a generic protocol defining AC and WTP control and data plane communication via a CAPWAP protocol transport mechanism. CAPWAP control messages, and optionally CAPWAP data messages, are secured using Datagram Transport Layer Security (DTLS) [RFC4347].

CAPWAP Control Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC control port, WTP control port, and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP control packets are sent and received.

CAPWAP Data Channel: A bi-directional flow defined by the AC IP Address, WTP IP Address, AC data port, WTP data port, and the transport-layer protocol (UDP or UDP-Lite) over which CAPWAP data packets are sent and received.

Station (STA): A device that contains an interface to a wireless medium (WM).

Split and Local MAC: The CAPWAP protocol supports two modes of operation: Split and Local MAC (medium access control). In Split MAC mode, all Layer 2 wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between the AC and the WTPs. The Local MAC mode allows the data frames to be either locally bridged or tunneled as 802.3 frames.

Wireless Binding: The CAPWAP protocol is independent of a specific WTP radio technology, as well its associated wireless link-layer protocol. Elements of the CAPWAP protocol are designed to accommodate the specific needs of each wireless technology in a standard way. Implementation of the CAPWAP protocol for a particular wireless technology MUST define a binding protocol for it, e.g., the binding for IEEE 802.11, provided in [RFC5416].

Autonomous Wireless Local Area Network (WLAN) Architecture: It is the traditional autonomous WLAN architecture, in which each WTP is a single physical device that implements all the wireless services.

Centralized WLAN Architecture: It is an emerging hierarchical architecture utilizing one or more centralized controllers for managing a large number of WTP devices. It can be said that the full wireless functions are implemented across multiple physical network devices, namely, the WTPs and ACs.

4. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

5. Overview

5.1. Requirements and Constraints

The CAPWAP Protocol MIB module (CAPWAP-BASE-MIB) is designed to:

- Support centralized management and monitoring of WTPs from the AC in combination with the CAPWAP protocol;

- Allow operators to make configurations for WTPs before and after they connect to the AC;

- Support querying of CAPWAP protocol parameters;

- Support displaying of WTPs' current states and configurations;

- Provide basic property information about the AC, WTPs, radios, and stations, and their relationships;

- Provide counters for events on WTPs and radios such as reboot and hardware failure;

- Provide various notifications such as channel up and join failure.

5.2. Wireless Binding MIB Modules

Other Standards Development Organizations (SDOs), such as IEEE, have already defined MIB modules for a specific wireless technology, e.g., IEEE 802.11 MIB module [IEEE.802-11.2007]. Such MIB modules are called wireless binding MIB modules.

5.3. Design Objectives

This document introduces a mechanism to avoid redefining MIB objects in the existing MIB modules for a specific wireless technology, in other words, a mechanism to reuse wireless binding MIB modules defined by other SDOs.

In summary, the CAPWAP-BASE-MIB module has the following design objectives:

- To implement an architecture that uses SNMP for the management and control of wireless networks, and answering the operator's requirements for centralized management, whatever the wireless devices are configured and deployed (centralized, autonomous, or some mix);

- To be consistent with the CAPWAP protocol;

- To be independent of any wireless technologies and be able to reuse wireless binding MIB modules defined by other SDOs;

- To enable interoperability between vendors;

- To meet the management requirements for the centralized WLAN architecture.

5.4. Design Idea

The basic design idea of the CAPWAP-BASE-MIB module is:

- The SNMP agent MUST be run on the AC devices and is not REQUIRED on the WTP devices. It follows the same model as the CAPWAP protocol: Centralized Control.

- It is designed to accommodate the specific needs of each wireless technology in a standard way. It is independent of any wireless technologies.

- The ifIndex [RFC2863] is used as a common index for corresponding interfaces in the CAPWAP-BASE-MIB and the MIB modules of specific wireless technologies.

- The operator could manage and control the centralized WLAN architectures using multiple MIB modules defined by multiple SDOs, while keeping them loosely coupled.

5.5. Mechanism of Reusing Wireless Binding MIB Modules

For any wireless technology, the configuration and management of radios are very important. As usual, wireless binding MIB modules support radio management on their own. For example, the MIB tables such as the dot11OperationTable [IEEE.802-11.2007] are able to support WTP radio configuration. These tables use the ifIndex as the index, and work well under autonomous WLAN architecture.

To reuse such wireless binding MIB modules is very important to centralized WLAN architectures. According to [RFC5415], a specific PHY radio could be identified by the combination of the identifiers of the WTP and radio (WTP ID + Radio ID), so the key point is to make use of the ifIndex idea and find a way to maintain the mappings between 'WTP ID + radio ID' and the ifIndex. As a generic mechanism, an ifIndex can identify an interface in an abstract way, and it does NOT care for the interface's PHY location (either on the WTP or AC). The AC can have WTP Virtual Radio Interfaces to logically represent PHY radios on the WTP. From the operator's perspective, it appears that PHY radios are located on the AC, and the PHY location of the

WTP (radio) is hidden. The operator can operate radios through MIB tables with the ifIndex of a WTP Virtual Radio Interface. As a type of abstract interface, the WTP Virtual Radio Interface could be used by any wireless technology such as IEEE 802.11 and 802.16. The capwapBaseWirelessBindingTable in the CAPWAP-BASE-MIB module is used to store the mappings between the 'WTP ID + Radio ID' and the ifIndex.

5.6. CAPWAP Protocol Wireless Binding MIB Module

According to the CAPWAP Protocol specification [RFC5415], when defining a binding for wireless technologies, the authors MUST include any necessary definitions for technology-specific messages and all technology-specific message elements for those messages. A CAPWAP binding protocol is required for a specific wireless binding technology, e.g., the protocol of [RFC5416] for IEEE 802.11 binding.

Sometimes, not all the technology-specific message elements in a CAPWAP binding protocol have MIB objects defined by other SDOs. For example, the protocol of [RFC5416] defines WLAN management. The WLAN refers to a logical component instantiated on a WTP device. A single physical WTP MAY operate a number of WLANs. Also, Local or Split MAC modes could be specified for a WLAN. The MAC mode for a WLAN is not in the scope of IEEE 802.11 [IEEE.802-11.2007]. In such cases, in addition to the existing wireless binding MIB modules defined by other SDOs, a CAPWAP protocol wireless binding MIB module is required to be defined for a wireless binding, e.g, the CAPWAP Protocol Binding MIB for IEEE 802.11 [RFC5834].

5.7. WTP Profile

In a centralized WLAN architecture, a WTP profile is used to make configurations such as a static IP address for a WTP before and after it connects to the AC. It MUST contain the Base MAC address [RFC5415] of the WTP because the CAPWAP message received from the WTP contains the Base MAC address and the AC uses this Base MAC address to find the corresponding WTP profile.

Section 4.6.40 of [RFC5415] omits indicating that the WTP's Base MAC address MUST be included in the WTP Board Data message element. This is a known errata item [Err1832] and should be fixed in any future revision of RFC 5415.

Another important function of WTP profile is to trigger the creation of WTP Virtual Radio Interfaces on the AC. To implement this function, a WTP profile MUST include the WTP's model number [RFC5415], which reflects the number of PHY radios on the WTP. In this way, the creation of a WTP profile triggers the AC to

automatically create the same number of WTP Virtual Radio Interfaces corresponding to the WTP's PHY radios without manual intervention. With the ifIndexes of WTP Virtual Radio Interfaces, the operator could configure and manage the WTP's PHY radios through the wireless binding MIB modules.

6. Structure of the MIB Module

The MIB objects are derived from the CAPWAP protocol document [RFC5415].

1) capwapBaseAcNameListTable

   The AC name list table is used to configure the AC name list.

2) capwapBaseMacAclTable

   The ACL table is used to configure stations' Access Control Lists (ACLs).

3) capwapBaseWtpProfileTable

   The WTP profile table is used to configure WTP profiles for WTPs to be managed before they connect to the AC. An operator could change a WTP's current configuration by changing the values of parameters in the corresponding WTP profile, then the WTP could get the new configuration through the CAPWAP control channel.

4) capwapBaseWtpStateTable

   The state table of WTPs is used to indicate the AC's CAPWAP FSM state for each WTP, and helps the operator to query a WTP's current configuration.

5) capwapBaseWtpTable

   The WTP table is used to display properties of the WTPs in running state.

6) capwapBaseWirelessBindingTable

   The wireless binding table is used to display the mappings between WTP Virtual Radio Interfaces and PHY radios, and the wireless binding type for each PHY radio.

7) capwapBaseStationTable

The station table is used for providing stations' basic property information.

8) capwapBaseWtpEventsStatsTable

The WTP events statistic table is used for collecting WTP reboot count, link failure count, hardware failure count and so on.

9) capwapBaseRadioEventsStatsTable

The radio events statistic table is used for collecting radio reset count, channel change count, hardware failure count, and so on.

7. Relationship to Other MIB Modules

7.1. Relationship to SNMPv2-MIB Module

The CAPWAP-BASE-MIB module does not duplicate the objects of the 'system' group in the SNMPv2-MIB [RFC3418] that is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data.

7.2. Relationship to IF-MIB Module

The Interfaces Group [RFC2863] defines generic managed objects for managing interfaces. This memo contains the media-specific extensions to the Interfaces Group for managing WTP PHY radios that are modeled as interfaces.

The IF-MIB module is required to be supported on the AC. Each PHY radio on the WTP corresponds to a WTP Virtual Radio Interface on the AC. The WTP Virtual Radio Interface provides a way to configure the radio's parameters and query radio's traffic statistics, and reuse wireless binding modules defined by other SDOs. The interface MUST be modeled as an ifEntry, and ifEntry objects such as ifIndex, ifDescr, ifName, and ifAlias are to be used as per [RFC2863].

Also, as an ifIndex [RFC2863] is used as a common index for corresponding interfaces in the CAPWAP-BASE-MIB and specific wireless technologies MIB modules, the AC MUST have a mechanism that preserves the values of the ifIndexes in the ifTable at AC reboot.

7.3. Relationship to ENTITY-MIB Module

The ENTITY-MIB module [RFC4133] meets the need for a standardized way of representing a single agent that supports multiple instances of one MIB. It could express a certain relationship between multiple entities and provide entity properties for each entity.

In a centralized WLAN architecture, the SNMP agent runs on the AC and is not required on the WTP. With the ENTITY-MIB module on the AC, it could keep entity information such as firmware revision and software revision of the AC and WTPs. From the ENTITY-MIB module's perspective, the overall physical entity (AC) is a 'compound' of multiple physical entities (that is, the WTPs connected to AC), and all entities are each identified by a physical index. The capwapBaseWtpTable of the CAPWAP-BASE-MIB module uses the capwapBaseWtpPhyIndex object to store the mappings of WTP object between CAPWAP-BASE-MIB and ENTITY-MIB modules.

By querying both the CAPWAP-BASE-MIB and ENTITY-MIB modules, operators could query the status and properties of the AC and WTPs. For example, they could get a WTP's current status through the CAPWAP-BASE-MIB module, and a WTP's software revision information through the ENTITY-MIB module. The CAPWAP-BASE-MIB module does not duplicate those objects defined in the ENTITY-MIB module.

7.4. Relationship to Wireless Binding MIB Modules

The wireless binding MIB module of a wireless technology (such as [IEEE.802-11.2007]) is required to be supported on the AC. The CAPWAP-BASE-MIB module is able to support any wireless binding. Through the ifIndexes of WTP Virtual Radio Interfaces, it provides a consistent and abstract way of reusing MIB objects in the wireless binding MIB modules. The CAPWAP-BASE-MIB module does not duplicate those objects defined in the wireless binding MIB modules.

7.5. MIB Modules Required for IMPORTS

The following MIB module IMPORTS objects from SYSAPPL-MIB [RFC2287], SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863], SNMP-FRAMEWORK-MIB [RFC3411], INET-ADDRESS-MIB [RFC4001], and ENTITY-MIB [RFC4133].

8. Example of CAPWAP-BASE-MIB Module Usage

Below, the IEEE 802.11 binding is used as an example of how the MIB modules operate.

1) Create a WTP profile.

Suppose the WTP's Base MAC address is '00:01:01:01:01:00'. Create the WTP profile as follows:

```
In capwapBaseWtpProfileTable
{
  capwapBaseWtpProfileId                  = 1,
  capwapBaseWtpProfileName                = 'WTP Profile 123456',
  capwapBaseWtpProfileWtpMacAddress       = '00:01:01:01:01:00',
  capwapBaseWtpProfileWtpModelNumber            = 'WTP123',
  capwapBaseWtpProfileWtpName                   = 'WTP 123456',
  capwapBaseWtpProfileWtpLocation               = 'office',
  capwapBaseWtpProfileWtpStaticIpEnable         = true(1),
  capwapBaseWtpProfileWtpStaticIpType           = ipv4(1),
  capwapBaseWtpProfileWtpStaticIpAddress        = '192.0.2.10',
  capwapBaseWtpProfileWtpNetmask                = '255.255.255.0',
  capwapBaseWtpProfileWtpGateway                = '192.0.2.1',
  capwapBaseWtpProfileWtpFallbackEnable         = true(1),
  capwapBaseWtpProfileWtpEchoInterval           = 30,
  capwapBaseWtpProfileWtpIdleTimeout            = 300,
  capwapBaseWtpProfileWtpMaxDiscoveryInterval   = 20,
  capwapBaseWtpProfileWtpReportInterval         = 120,
  capwapBaseWtpProfileWtpStatisticsTimer        = 120,
  capwapBaseWtpProfileWtpEcnSupport             = limited(0)
}
```

Suppose the WTP with model number 'WTP123' has one PHY radio, which is identified by ID 1. The creation of this WTP profile triggers the AC to automatically create a WTP Virtual Radio Interface and add a new row object to the capwapBaseWirelessBindingTable without manual intervention. Suppose the ifIndex of the WTP Virtual Radio Interface is 10. The following information is stored in the capwapBaseWirelessBindingTable.

```
In capwapBaseWirelessBindingTable
{
  capwapBaseWtpProfileId                          = 1,
  capwapBaseWirelessBindingRadioId                = 1,
  capwapBaseWirelessBindingVirtualRadioIfIndex    = 10,
  capwapBaseWirelessBindingType                   = dot11(2)
}
```

The WTP Virtual Radio Interfaces on the AC correspond to the PHY radios on the WTP. The WTP Virtual Radio Interface is modeled by ifTable [RFC2863].

```
In ifTable
{
  ifIndex              = 10,
  ifDescr              = 'WTP Virtual Radio Interface',
  ifType               = 254,
  ifMtu                = 0,
  ifSpeed              = 0,
  ifPhysAddress        = '00:00:00:00:00:00',
  ifAdminStatus        = true(1),
  ifOperStatus         = false(0),
  ifLastChange         = 0,
  ifInOctets           = 0,
  ifInUcastPkts        = 0,
  ifInDiscards         = 0,
  ifInErrors           = 0,
  ifInUnknownProtos    = 0,
  ifOutOctets          = 0,
  ifOutUcastPkts       = 0,
  ifOutDiscards        = 0,
  ifOutErrors          = 0
 }
```

2) Query the ifIndexes of WTP Virtual Radio Interfaces.

Before configuring PHY radios, the operator needs to get the ifIndexes of WTP Virtual Radio Interfaces corresponding to the PHY radios.

As capwapBaseWirelessBindingTable already stores the mappings between PHY radios (Radio IDs) and the ifIndexes of WTP Virtual Radio Interfaces, the operator can get the ifIndex information by querying this table. Such a query operation SHOULD run from radio ID 1 to radio ID 31 according to [RFC5415]), and stop when an invalid ifIndex value (0) is returned.

This example uses capwapBaseWtpProfileId = 1 and capwapBaseWirelessBindingRadioId = 1 as inputs to query the capwapBaseWirelessBindingTable, and gets capwapBaseWirelessBindingVirtualRadioIfIndex = 10. Then it uses capwapBaseWtpProfileId = 1 and capwapBaseWirelessBindingRadioId = 2, and gets an invalid ifIndex value (0), so the query operation ends. This method gets not only the ifIndexes of WTP Virtual Radio Interfaces, but also the numbers of PHY radios. Besides checking whether the ifIndex value is valid, the operator SHOULD check whether the capwapBaseWirelessBindingType is the desired binding type.

3) Configure specific wireless binding parameters for a WTP Virtual Radio Interface.

This configuration is made on the AC through a specific wireless binding MIB module such as the IEEE 802.11 MIB module.

The following shows an example of configuring parameters for a WTP Virtual Radio Interface with ifIndex 10 through the IEEE 802.11 dot11OperationTable [IEEE.802-11.2007].

```
In dot11OperationTable
{
  ifIndex                                  = 10,
  dot11MACAddress                          = '00:00:00:00:00:00',
  dot11RTSThreshold                        = 2347,
  dot11ShortRetryLimit                     = 7,
  dot11LongRetryLimit                      = 4,
  dot11FragmentationThreshold              = 256,
  dot11MaxTransmitMSDULifetime             = 512,
  dot11MaxReceiveLifetime                  = 512,
  dot11ManufacturerID                      = 'capwap',
  dot11ProductID                           = 'capwap',
  dot11CAPLimit                            = 2,
  dot11HCCWmin                             = 0,
  dot11HCCWmax                             = 0,
  dot11HCCAIFSN                            = 1,
  dot11ADDBAResponseTimeout                = 1,
  dot11ADDTSResponseTimeout                = 1,
  dot11ChannelUtilizationBeaconInterval    = 50,
  dot11ScheduleTimeout                     = 10,
  dot11DLSResponseTimeout                  = 10,
  dot11QAPMissingAckRetryLimit             = 1,
  dot11EDCAAveragingPeriod                 = 5
}
```

4) Get the current configuration status report from the WTP to the AC.

According to [RFC5415], before a WTP that has joined the AC gets configuration from the AC, it needs to report its current configuration status by sending a configuration status request message to the AC, which uses the message to update MIB objects on the AC. For example, for IEEE 802.11 binding, the AC updates data in the ifTable [RFC2863] and IEEE 802.11 MIB module, and so on, according to the message. For ifIndex 10, its ifOperStatus in ifTable is updated according to the current radio operational status in the CAPWAP message.

5) Query WTP and radio statistical data.

After WTPs start to run, the operator could query WTP and radio statistical data through CAPWAP-BASE-MIB and the specific binding MIB module on the AC. For example, through dot11CountersTable in the IEEE 802.11 MIB module, the operator could query the counter data of a radio using the ifIndex of the corresponding WTP Virtual Radio Interface. With the capwapBaseWtpTable table in the CAPWAP-BASE-MIB module, the operator could query the properties of running WTPs.

6) Run MIB operations through a CAPWAP protocol wireless binding MIB module.

For example, for the CAPWAP IEEE 802.11 binding protocol [RFC5416], some MIB operations such as MAC mode configuration for a WLAN depend on the CAPWAP Protocol Binding MIB for IEEE 802.11 [RFC5834]. For more information, refer to [RFC5834].

7) Query other properties of a WTP.

The Operator could query MIB objects in the ENTITY-MIB [RFC4133] module by using the capwapBaseWtpPhyIndex in the capwapBaseWtpTable of CAPWAP-BASE-MIB module. The properties of a WTP such as software version, hardware version are available in the ENTITY-MIB module.

9. Definitions

```
CAPWAP-BASE-MIB DEFINITIONS ::= BEGIN

IMPORTS

   PhysAddress, TEXTUAL-CONVENTION, TruthValue,
   DateAndTime, RowStatus
       FROM SNMPv2-TC
   LongUtf8String
       FROM SYSAPPL-MIB
   InterfaceIndex, ifGeneralInformationGroup
       FROM IF-MIB
   PhysicalIndex
       FROM ENTITY-MIB
   SnmpAdminString
       FROM SNMP-FRAMEWORK-MIB
   NOTIFICATION-GROUP, OBJECT-GROUP, MODULE-COMPLIANCE
       FROM SNMPv2-CONF
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, mib-2,
   Integer32, Unsigned32, Counter32, Gauge32, TimeTicks
```

```
       FROM SNMPv2-SMI
   InetAddressType, InetAddress
       FROM INET-ADDRESS-MIB;

capwapBaseMIB MODULE-IDENTITY
    LAST-UPDATED "201004300000Z"        -- 30 April 2010
    ORGANIZATION "IETF Control And Provisioning of Wireless Access
                  Points (CAPWAP) Working Group
                  http://www.ietf.org/html.charters/capwap-charter.html"
    CONTACT-INFO
        "General Discussion: capwap@frascone.com
         To Subscribe: http://lists.frascone.com/mailman/listinfo/capwap

         Yang Shi (editor)
         Hangzhou H3C Tech. Co., Ltd.
         Beijing R&D Center of H3C, Digital Technology Plaza
         NO. 9 Shangdi 9th Street, Haidian District
         Beijing  100085
         China
         Phone: +86 010 82775276
         Email: rishyang@gmail.com

         David T. Perkins (editor)
         228 Bayview Dr.
         San Carlos, CA  94070
         USA
         Phone: +1 408 394-8702
         Email: dperkins@dsperkins.com

         Chris Elliott (editor)
         1516 Kent St.
         Durham, NC  27707
         USA
         Phone: +1 919-308-1216
         Email: chelliot@pobox.com

         Yong Zhang (editor)
         Fortinet, Inc.
         1090 Kifer Road
         Sunnyvale, CA  94086
         USA
         Email: yzhang@fortinet.com"

    DESCRIPTION
        "Copyright (c) 2010 IETF Trust and the persons identified as
         authors of the code. All rights reserved.
```

```
          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (http://trustee.ietf.org/license-info).

          This version of this MIB module is part of RFC 5833;
          see the RFC itself for full legal notices.

          This MIB module contains managed object definitions for
          the CAPWAP Protocol."
      REVISION    "201004300000Z"
      DESCRIPTION
          "Initial version published as RFC 5833"
          ::= { mib-2 196 }

-- Textual Conventions

CapwapBaseWtpProfileIdTC ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "d"
      STATUS      current
      DESCRIPTION
          "Represents the unique identifier of a WTP profile."
      SYNTAX      Unsigned32 (0..4096)

CapwapBaseWtpIdTC ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "1x:"
      STATUS      current
      DESCRIPTION
          "Represents the unique identifier of a WTP instance.
           As usual, the Base MAC address of the WTP is used."
      SYNTAX      OCTET STRING (SIZE(6|8))

CapwapBaseStationIdTC ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "1x:"
      STATUS      current
      DESCRIPTION
          "Represents the unique identifier of a station instance.
           As usual, the MAC address of the station is used."
      SYNTAX      OCTET STRING (SIZE(6|8))

CapwapBaseRadioIdTC ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "d"
      STATUS      current
      DESCRIPTION
          "Represents the unique identifier of a radio on a WTP."
      SYNTAX      Unsigned32 (1..31)
```

```
CapwapBaseTunnelModeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Represents the tunneling modes of operation that are
         supported by a WTP.
         The WTP MAY support more than one option, represented by
         the bit field below:
           localBridging(0) - Local bridging mode
           dot3Tunnel(1)    - 802.3 frame tunnel mode
           nativeTunnel(2)  - Native frame tunnel mode"
    REFERENCE
        "Section 4.6.43 of CAPWAP Protocol Specification, RFC 5415."
    SYNTAX      BITS {
                  localBridging(0),
                  dot3Tunnel(1),
                  nativeTunnel(2)
                }

CapwapBaseMacTypeTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Represents the MAC mode of operation supported by a WTP.
         The following enumerated values are supported:
           localMAC(0) - Local-MAC mode
           splitMAC(1) - Split-MAC mode
           both(2)     - Both Local-MAC and Split-MAC
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object
         follows that rule."
    REFERENCE
        "Section 4.6.44 of CAPWAP Protocol Specification, RFC 5415."
    SYNTAX      INTEGER {
                  localMAC(0),
                  splitMAC(1),
                  both(2)
                }

CapwapBaseChannelTypeTC::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Represents the channel type for CAPWAP protocol.
         The following enumerated values are supported:
           data(1)    - Data channel
           control(2) - Control channel"
    SYNTAX      INTEGER {
                  data(1),
                  control(2)
                }
```

```
CapwapBaseAuthenMethodTC ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Represents the authentication credential type for a WTP.
         The following enumerated values are supported:
           other(1) - Other method, for example, vendor specific
           clear(2) - Clear text and no authentication
           x509(3)  - X.509 certificate authentication
           psk(4)   - Pre-Shared secret authentication
         As a mandatory requirement, CAPWAP control channel
         authentication SHOULD use DTLS, either by certificate or
         PSK.  For data channel authentication, DTLS is optional."
    SYNTAX      INTEGER {
                  other(1),
                  clear(2),
                  x509(3),
                  psk(4)
                }

-- Top-level components of this MIB module

-- Notifications
capwapBaseNotifications OBJECT IDENTIFIER
    ::= { capwapBaseMIB 0 }
-- Tables, Scalars
capwapBaseObjects OBJECT IDENTIFIER
    ::= { capwapBaseMIB 1 }
-- Conformance
capwapBaseConformance OBJECT IDENTIFIER
    ::= { capwapBaseMIB 2 }


-- AC Objects Group

capwapBaseAc OBJECT IDENTIFIER
    ::= { capwapBaseObjects 1 }

capwapBaseWtpSessions OBJECT-TYPE
    SYNTAX      Gauge32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the total number of WTPs that are connecting to
         the AC."
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 1 }
```

```
capwapBaseWtpSessionsLimit  OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum number of WTP sessions configured on
         the AC.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 2 }

capwapBaseStationSessions  OBJECT-TYPE
    SYNTAX      Gauge32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the total number of stations that are accessing
         the wireless service provided by the AC."
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 3 }

capwapBaseStationSessionsLimit  OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum number of station sessions configured
         on the AC.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 4 }

capwapBaseDataChannelDTLSPolicyOptions  OBJECT-TYPE
    SYNTAX      BITS {
                  other(0),
                  clear(1),
                  dtls(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The AC communicates its policy on the use of DTLS for
         the CAPWAP data channel.
         The AC MAY support more than one option, represented by the bit
         field below:
```

```
            other(0) - Other method, for example, vendor specific
            clear(1) - Clear text
            dtls(2)  - DTLS"
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 5 }

capwapBaseControlChannelAuthenOptions  OBJECT-TYPE
    SYNTAX      BITS {
                  x509(0),
                  psk(1)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the authentication credential type supported by the
         AC for CAPWAP control channel.
         The AC MAY support more than one option, represented by the bit
         field below:
           x509(0) - X.509 certificate based
           psk(1)  - Pre-Shared secret"
    REFERENCE
        "Section 4.6.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 6 }

-- capwapBaseAcNameListTable table

capwapBaseAcNameListTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseAcNameListEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that configure the AC name list.
         Values of all read-create objects in this table are
         persistent at restart/reboot."
    REFERENCE
        "Section 4.6.5 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 9 }

capwapBaseAcNameListEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseAcNameListEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that configures the AC name list."
    INDEX { capwapBaseAcNameListId }
    ::= { capwapBaseAcNameListTable 1 }
```

```
CapwapBaseAcNameListEntry ::= SEQUENCE {
      capwapBaseAcNameListId          Unsigned32,
      capwapBaseAcNameListName        LongUtf8String,
      capwapBaseAcNameListPriority    Unsigned32,
      capwapBaseAcNameListRowStatus   RowStatus
    }

capwapBaseAcNameListId OBJECT-TYPE
    SYNTAX      Unsigned32 (1..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of an AC Name list."
    ::= { capwapBaseAcNameListEntry 1 }

capwapBaseAcNameListName OBJECT-TYPE
    SYNTAX      LongUtf8String (SIZE(1..512))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the name of an AC, and it is expected to be
         an UTF-8 encoded string."
    REFERENCE
        "Section 4.6.5 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAcNameListEntry 2 }

capwapBaseAcNameListPriority OBJECT-TYPE
    SYNTAX      Unsigned32 (1..255)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the priority order of the preferred AC.
         For instance, the value of one (1) is used to set the primary
         AC, the value of two (2) is used to set the secondary AC, etc."
    REFERENCE
        "Section 4.6.5 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAcNameListEntry 3 }

capwapBaseAcNameListRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used to create, modify, and/or delete a row
         in this table.
         The value of capwapBaseAcNameListName and
         capwapBaseAcNameListPriority can be changed when this
         object is in state 'active' or in 'notInService'.
```

```
          The capwapBaseAcNameListRowStatus may be changed to 'active'
          if all the managed objects in the conceptual row with
          MAX-ACCESS read-create have been assigned valid values."
    ::= { capwapBaseAcNameListEntry 4 }

-- End of capwapBaseAcNameListTable table


-- capwapBaseMacAclTable table

capwapBaseMacAclTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseMacAclEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that configure station Access Control
         Lists (ACLs).
         The WTP will not provide service to the MAC addresses
         configured in this table.
         Values of all read-create objects in this table are persistent
         at AC restart/reboot."
    REFERENCE
        "Section 4.6.7 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseAc 10 }

capwapBaseMacAclEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseMacAclEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that configures station Access Control
         Lists (ACLs)."
    INDEX { capwapBaseMacAclId }
    ::= { capwapBaseMacAclTable 1 }

CapwapBaseMacAclEntry ::= SEQUENCE {
      capwapBaseMacAclId           Unsigned32,
      capwapBaseMacAclStationId    CapwapBaseStationIdTC,
      capwapBaseMacAclRowStatus    RowStatus
    }

capwapBaseMacAclId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of an ACL."
    ::= { capwapBaseMacAclEntry 1 }
```

```
capwapBaseMacAclStationId OBJECT-TYPE
    SYNTAX      CapwapBaseStationIdTC
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the MAC address of a station to which WTPs will
         no longer provides service."
    REFERENCE
        "Section 4.6.7 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseMacAclEntry 2 }

capwapBaseMacAclRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used to create, modify, and/or delete a row
         in this table.
         The value of capwapBaseMacAclStationId can be changed when
         this object is in state 'active' or in 'notInService'.
         The capwapBaseMacAclRowStatus may be changed to 'active'
         if all the managed objects in the conceptual row with
         MAX-ACCESS read-create have been assigned valid values."
    ::= { capwapBaseMacAclEntry 3 }

-- End of capwapBaseMacAclTable table

-- End of AC Objects Group


-- WTP Objects Group

capwapBaseWtps OBJECT IDENTIFIER
    ::= { capwapBaseObjects 2 }

-- capwapBaseWtpProfileTable Table

capwapBaseWtpProfileTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseWtpProfileEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that configure WTP profiles for WTPs to
         be managed before they connect to the AC.
         An operator could change a WTP's configuration by changing
         the values of parameters in the corresponding WTP profile,
         then the WTP could get the new configuration through the
         CAPWAP control channel.
```

```
         Values of all read-create objects in this table are
         persistent at restart/reboot."
    ::= { capwapBaseWtps 1 }

capwapBaseWtpProfileEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseWtpProfileEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that configures and displays a WTP profile."
    INDEX { capwapBaseWtpProfileId }
    ::= { capwapBaseWtpProfileTable 1 }

CapwapBaseWtpProfileEntry ::= SEQUENCE {
      capwapBaseWtpProfileId                  CapwapBaseWtpProfileIdTC,
      capwapBaseWtpProfileName                       SnmpAdminString,
      capwapBaseWtpProfileWtpMacAddress              CapwapBaseWtpIdTC,
      capwapBaseWtpProfileWtpModelNumber             SnmpAdminString,
      capwapBaseWtpProfileWtpName                    LongUtf8String,
      capwapBaseWtpProfileWtpLocation                LongUtf8String,
      capwapBaseWtpProfileWtpStaticIpEnable          TruthValue,
      capwapBaseWtpProfileWtpStaticIpType            InetAddressType,
      capwapBaseWtpProfileWtpStaticIpAddress         InetAddress,
      capwapBaseWtpProfileWtpNetmask                 InetAddress,
      capwapBaseWtpProfileWtpGateway                 InetAddress,
      capwapBaseWtpProfileWtpFallbackEnable          INTEGER,
      capwapBaseWtpProfileWtpEchoInterval            Unsigned32,
      capwapBaseWtpProfileWtpIdleTimeout             Unsigned32,
      capwapBaseWtpProfileWtpMaxDiscoveryInterval    Unsigned32,
      capwapBaseWtpProfileWtpReportInterval          Unsigned32,
      capwapBaseWtpProfileWtpStatisticsTimer         Unsigned32,
      capwapBaseWtpProfileWtpEcnSupport              INTEGER,
      capwapBaseWtpProfileRowStatus                  RowStatus
    }

capwapBaseWtpProfileId OBJECT-TYPE
    SYNTAX      CapwapBaseWtpProfileIdTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WTP profile."
    ::= { capwapBaseWtpProfileEntry 1 }

capwapBaseWtpProfileName  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
```

```
        "Represents the name of a WTP profile."
    ::= { capwapBaseWtpProfileEntry 2 }

capwapBaseWtpProfileWtpMacAddress  OBJECT-TYPE
    SYNTAX      CapwapBaseWtpIdTC
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the Base MAC address of a WTP.
         A WTP profile MUST contain the Base MAC address of the WTP
         because the CAPWAP message received from the WTP contains
         its Base MAC address and the AC uses the Base MAC address to
         find the corresponding WTP profile.
         Section 4.6.40 of [RFC5415] omits indicating that the WTP's
         Base MAC address must be included in the WTP Board Data
         message element.  This is a known errata item and should be
         fixed in any future revision of the RFC 5415."
    REFERENCE
        "Section 4.6.40 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 3 }

capwapBaseWtpProfileWtpModelNumber  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the model number of a WTP.
         A WTP profile MUST include the WTP's model number, which
         reflects the number of Physical Layer (PHY) radios on the WTP.
         In this way, the creation of a WTP profile triggers the AC
         to automatically create the same number of WTP Virtual Radio
         Interfaces corresponding to the WTP's PHY radios without
         manual intervention.  With the ifIndexes of WTP Virtual
         Radio Interfaces, the operator could configure and manage
         the WTP's PHY radios through the wireless binding MIB
         modules."
    REFERENCE
        "Section 4.6.40 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 4 }

capwapBaseWtpProfileWtpName  OBJECT-TYPE
    SYNTAX      LongUtf8String (SIZE(1..512))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the name of the WTP."
    REFERENCE
        "Section 4.6.45 of CAPWAP Protocol Specification, RFC 5415."
```

```
    ::= { capwapBaseWtpProfileEntry 5 }

capwapBaseWtpProfileWtpLocation  OBJECT-TYPE
    SYNTAX      LongUtf8String (SIZE(1..1024))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the location of the WTP."
    REFERENCE
        "Section 4.6.30 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 6 }

capwapBaseWtpProfileWtpStaticIpEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents whether the WTP SHOULD use a static IP address
         or not.  A value of false disables the static IP address,
         while a value of true enables it."
    REFERENCE
        "Section 4.6.48 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 7 }

capwapBaseWtpProfileWtpStaticIpType OBJECT-TYPE
    SYNTAX      InetAddressType {ipv4(1)}
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the static IP address type used by the WTP.
         Only ipv4(1) is supported by the object.
         Although the CAPWAP protocol [RFC5415] supports both IPv4
         and IPv6, note that the CAPWAP field modeled by this
         object does not support IPv6, so the object does not
         support ipv6(2)."
    REFERENCE
        "Section 4.6.48 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 8 }

capwapBaseWtpProfileWtpStaticIpAddress OBJECT-TYPE
    SYNTAX      InetAddress (SIZE(4))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "When capwapBaseWtpProfileWtpStaticIpEnable is true,
         it represents the static IP address to be assigned to the WTP.
         The format of this IP address is determined by the
         corresponding instance of object
```

```
         capwapBaseWtpProfileWtpStaticIpType."
    REFERENCE
        "Section 4.6.48 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 9 }

capwapBaseWtpProfileWtpNetmask OBJECT-TYPE
    SYNTAX      InetAddress (SIZE(4))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "When capwapBaseWtpProfileWtpStaticIpEnable is true,
         it represents the netmask to be assigned to the WTP.
         The format of this netmask is determined by the
         corresponding instance of object
         capwapBaseWtpProfileWtpStaticIpType."
    REFERENCE
        "Section 4.6.48 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 10 }

capwapBaseWtpProfileWtpGateway OBJECT-TYPE
    SYNTAX      InetAddress (SIZE(4))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "When capwapBaseWtpProfileWtpStaticIpEnable is true,
         it represents the gateway to be assigned to the WTP.
         The format of this IP address is determined by the
         corresponding instance of object
         capwapBaseWtpProfileWtpStaticIpType."
    REFERENCE
        "Section 4.6.48 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 11 }

capwapBaseWtpProfileWtpFallbackEnable OBJECT-TYPE
    SYNTAX      INTEGER {
                  enabled(1),
                  disabled(2)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents whether to enable or disable automatic CAPWAP
         fallback in the event that a WTP detects its preferred AC
         and is not currently connected to it.
         The following enumerated values are supported:
           enabled(1)  - The fallback mode is enabled
           disabled(2) - The fallback mode is disabled"
    REFERENCE
```

```
        "Section 4.6.42 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { enabled }
     ::= { capwapBaseWtpProfileEntry 12 }

capwapBaseWtpProfileWtpEchoInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the minimum time, in seconds, between sending Echo
         Request messages to the AC that the WTP has joined."
    REFERENCE
        "Section 4.7.7 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 30 }
    ::= { capwapBaseWtpProfileEntry 13 }

capwapBaseWtpProfileWtpIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the idle timeout value that the WTP SHOULD enforce
         for its active stations."
    REFERENCE
        "Section 4.7.8 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 300 }
    ::= { capwapBaseWtpProfileEntry 14 }

capwapBaseWtpProfileWtpMaxDiscoveryInterval OBJECT-TYPE
    SYNTAX      Unsigned32 (2..180)
    UNITS       "second"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the maximum time allowed between sending Discovery
         Request messages, in seconds."
    REFERENCE
        "Section 4.7.10 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 20 }
    ::= { capwapBaseWtpProfileEntry 15 }

capwapBaseWtpProfileWtpReportInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-create
    STATUS      current
```

```
    DESCRIPTION
        "Represents the interval for WTP to send the Decryption Error
         report."
    REFERENCE
        "Section 4.7.11 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 120 }
    ::= { capwapBaseWtpProfileEntry 16 }

capwapBaseWtpProfileWtpStatisticsTimer OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the interval the WTP uses between the WTP Event
         Requests it transmits to the AC to communicate its statistics,
         in seconds."
    REFERENCE
        "Section 4.7.14 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 120 }
    ::= { capwapBaseWtpProfileEntry 17 }

capwapBaseWtpProfileWtpEcnSupport OBJECT-TYPE
    SYNTAX      INTEGER {
                  limited(0),
                  fullAndLimited(1)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents the support for the Explicit Congestion Notification
         (ECN) bits, as defined in [RFC3168].
         The following enumerated values are supported:
           limited(0)        - Limited ECN support
           fullAndLimited(1) - Full and limited ECN support
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object follows
         that rule."
    REFERENCE
        "Section 4.6.25 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpProfileEntry 18 }

capwapBaseWtpProfileRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used to create, modify, and/or delete a row
```

```
        in this table.
        The value of capwapBaseWtpProfileName,
        capwapBaseWtpProfileWtpName and capwapBaseWtpProfileWtpLocation
        can be changed when this object is in state 'active' or in
        'notInService'.
        The other objects in a row can be modified only when the value
        of this object in the corresponding conceptual row is not
        'active'.  Thus, to modify one or more of the objects in
        this conceptual row:
             a. change the row status to 'notInService'
             b. change the values of the row
             c. change the row status to 'active'
        The capwapBaseWtpProfileRowStatus may be changed to 'active'
        if the managed objects capwapBaseWtpProfileName,
        capwapBaseWtpProfileWtpMacAddress,
        capwapBaseWtpProfileWtpModelNumber,
        capwapBaseWtpProfileWtpName, and
        capwapBaseWtpProfileWtpLocation in the conceptual row
        have been assigned valid values.

        Deleting a WTP profile in use will disconnect the WTP from
        the AC.  So the network management system SHOULD
        ask the operator to confirm such an operation.
        When a WTP profile entry is removed from the table,
        the corresponding WTP Virtual Radio Interfaces are also
        removed from the capwapBaseWirelessBindingTable and
        ifTable [RFC2863].
        Also, the related object instances SHOULD be removed from
        the wireless binding MIB modules such as the IEEE 802.11
        MIB module [IEEE.802-11.2007]."
    ::= { capwapBaseWtpProfileEntry 19 }

-- End of capwapBaseWtpProfileTable table


-- capwapBaseWtpStateTable table

capwapBaseWtpStateTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseWtpStateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that indicate the AC's CAPWAP FSM state
         for each WTP, and helps the operator to query a WTP's
         current configuration."
    ::= { capwapBaseWtps 2 }

capwapBaseWtpStateEntry  OBJECT-TYPE
```

```
    SYNTAX      CapwapBaseWtpStateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays the AC's CAPWAP FSM state
         for each WTP.
         Also, the operator could query the current configuration
         of a WTP by using the identifier of the corresponding
         WTP profile."
    INDEX { capwapBaseWtpStateWtpId }
    ::= { capwapBaseWtpStateTable 1 }

CapwapBaseWtpStateEntry ::= SEQUENCE {
      capwapBaseWtpStateWtpId                   CapwapBaseWtpIdTC,
      capwapBaseWtpStateWtpIpAddressType        InetAddressType,
      capwapBaseWtpStateWtpIpAddress            InetAddress,
      capwapBaseWtpStateWtpLocalIpAddressType   InetAddressType,
      capwapBaseWtpStateWtpLocalIpAddress       InetAddress,
      capwapBaseWtpStateWtpBaseMacAddress       PhysAddress,
      capwapBaseWtpState                        INTEGER,
      capwapBaseWtpStateWtpUpTime               TimeTicks,
      capwapBaseWtpStateWtpCurrWtpProfileId     CapwapBaseWtpProfileIdTC
    }

capwapBaseWtpStateWtpId OBJECT-TYPE
    SYNTAX      CapwapBaseWtpIdTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WTP."
    ::= { capwapBaseWtpStateEntry 1 }

capwapBaseWtpStateWtpIpAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the IP address type of a WTP.
         Only ipv4(1) and ipv6(2) are supported by the object."
    ::= { capwapBaseWtpStateEntry 2 }

capwapBaseWtpStateWtpIpAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the IP address of a WTP that corresponds to
         the IP address in the IP packet header.
```

```
         The format of this IP address is determined by
         the corresponding instance of object
         capwapBaseWtpStateWtpIpAddressType."
    REFERENCE
        "Section 4 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpStateEntry 3 }

capwapBaseWtpStateWtpLocalIpAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the local IP address type of a WTP.
         Only ipv4(1) and ipv6(2) are supported by the object."
    ::= { capwapBaseWtpStateEntry 4 }

capwapBaseWtpStateWtpLocalIpAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the local IP address of a WTP and models the
         CAPWAP Local IPv4 Address or CAPWAP Local IPv6 Address fields
         [RFC5415].
         If a Network Address Translation (NAT) device is present
         between WTP and AC, the value of
         capwapBaseWtpStateWtpLocalIpAddress will be different from the
         value of capwapBaseWtpStateWtpIpAddress.
         The format of this IP address is determined by
         the corresponding instance of object
         capwapBaseWtpStateWtpLocalIpAddressType."
    REFERENCE
        "Sections 4.6.11 and 4.6.12 of CAPWAP Protocol Specification,
         RFC 5415."
    ::= { capwapBaseWtpStateEntry 5 }

capwapBaseWtpStateWtpBaseMacAddress  OBJECT-TYPE
    SYNTAX      PhysAddress (SIZE(6|8))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the WTP's Base MAC Address, which MAY be
         assigned to the primary Ethernet interface.
         The instance of the object corresponds to the Base MAC Address
         sub-element in the CAPWAP protocol [RFC5415]."
    REFERENCE
        "Section 4.6.40 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpStateEntry 6 }
```

```
capwapBaseWtpState  OBJECT-TYPE
    SYNTAX      INTEGER {
                  dtls(1),
                  join(2),
                  image(3),
                  configure(4),
                  dataCheck(5),
                  run(6),
                  reset(7),
                  dtlsTeardown(8),
                  unknown(9)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the various possibilities of the AC's CAPWAP FSM
         state for each WTP.
         The following enumerated values are supported:
           dtls(1)         - DTLS negotiation states, which include
                             DTLS setup, authorize, DTLS connect
           join(2)         - The WTP is joining with the AC
           image(3)        - The WTP is downloading software
           configure(4)    - The WTP is getting configuration from
                             the AC
           dataCheck(5)    - The AC is waiting for the Data Channel Keep
                             Alive Packet
           run(6)          - The WTP enters the running state
           reset(7)        - The AC transmits a reset request message
                             to the WTP
           dtlsTeardown(8) - DTLS session is torn down
           unknown(9)      - Operator already prepared configuration
                             for the WTP, while the WTP has not
                             contacted the AC until now"
    REFERENCE
        "Section 2.3.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpStateEntry 7 }

capwapBaseWtpStateWtpUpTime  OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the time (in hundredths of a second) since the
         WTP has been in the running state (corresponding to the
         value run(6) of capwapBaseWtpState)."
    ::= { capwapBaseWtpStateEntry 8 }

capwapBaseWtpStateWtpCurrWtpProfileId OBJECT-TYPE
```

```
    SYNTAX      CapwapBaseWtpProfileIdTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the current identifier of a WTP profile.
         The operator could query a WTP's current configuration
         with the identifier of a WTP profile."
    ::= { capwapBaseWtpStateEntry 9 }

-- End of capwapBaseWtpStateTable Table


-- capwapBaseWtpTable Table

capwapBaseWtpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseWtpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display properties of the WTPs
         in running state."
    ::= { capwapBaseWtps 3 }

capwapBaseWtpEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseWtpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays properties of the WTPs
         in running state."
    INDEX { capwapBaseWtpCurrId }
    ::= { capwapBaseWtpTable 1 }

CapwapBaseWtpEntry ::= SEQUENCE {
      capwapBaseWtpCurrId                    CapwapBaseWtpIdTC,
      capwapBaseWtpPhyIndex                  PhysicalIndex,
      capwapBaseWtpBaseMacAddress            PhysAddress,
      capwapBaseWtpTunnelModeOptions         CapwapBaseTunnelModeTC,
      capwapBaseWtpMacTypeOptions            CapwapBaseMacTypeTC,
      capwapBaseWtpDiscoveryType             INTEGER,
      capwapBaseWtpRadiosInUseNum            Gauge32,
      capwapBaseWtpRadioNumLimit             Unsigned32,
      capwapBaseWtpRetransmitCount           Counter32
    }

capwapBaseWtpCurrId OBJECT-TYPE
    SYNTAX      CapwapBaseWtpIdTC
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WTP in running state."
    ::= { capwapBaseWtpEntry 1 }

capwapBaseWtpPhyIndex OBJECT-TYPE
    SYNTAX      PhysicalIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the unique physical index of a physical entity
         in the ENTITY-MIB module [RFC4133].
         Information about a specific WTP such as its software version
         could be accessed through this index."
    ::= { capwapBaseWtpEntry 2 }

capwapBaseWtpBaseMacAddress  OBJECT-TYPE
    SYNTAX      PhysAddress (SIZE(6|8))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the WTP's Base MAC Address, which MAY be assigned
         to the primary Ethernet interface.
         The instance of the object corresponds to the Base MAC Address
         sub-element in the CAPWAP protocol [RFC5415]."
    REFERENCE
        "Section 4.6.40 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 3 }

capwapBaseWtpTunnelModeOptions  OBJECT-TYPE
    SYNTAX      CapwapBaseTunnelModeTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the tunneling modes of operation supported by
         the WTP."
    REFERENCE
        "Section 4.6.43 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 4 }

capwapBaseWtpMacTypeOptions  OBJECT-TYPE
    SYNTAX      CapwapBaseMacTypeTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the MAC mode of operation supported by the WTP."
    REFERENCE
        "Section 4.6.44 of CAPWAP Protocol Specification, RFC 5415."
```

```
    ::= { capwapBaseWtpEntry 5 }

capwapBaseWtpDiscoveryType OBJECT-TYPE
    SYNTAX      INTEGER {
                  unknown(0),
                  staticConfig(1),
                  dhcp(2),
                  dns(3),
                  acRef(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents how the WTP discovers the AC.
         The following enumerated values are supported:
           unknown(0)      - Unknown
           staticConfig(1) - Static configuration
           dhcp(2)         - DHCP
           dns(3)          - DNS
           acRef(4)        - AC referral
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object
         follows that rule."
    REFERENCE
        "Section 4.6.21 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 6 }

capwapBaseWtpRadiosInUseNum OBJECT-TYPE
    SYNTAX      Gauge32 (0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of radios in use on the WTP."
    REFERENCE
        "Section 4.6.41 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 7 }

capwapBaseWtpRadioNumLimit OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the maximum radio number supported by the WTP."
    REFERENCE
        "Section 4.6.41 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 8 }

capwapBaseWtpRetransmitCount OBJECT-TYPE
```

```
    SYNTAX      Counter32
    UNITS       "retransmissions"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of retransmissions for a given
         CAPWAP packet."
    REFERENCE
        "Section 4.8.8 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEntry 9 }

-- End of capwapBaseWtpTable table


-- capwapBaseWirelessBindingTable Table

capwapBaseWirelessBindingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseWirelessBindingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display the mappings between
         WTP Virtual Radio Interfaces and PHY radios, and the
         wireless binding type for each PHY radio.
         As capwapBaseWirelessBindingTable stores the mappings between
         PHY radios (Radio IDs) and the ifIndexes of WTP Virtual Radio
         Interfaces, the operator can get the ifIndex information by
         querying this table.  Such a query operation SHOULD run from
         radio ID 1 to radio ID 31 according to [RFC5415],
         and stop when an invalid ifIndex value (0) is returned.
         Values of all objects in this table are persistent at
         restart/reboot."
    ::= { capwapBaseWtps 4 }

capwapBaseWirelessBindingEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseWirelessBindingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays the mapping between
         a specific WTP Virtual Radio Interface and a PHY
         radio, and the wireless binding type for the PHY radio."
    INDEX {
      capwapBaseWtpProfileId,
      capwapBaseWirelessBindingRadioId
    }
    ::= { capwapBaseWirelessBindingTable 1 }
```

```
CapwapBaseWirelessBindingEntry ::= SEQUENCE {
      capwapBaseWirelessBindingRadioId              CapwapBaseRadioIdTC,
      capwapBaseWirelessBindingVirtualRadioIfIndex  InterfaceIndex,
      capwapBaseWirelessBindingType                 INTEGER
    }

capwapBaseWirelessBindingRadioId OBJECT-TYPE
    SYNTAX      CapwapBaseRadioIdTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the identifier of a PHY radio on a WTP, which
         is required to be unique on a WTP.
         For example, WTP A and WTP B use a same value of
         capwapBaseWirelessBindingRadioId for their first radio."
    REFERENCE
        "Section 4.3 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWirelessBindingEntry 1 }

capwapBaseWirelessBindingVirtualRadioIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the index value that uniquely identifies a
         WLAN Virtual Radio Interface.  The interface identified by
         a particular value of this index is the same interface as
         identified by the same value of the ifIndex.
         Before WTPs contact the AC to get configuration,
         the operator configures WTP profiles for them.
         The creation of a WTP profile triggers the system to
         automatically create a specific number of WTP Virtual Radio
         Interfaces and add a new row object in the
         capwapBaseWirelessBindingTable without manual intervention.
         As most MIB modules use the ifIndex to identify an interface
         for configuration and statistical data (for example, the IEEE
         802.11 MIB module [IEEE.802-11.2007]), it will be easy to
         reuse other wireless binding MIB modules through the WTP
         Virtual Radio Interface in the Centralized WLAN
         Architecture."
    ::= { capwapBaseWirelessBindingEntry 2 }

capwapBaseWirelessBindingType OBJECT-TYPE
    SYNTAX      INTEGER {
                  dot11(1),
                  epc(3)
                }
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
        "Represents the wireless binding type for the radio.
         The following enumerated values are supported:
           dot11(1) - IEEE 802.11
           epc(3)   - EPCGlobal"
    REFERENCE
        "Section 4.3 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWirelessBindingEntry 3 }

-- End of capwapBaseWirelessBindingTable Table


-- capwapBaseStationTable Table

capwapBaseStationTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseStationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display stations that are accessing
         the wireless service provided by the AC."
    REFERENCE
        "Section 4.6.8 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtps 5 }

capwapBaseStationEntry OBJECT-TYPE
    SYNTAX      CapwapBaseStationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays a station that is
         associated with the specific radio on the WTP.
         Note that in some cases such as roaming that a station may
         simultaneously associate with two WTPs for some (short) time.
         The MIB implementation MUST ensure there is only one valid
         and meaningful entry for a specific station."
    INDEX { capwapBaseStationId }
    ::= { capwapBaseStationTable 1 }

CapwapBaseStationEntry ::= SEQUENCE {
      capwapBaseStationId            CapwapBaseStationIdTC,
      capwapBaseStationWtpId         CapwapBaseWtpIdTC,
      capwapBaseStationWtpRadioId    CapwapBaseRadioIdTC,
      capwapBaseStationAddedTime     DateAndTime,
      capwapBaseStationVlanName      SnmpAdminString
    }
```

```
capwapBaseStationId OBJECT-TYPE
    SYNTAX      CapwapBaseStationIdTC
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of the station."
    REFERENCE
        "Section 4.6.8 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStationEntry 1 }

capwapBaseStationWtpId OBJECT-TYPE
    SYNTAX      CapwapBaseWtpIdTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WTP in running state."
    ::= { capwapBaseStationEntry 2 }

capwapBaseStationWtpRadioId OBJECT-TYPE
    SYNTAX      CapwapBaseRadioIdTC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the identifier of a PHY radio on a WTP, which
         is required to be unique on a WTP.
         For example, WTP A and WTP B use a same value of
         capwapBaseStationWtpRadioId for their first radio."
    REFERENCE
        "Section 4.3 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStationEntry 3 }

capwapBaseStationAddedTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the time when the station is added."
    REFERENCE
        "Section 4.6.8 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStationEntry 4 }

capwapBaseStationVlanName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(0..32))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents VLAN name to which the station is associated."
    REFERENCE
```

```
        "Section 4.6.8 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStationEntry 5 }

-- End of capwapBaseStationTable Table


--  capwapBaseWtpEventsStatsTable

capwapBaseWtpEventsStatsTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseWtpEventsStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display the WTPs' events statistics."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtps 6 }

capwapBaseWtpEventsStatsEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseWtpEventsStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays the events statistics
         of a WTP."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    INDEX { capwapBaseWtpCurrId }
    ::= { capwapBaseWtpEventsStatsTable 1 }

CapwapBaseWtpEventsStatsEntry ::= SEQUENCE {
      capwapBaseWtpEventsStatsRebootCount           Counter32,
      capwapBaseWtpEventsStatsInitCount             Counter32,
      capwapBaseWtpEventsStatsLinkFailureCount      Counter32,
      capwapBaseWtpEventsStatsSwFailureCount        Counter32,
      capwapBaseWtpEventsStatsHwFailureCount        Counter32,
      capwapBaseWtpEventsStatsOtherFailureCount     Counter32,
      capwapBaseWtpEventsStatsUnknownFailureCount   Counter32,
      capwapBaseWtpEventsStatsLastFailureType       INTEGER
    }

capwapBaseWtpEventsStatsRebootCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of reboots that have occurred due to a
         WTP crash.
```

```
         Note that the CAPWAP field [RFC5415] modeled by this counter
         takes the value 65535 to indicate that the information is not
         available on the WTP.  This MIB object does not follow this
         behavior, which would not be standard in SMIv2.  If the WTP
         does not have the information, the agent will not instantiate
         the object."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 1 }

capwapBaseWtpEventsStatsInitCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of reboots that have occurred at the
         request of a CAPWAP protocol message, such as a change in
         configuration that requires a reboot or an explicit CAPWAP
         protocol reset request.
         Note that the CAPWAP field [RFC5415] modeled by this counter
         takes the value 65535 to indicate that the information is not
         available on the WTP.  This MIB object does not follow this
         behavior, which would not be standard in SMIv2.  If the WTP
         does not have the information, the agent will not instantiate
         the object."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 2 }

capwapBaseWtpEventsStatsLinkFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that a CAPWAP protocol
         connection with an AC has failed due to link failures."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 3 }

capwapBaseWtpEventsStatsSwFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that a CAPWAP protocol
         connection with an AC has failed due to software-related
         reasons."
```

```
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 4 }

capwapBaseWtpEventsStatsHwFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that a CAPWAP protocol
         connection with an AC has failed due to hardware-related
         reasons."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 5 }

capwapBaseWtpEventsStatsOtherFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that a CAPWAP protocol
         connection with an AC has failed due to known reasons, other
         than the AC-initiated, link, software or hardware failures."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 6 }

capwapBaseWtpEventsStatsUnknownFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that a CAPWAP protocol
         connection with an AC has failed for unknown reasons."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 7 }

capwapBaseWtpEventsStatsLastFailureType OBJECT-TYPE
    SYNTAX      INTEGER {
                  unsupported(0),
                  acInit(1),
                  linkFailure(2),
                  swFailure(3),
                  hwFailure(4),
                  otherFailure(5),
                  unknown(255)
```

```
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the failure type of the most recent WTP failure.
         The following enumerated values are supported:
           unsupported(0)  - Not supported
           acInit(1)       - The AC initiated
           linkFailure(2)  - Link failure
           swFailure(3)    - Software failure
           hwFailure(4)    - Hardware failure
           otherFailure(5) - Other failure
           unknown(255)    - Unknown (e.g., WTP doesn't keep track
                             of info)
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object
         follows that rule."
    REFERENCE
        "Section 4.6.47 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtpEventsStatsEntry 8 }

--  End of capwapBaseWtpEventsStatsTable table


--  capwapBaseRadioEventsStatsTable table

capwapBaseRadioEventsStatsTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF CapwapBaseRadioEventsStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display statistics on the radios'
         behaviors and reasons why the WTP radio has been reset.
         To get the events statistics of all radios on a specific WTP
         (identified by the capwapBaseWtpCurrId), a query
         operation SHOULD run from radio ID 1 to radio ID 31 until there
         is no data returned.  The radio ID here corresponds to the
         object capwapBaseRadioEventsWtpRadioId.  If the previous MIB
         operations such as query on the capwapBaseWirelessBindingTable
         know the exact value of each radio ID, the query operation on
         the capwapBaseRadioEventsStatsTable could use that value
         of Radio IDs."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseWtps 7 }

capwapBaseRadioEventsStatsEntry  OBJECT-TYPE
    SYNTAX      CapwapBaseRadioEventsStatsEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that displays the statistical data of
         events that happened on a specific radio of a WTP."
    INDEX { capwapBaseWtpCurrId, capwapBaseRadioEventsWtpRadioId }
    ::= { capwapBaseRadioEventsStatsTable 1 }

CapwapBaseRadioEventsStatsEntry ::= SEQUENCE {
      capwapBaseRadioEventsWtpRadioId             CapwapBaseRadioIdTC,
      capwapBaseRadioEventsStatsResetCount             Counter32,
      capwapBaseRadioEventsStatsSwFailureCount         Counter32,
      capwapBaseRadioEventsStatsHwFailureCount         Counter32,
      capwapBaseRadioEventsStatsOtherFailureCount      Counter32,
      capwapBaseRadioEventsStatsUnknownFailureCount    Counter32,
      capwapBaseRadioEventsStatsConfigUpdateCount      Counter32,
      capwapBaseRadioEventsStatsChannelChangeCount     Counter32,
      capwapBaseRadioEventsStatsBandChangeCount        Counter32,
      capwapBaseRadioEventsStatsCurrNoiseFloor         Integer32,
      capwapBaseRadioEventsStatsDecryptErrorCount      Counter32,
      capwapBaseRadioEventsStatsLastFailureType        INTEGER
    }

capwapBaseRadioEventsWtpRadioId OBJECT-TYPE
     SYNTAX      CapwapBaseRadioIdTC
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "Represents the identifier of a PHY radio on a WTP, which
          is required to be unique on a WTP.
          For example, WTP A and WTP B use the same value of
          capwapBaseRadioEventsWtpRadioId for their first radio."
     REFERENCE
         "Section 4.3 of CAPWAP Protocol Specification, RFC 5415."
     ::= { capwapBaseRadioEventsStatsEntry 1 }

capwapBaseRadioEventsStatsResetCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has been
         reset."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 2 }

capwapBaseRadioEventsStatsSwFailureCount OBJECT-TYPE
```

```
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has failed due
         to software-related reasons."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 3 }

capwapBaseRadioEventsStatsHwFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has failed due
         to hardware-related reasons."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 4 }

capwapBaseRadioEventsStatsOtherFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has failed due to
         known reasons, other than software or hardware failure."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 5 }

capwapBaseRadioEventsStatsUnknownFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has failed for
         unknown reasons."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 6 }

capwapBaseRadioEventsStatsConfigUpdateCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "Represents the number of times that the radio configuration has
         been updated."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 7 }

capwapBaseRadioEventsStatsChannelChangeCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio channel has
         been changed."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 8 }

capwapBaseRadioEventsStatsBandChangeCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of times that the radio has changed
         frequency bands."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 9 }

capwapBaseRadioEventsStatsCurrNoiseFloor OBJECT-TYPE
    SYNTAX      Integer32
    UNITS       "dBm"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the noise floor of the radio receiver in units of
         dBm."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 10 }

capwapBaseRadioEventsStatsDecryptErrorCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of decryption errors that have occurred
         on the WTP.  Note that this field is only valid in cases where
         the WTP provides encryption/decryption services."
```

```
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 11 }

capwapBaseRadioEventsStatsLastFailureType OBJECT-TYPE
    SYNTAX      INTEGER {
                  unsupported(0),
                  swFailure(1),
                  hwFailure(2),
                  otherFailure(3),
                  unknown(255)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the failure type of the most recent radio failure.
         The following enumerated values are supported:
           unsupported(0)  - Not supported
           swFailure(1)    - Software failure
           hwFailure(2)    - Hardware failure
           otherFailure(3) - Other failure
           unknown(255)    - Unknown
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object follows
         that rule."
    REFERENCE
        "Section 4.6.46 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseRadioEventsStatsEntry 12 }

--  End of capwapBaseRadioEventsStatsTable table

-- End of WTP Objects Group


-- CAPWAP Base Parameters Group

capwapBaseParameters OBJECT IDENTIFIER
    ::= { capwapBaseObjects 3 }

capwapBaseAcMaxRetransmit OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum number of retransmissions for a given
         CAPWAP packet before the link layer considers the peer dead.
         The value of the object is persistent at restart/reboot."
    REFERENCE
```

```
        "Section 4.8.7 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 5 }
    ::= { capwapBaseParameters 1 }

capwapBaseAcChangeStatePendingTimer OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum time, in seconds, the AC will wait
         for the Change State Event Request from the WTP after having
         transmitted a successful Configuration Status Response
         message.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.1 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 25 }
    ::= { capwapBaseParameters 2 }

capwapBaseAcDataCheckTimer OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents The number of seconds the AC will wait for
         the Data Channel Keep Alive, which is required by the
         CAPWAP state machine's Data Check state.
         The AC resets the state machine if this timer expires
         prior to transitioning to the next state.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.4 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 30 }
    ::= { capwapBaseParameters 3 }

capwapBaseAcDTLSSessionDeleteTimer OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the minimum time, in seconds, the AC MUST wait
         for DTLS session deletion.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.6 of CAPWAP Protocol Specification, RFC 5415."
```

```
    DEFVAL { 5 }
    ::= { capwapBaseParameters 4 }

capwapBaseAcEchoInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the minimum time, in seconds, between sending Echo
         Request messages to the AC with which the WTP has joined.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.7 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 30 }
    ::= { capwapBaseParameters 5 }

capwapBaseAcRetransmitInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the minimum time, in seconds, in which a
         non-acknowledged CAPWAP packet will be retransmitted.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.12 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 3 }
    ::= { capwapBaseParameters 6 }

capwapBaseAcSilentInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the minimum time, in seconds, during which the AC
         SHOULD ignore all CAPWAP and DTLS packets received from the
         WTP that is in the Sulking state.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.13 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 30 }
    ::= { capwapBaseParameters 7 }

capwapBaseAcWaitDTLSTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (30..4294967295)
```

```
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum time, in seconds, the AC MUST wait
         without having received a DTLS Handshake message from an AC.
         This timer MUST be greater than 30 seconds.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.15 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 60 }
    ::= { capwapBaseParameters 8 }

capwapBaseAcWaitJoinTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (20..4294967295)
    UNITS       "second"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the maximum time, in seconds, the AC will wait
         after the DTLS session has been established until it receives
         the Join Request from the WTP.  This timer MUST be greater
         than 20 seconds.
         The value of the object is persistent at restart/reboot."
    REFERENCE
        "Section 4.7.16 of CAPWAP Protocol Specification, RFC 5415."
    DEFVAL { 60 }
    ::= { capwapBaseParameters 9 }

capwapBaseAcEcnSupport OBJECT-TYPE
    SYNTAX      INTEGER {
                  limited(0),
                  fullAndLimited(1)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents the support for the Explicit Congestion Notification
         (ECN) bits, as defined in [RFC3168].
         The value of the object is persistent at restart/reboot.
         The following enumerated values are supported:
           limited(0)        - Limited ECN support
           fullAndLimited(1) - Full and limited ECN support
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object follows
         that rule."
    REFERENCE
        "Section 4.6.25 of CAPWAP Protocol Specification, RFC 5415."
```

```
    ::= { capwapBaseParameters 10 }

-- End of CAPWAP Base Parameters Group


-- CAPWAP Statistics Group

capwapBaseStats OBJECT IDENTIFIER
    ::= { capwapBaseObjects 4 }

capwapBaseFailedDTLSAuthFailureCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of failed DTLS session establishment
         attempts due to authentication failures."
    REFERENCE
        "Section 4.8.3 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStats 1 }

capwapBaseFailedDTLSSessionCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the number of failed DTLS session
         establishment attempts."
    REFERENCE
        "Section 4.8.4 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseStats 2 }

-- Notifications

capwapBaseChannelUp NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfChannelType,
                  capwapBaseNtfAuthenMethod
                }
    STATUS      current
    DESCRIPTION
        "This notification is sent by the AC when a CAPWAP channel
         is established.
         The notification is separated for data or control channel."
    ::= { capwapBaseNotifications 1 }

capwapBaseChannelDown NOTIFICATION-TYPE
```

```
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfChannelType,
                  capwapBaseNtfChannelDownReason
                }
    STATUS      current
    DESCRIPTION
        "This notification is sent by the AC when a CAPWAP channel
         is down.
         The notification is separated for data or control channel."
    ::= { capwapBaseNotifications 2 }

capwapBaseDecryptErrorReport NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfRadioId,
                  capwapBaseNtfStationIdList
                }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a WTP has had a
         decryption error since the last report."
    REFERENCE
        "Section 4.6.17 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifications 3 }

capwapBaseJoinFailure NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfJoinFailureReason
                }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a WTP fails to join."
    REFERENCE
        "Section 4.6.35 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifications 4 }

capwapBaseImageUpgradeFailure NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfImageFailureReason
                }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a WTP fails to update
         the firmware image."
    REFERENCE
```

```
        "Section 4.6.35 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifications 5 }

capwapBaseConfigMsgError NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfConfigMsgErrorType,
                  capwapBaseNtfMsgErrorElements
                }
    STATUS      current
    DESCRIPTION
        "This notification is generated when a WTP receives message
         elements in the configuration management messages that it
         is unable to apply locally."
    REFERENCE
        "Section 4.6.35 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifications 6 }

capwapBaseRadioOperableStatus NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfRadioId,
                  capwapBaseNtfRadioOperStatusFlag,
                  capwapBaseNtfRadioStatusCause
                }
    STATUS      current
    DESCRIPTION
        "The notification is generated when a radio's operational state
         has changed."
    REFERENCE
        "Section 4.6.34 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifications 7 }

capwapBaseAuthenFailure NOTIFICATION-TYPE
    OBJECTS     {
                  capwapBaseNtfWtpId,
                  capwapBaseNtfChannelType,
                  capwapBaseNtfAuthenMethod,
                  capwapBaseNtfAuthenFailureReason
                }
    STATUS      current
    DESCRIPTION
        "This is notification of an authentication failure event
         and provides the reason for it."
    ::= { capwapBaseNotifications 8 }

-- Objects used only in notifications
```

```
-- Notification Objects
capwapBaseNotifyVarObjects OBJECT IDENTIFIER
    ::= { capwapBaseObjects 5 }

capwapBaseNtfWtpId OBJECT-TYPE
    SYNTAX      CapwapBaseWtpIdTC
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the unique identifier of a WTP."
    ::= { capwapBaseNotifyVarObjects 1 }

capwapBaseNtfRadioId OBJECT-TYPE
    SYNTAX      CapwapBaseRadioIdTC
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the identifier of a PHY radio on a WTP, which is
         only required to be unique on a WTP.
         For example, WTP A and WTP B can use the same value of
         capwapBaseNtfRadioId for their first radio."
    REFERENCE
        "Section 4.3 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 2 }

capwapBaseNtfChannelType OBJECT-TYPE
    SYNTAX      CapwapBaseChannelTypeTC
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the channel type for the CAPWAP protocol."
    ::= { capwapBaseNotifyVarObjects 3 }

capwapBaseNtfAuthenMethod OBJECT-TYPE
    SYNTAX      CapwapBaseAuthenMethodTC
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the authentication method for the CAPWAP Channel."
    ::= { capwapBaseNotifyVarObjects 4 }

capwapBaseNtfChannelDownReason OBJECT-TYPE
    SYNTAX      INTEGER {
                  timeout(1),
                  rekeyFailure(2),
                  acRebootWtp(3),
                  dtlsError(4),
                  maxRetransmit(5)
```

```
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the reason the channel is down.
         The following enumerated values are supported:
           timeout(1)       - The keepalive timed out
           rekeyFailure(2)  - Rekey process failed; channel will be
                              broken
           acRebootWtp(3)   - The AC rebooted the WTP
           dtlsError(4)     - DTLS notifications: DTLSAborted,
                              DTLSReassemblyFailure, DTLSPeerDisconnect,
                              or frequent DTLSDecapFailure
           maxRetransmit(5) - The underlying reliable transport's
                              RetransmitCount counter has reached the
                              MaxRetransmit variable"
    ::= { capwapBaseNotifyVarObjects 5 }

capwapBaseNtfStationIdList OBJECT-TYPE
    SYNTAX      LongUtf8String (SIZE (6..1024))
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents a list of station MAC addresses separated by
         semicolons."
    REFERENCE
        "Section 4.6.17 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 6 }

capwapBaseNtfAuthenFailureReason OBJECT-TYPE
    SYNTAX      INTEGER {
                  keyMismatch(1),
                  invalidCert(2),
                  reassemblyFailure(3),
                  decapFailure(4),
                  encapFailure(5),
                  timeout(6),
                  unknown(8)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the reason for WTP authorization failure.
         The following enumerated values are supported:
           keyMismatch(1)       - WTP's and AC's keys did not match
           invalidCert(2)       - Certification is not valid
           reassemblyFailure(3) - Fragment reassembly failure
           decapFailure(4)      - Decapsulation error
```

```
            encapFailure(5)      - Encapsulation error
            timeout(6)           - WaitDTLS timer timeout
            unknown(8)           - Unknown reason"
    REFERENCE
        "Section 2.3.1 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 7 }

capwapBaseNtfRadioOperStatusFlag OBJECT-TYPE
    SYNTAX      INTEGER {
                  operable(0),
                  inoperable(1)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the operation status of a radio.
         The following enumerated values are supported:
           operable(0)   - The radio is operable
           inoperable(1) - The radio is inoperable, and the
                           capwapBaseNtfRadioStatusCause object
                           gives the reason in detail
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object
         follows that rule."
    REFERENCE
        "Section 4.6.34 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 8 }

capwapBaseNtfRadioStatusCause OBJECT-TYPE
    SYNTAX      INTEGER {
                  normal(0),
                  hwError(1),
                  swError(2),
                  adminSet(3)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the reason why the radio is out of service.
         The following enumerated values are supported:
           normal(0)   - Normal status
           hwError(1)  - Radio failure
           swError(2)  - Software failure
           adminSet(3) - Administratively set
         Note that the CAPWAP field [RFC5415] modeled by this
         object takes zero as starting value; this MIB object
         follows that rule."
    REFERENCE
```

```
        "Section 4.6.34 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 9 }

capwapBaseNtfJoinFailureReason  OBJECT-TYPE
    SYNTAX      INTEGER {
                  unspecified(1),
                  resDepletion(2),
                  unknownSource(3),
                  incorrectData(4),
                  sessionIdInUse(5),
                  unsupportedHw(6),
                  unsupportedBinding(7)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the reason of join failure.
         The following enumerated values are supported:
           unspecified(1)        - Unspecified failure
           resDepletion(2)       - Resource depletion
           unknownSource(3)      - Unknown source
           incorrectData(4)      - Incorrect data
           sessionIdInUse(5)     - Session ID already in use
           unsupportedHw(6)      - WTP hardware not supported
           unsupportedBinding(7) - Binding not supported"
    REFERENCE
        "Section 4.6.35 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 10 }

capwapBaseNtfImageFailureReason  OBJECT-TYPE
    SYNTAX      INTEGER {
                  invalidChecksum(1),
                  invalidLength(2),
                  other(3),
                  inStorage(4)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the reason of image failure.
         The following enumerated values are supported:
           invalidChecksum(1) - Invalid checksum
           invalidLength(2)   - Invalid data length
           other(3)           - Other error
           inStorage(4)       - Image already present"
    REFERENCE
        "Section 4.6.35 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 11 }
```

```
capwapBaseNtfConfigMsgErrorType  OBJECT-TYPE
    SYNTAX      INTEGER {
                  unknownElement(1),
                  unsupportedElement(2),
                  unknownValue(3),
                  unsupportedValue(4)
                }
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the type of configuration message error.
         The following enumerated values are supported:
           unknownElement(1)     - Unknown message element
           unsupportedElement(2) - Unsupported message element
           unknownValue(3)       - Unknown message element value
           unsupportedValue(4)   - Unsupported message element value"
    REFERENCE
        "Section 4.6.36 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 12 }

capwapBaseNtfMsgErrorElements  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  accessible-for-notify
    STATUS      current
    DESCRIPTION
        "Represents the message elements sent by the AC in the
         Configuration Status Response message that caused the error."
    REFERENCE
        "Section 4.6.36 of CAPWAP Protocol Specification, RFC 5415."
    ::= { capwapBaseNotifyVarObjects 13 }

-- Notification Control
capwapBaseNotifyControlObjects OBJECT IDENTIFIER
    ::= { capwapBaseObjects 6 }

capwapBaseChannelUpDownNotifyEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the Channel Up / Channel Down notification
         should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { false }
    ::= { capwapBaseNotifyControlObjects 1 }
```

```
capwapBaseDecryptErrorNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the decryption error notification should
         be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { true }
    ::= { capwapBaseNotifyControlObjects 2 }

capwapBaseJoinFailureNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the notification of a WTP join failure
         should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { true }
    ::= { capwapBaseNotifyControlObjects 3 }

capwapBaseImageUpgradeFailureNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the notification of a WTP image upgrade
         failure should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { true }
    ::= { capwapBaseNotifyControlObjects 4 }

capwapBaseConfigMsgErrorNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the notification of configuration message
         error should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
```

```
         The value of the object is persistent at restart/reboot."
    DEFVAL { false }
    ::= { capwapBaseNotifyControlObjects 5 }

capwapBaseRadioOperableStatusNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the notification of a radio's operational
         state change should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { false }
    ::= { capwapBaseNotifyControlObjects 6 }

capwapBaseAuthenFailureNotifyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Represents whether the notification of authentication failure
         should be generated.
         A value of true(1) means that the notification is enabled.
         A value of false(2) means that the notification is disabled.
         The value of the object is persistent at restart/reboot."
    DEFVAL { true }
    ::= { capwapBaseNotifyControlObjects 7 }

-- Module compliance

capwapBaseCompliances OBJECT IDENTIFIER
    ::= { capwapBaseConformance 1 }

capwapBaseGroups OBJECT IDENTIFIER
    ::= { capwapBaseConformance 2 }

capwapBaseCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
       "Describes the requirements for conformance to the
        CAPWAP-BASE-MIB module."

    MODULE IF-MIB -- The Interfaces MIB, RFC 2863
    MANDATORY-GROUPS {
       ifGeneralInformationGroup
    }
```

```
    MODULE -- this module
     MANDATORY-GROUPS {
       capwapBaseAcNodeGroup,
       capwapBaseWtpProfileGroup,
       capwapBaseWtpStateGroup,
       capwapBaseWtpGroup,
       capwapBaseRadioGroup,
       capwapBaseStationGroup
     }

     GROUP capwapBaseAcNodeGroup2
     DESCRIPTION
         "The capwapBaseAcNodeGroup2 group is optional."

     GROUP capwapBaseAcNameListGroup
     DESCRIPTION
         "The capwapBaseAcNameListGroup group is optional."

     GROUP capwapBaseMacAclsGroup
     DESCRIPTION
         "The capwapBaseMacAclsGroup group is optional."

     GROUP capwapBaseWtpProfileGroup2
     DESCRIPTION
         "The capwapBaseWtpProfileGroup2 group is optional."

     GROUP capwapBaseWtpGroup2
     DESCRIPTION
         "The capwapBaseWtpGroup2 group is optional."

     GROUP capwapBaseWtpEventsStatsGroup
     DESCRIPTION
         "The capwapBaseWtpEventsStatsGroup group is optional."

     GROUP capwapBaseRadioEventsStatsGroup
     DESCRIPTION
         "The capwapBaseRadioEventsStatsGroup group is optional."

     GROUP capwapBaseParametersGroup
     DESCRIPTION
         "The capwapBaseParametersGroup group is optional."

     GROUP capwapBaseStatsGroup
     DESCRIPTION
         "The capwapBaseStatsGroup group is optional."

     GROUP capwapBaseNotificationsGroup
     DESCRIPTION
```

```
          "The capwapBaseNotificationsGroup group is optional."

     GROUP capwapBaseNotifyVarsGroup
     DESCRIPTION
         "The capwapBaseNotifyVarsGroup group is optional.
          If capwapBaseNotificationsGroup is supported,
          this group must be implemented."

     GROUP capwapBaseNotifyControlGroup
     DESCRIPTION
        "The capwapBaseNotifyControlGroup group is optional.
         If capwapBaseNotificationsGroup is supported,
         this group must be implemented."
     ::= { capwapBaseCompliances 1 }

capwapBaseAcNodeGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpSessions,
      capwapBaseWtpSessionsLimit,
      capwapBaseStationSessions,
      capwapBaseStationSessionsLimit
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the basic properties of the AC from the CAPWAP
         protocol perspective."
    ::= { capwapBaseGroups 1 }

capwapBaseAcNodeGroup2   OBJECT-GROUP
    OBJECTS {
      capwapBaseDataChannelDTLSPolicyOptions,
      capwapBaseControlChannelAuthenOptions
     }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the other properties (such as security) of the AC from
         the CAPWAP protocol perspective."
    ::= { capwapBaseGroups 2 }

capwapBaseAcNameListGroup  OBJECT-GROUP
    OBJECTS {
      capwapBaseAcNameListName,
      capwapBaseAcNameListPriority,
      capwapBaseAcNameListRowStatus
    }
    STATUS  current
```

```
    DESCRIPTION
        "A collection of objects that is used to configure
         the AC name list."
    ::= { capwapBaseGroups 3 }

capwapBaseMacAclsGroup  OBJECT-GROUP
    OBJECTS {
      capwapBaseMacAclStationId,
      capwapBaseMacAclRowStatus
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to configure
         the stations ACL."
    ::= { capwapBaseGroups 4 }

capwapBaseWtpProfileGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpProfileName,
      capwapBaseWtpProfileWtpMacAddress,
      capwapBaseWtpProfileWtpModelNumber,
      capwapBaseWtpProfileWtpName,
      capwapBaseWtpProfileWtpLocation,
      capwapBaseWtpProfileRowStatus
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to configure
         the WTP profile."
    ::= { capwapBaseGroups 5 }

capwapBaseWtpProfileGroup2    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpProfileWtpStaticIpEnable,
      capwapBaseWtpProfileWtpStaticIpType,
      capwapBaseWtpProfileWtpStaticIpAddress,
      capwapBaseWtpProfileWtpNetmask,
      capwapBaseWtpProfileWtpGateway,
      capwapBaseWtpProfileWtpFallbackEnable,
      capwapBaseWtpProfileWtpEchoInterval,
      capwapBaseWtpProfileWtpIdleTimeout,
      capwapBaseWtpProfileWtpMaxDiscoveryInterval,
      capwapBaseWtpProfileWtpReportInterval,
      capwapBaseWtpProfileWtpStatisticsTimer,
      capwapBaseWtpProfileWtpEcnSupport
    }
    STATUS  current
    DESCRIPTION
```

```
        "A collection of optional objects that is used to
         configure the WTP profile."
    ::= { capwapBaseGroups 6 }

capwapBaseWtpStateGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpStateWtpIpAddressType,
      capwapBaseWtpStateWtpIpAddress,
      capwapBaseWtpStateWtpLocalIpAddressType,
      capwapBaseWtpStateWtpLocalIpAddress,
      capwapBaseWtpStateWtpBaseMacAddress,
      capwapBaseWtpState,
      capwapBaseWtpStateWtpUpTime,
      capwapBaseWtpStateWtpCurrWtpProfileId
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the WTP's state information."
    ::= { capwapBaseGroups 7 }

capwapBaseWtpGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpBaseMacAddress,
      capwapBaseWtpTunnelModeOptions,
      capwapBaseWtpMacTypeOptions,
      capwapBaseWtpDiscoveryType,
      capwapBaseWtpRadiosInUseNum,
      capwapBaseWtpRadioNumLimit
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the properties information for the WTPs in running state."
    ::= { capwapBaseGroups 8 }

capwapBaseWtpGroup2   OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpPhyIndex,
      capwapBaseWtpRetransmitCount
    }
    STATUS  current
    DESCRIPTION
        "A collection of optional objects that is used to represent
         the properties of the WTPs in running state."
    ::= { capwapBaseGroups 9 }

capwapBaseRadioGroup    OBJECT-GROUP
```

```
    OBJECTS {
      capwapBaseWirelessBindingVirtualRadioIfIndex,
      capwapBaseWirelessBindingType
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the wireless binding type and the mappings between the
         ifIndexes of WLAN Virtual Radio Interfaces and PHY radios."
    ::= { capwapBaseGroups 10 }

capwapBaseStationGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseStationWtpId,
      capwapBaseStationWtpRadioId,
      capwapBaseStationAddedTime,
      capwapBaseStationVlanName
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used to represent
         the stations' basic properties."
    ::= { capwapBaseGroups 11 }

capwapBaseWtpEventsStatsGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseWtpEventsStatsRebootCount,
      capwapBaseWtpEventsStatsInitCount,
      capwapBaseWtpEventsStatsLinkFailureCount,
      capwapBaseWtpEventsStatsSwFailureCount,
      capwapBaseWtpEventsStatsHwFailureCount,
      capwapBaseWtpEventsStatsOtherFailureCount,
      capwapBaseWtpEventsStatsUnknownFailureCount,
      capwapBaseWtpEventsStatsLastFailureType
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used for collecting
         WTP reboot count, link failure count, hardware failure
         count, and so on."
    ::= { capwapBaseGroups 12 }

capwapBaseRadioEventsStatsGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseRadioEventsStatsResetCount,
      capwapBaseRadioEventsStatsSwFailureCount,
      capwapBaseRadioEventsStatsHwFailureCount,
      capwapBaseRadioEventsStatsOtherFailureCount,
```

```
      capwapBaseRadioEventsStatsUnknownFailureCount,
      capwapBaseRadioEventsStatsConfigUpdateCount,
      capwapBaseRadioEventsStatsChannelChangeCount,
      capwapBaseRadioEventsStatsBandChangeCount,
      capwapBaseRadioEventsStatsCurrNoiseFloor,
      capwapBaseRadioEventsStatsDecryptErrorCount,
      capwapBaseRadioEventsStatsLastFailureType
    }
    STATUS  current
    DESCRIPTION
        "A collection of objects that is used for collecting
         radio reset count, channel change count, hardware failure
         count, and so on"
    ::= { capwapBaseGroups 13 }

capwapBaseParametersGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseAcMaxRetransmit,
      capwapBaseAcChangeStatePendingTimer,
      capwapBaseAcDataCheckTimer,
      capwapBaseAcDTLSSessionDeleteTimer,
      capwapBaseAcEchoInterval,
      capwapBaseAcRetransmitInterval,
      capwapBaseAcSilentInterval,
      capwapBaseAcWaitDTLSTimer,
      capwapBaseAcWaitJoinTimer,
      capwapBaseAcEcnSupport
    }
    STATUS  current
    DESCRIPTION
        "Objects used for the CAPWAP protocol's parameters."
    ::= { capwapBaseGroups 14 }

capwapBaseStatsGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseFailedDTLSAuthFailureCount,
      capwapBaseFailedDTLSSessionCount
    }
    STATUS  current
    DESCRIPTION
        "Objects used for collecting the CAPWAP protocol's statistics."
    ::= { capwapBaseGroups 15 }

capwapBaseNotificationsGroup    NOTIFICATION-GROUP
    NOTIFICATIONS {
      capwapBaseChannelUp,
      capwapBaseChannelDown,
      capwapBaseDecryptErrorReport,
```

```
      capwapBaseJoinFailure,
      capwapBaseImageUpgradeFailure,
      capwapBaseConfigMsgError,
      capwapBaseRadioOperableStatus,
      capwapBaseAuthenFailure
    }
    STATUS  current
    DESCRIPTION
        "A collection of notifications in this MIB module."
    ::= { capwapBaseGroups 16 }

capwapBaseNotifyVarsGroup    OBJECT-GROUP
    OBJECTS {
      capwapBaseNtfWtpId,
      capwapBaseNtfRadioId,
      capwapBaseNtfChannelType,
      capwapBaseNtfAuthenMethod,
      capwapBaseNtfChannelDownReason,
      capwapBaseNtfStationIdList,
      capwapBaseNtfAuthenFailureReason,
      capwapBaseNtfRadioOperStatusFlag,
      capwapBaseNtfRadioStatusCause,
      capwapBaseNtfJoinFailureReason,
      capwapBaseNtfImageFailureReason,
      capwapBaseNtfConfigMsgErrorType,
      capwapBaseNtfMsgErrorElements
    }
    STATUS  current
    DESCRIPTION
        "Objects used for notifications."
    ::= { capwapBaseGroups 17 }

capwapBaseNotifyControlGroup OBJECT-GROUP
    OBJECTS {
      capwapBaseChannelUpDownNotifyEnable,
      capwapBaseDecryptErrorNotifyEnable,
      capwapBaseJoinFailureNotifyEnable,
      capwapBaseImageUpgradeFailureNotifyEnable,
      capwapBaseConfigMsgErrorNotifyEnable,
      capwapBaseRadioOperableStatusNotifyEnable,
      capwapBaseAuthenFailureNotifyEnable
   }
   STATUS  current
   DESCRIPTION
        "Objects used to enable or disable notifications."
   ::= { capwapBaseGroups 18 }

END
```

10. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects MAY be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The followings are the tables and objects and their sensitivity/vulnerability:

- Unauthorized changes to the capwapBaseWtProfileTable and writable objects under capwapBaseAcs group MAY disrupt allocation of resources in the network. For example, a WTP's static IP address could be changed by setting the capwapBaseWtpProfileWtpStaticIpAddress object.

- Unauthorized changes to writable objects under the capwapBaseAc group MAY disrupt allocation of resources in the network. For example, an invalid value for the capwapBaseWtpSessionsLimit object will increase the AC's traffic burden.

- Unauthorized changes to the capwapBaseMacAclTable MAY prevent legal stations from being able to access the network, while illegal stations are able to access it.

- Unauthorized changes to writable objects under the capwapBaseParameters group MAY influence CAPWAP protocol behavior and status. For example, an invalid value set for the capwapBaseAcDataCheckTimer MAY influence the CAPWAP state machine.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) MAY be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. The followings are the tables and objects and their sensitivity/vulnerability:

- The capwapBaseDataChannelDTLSPolicyOptions and capwapBaseControlChannelAuthenOptions under the capwapBaseAc group expose the current security option for CAPWAP data and control channels.

- The capwapBaseWtpTable exposes a WTP's important information like tunnel mode, MAC type, and so on.

- The capwapBaseWtpEventsStatsTable exposes a WTP's failure information.

- The capwapBaseRadioEventsStatsTable exposes a radio's failure information.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, the deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

11. IANA Considerations

11.1. IANA Considerations for CAPWAP-BASE-MIB Module

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
capwapBaseMIB	{ mib-2 196 }

11.2. IANA Considerations for ifType

IANA has assigned the following ifType:

Decimal	Name	Description
254	capwapWtpVirtualRadio	WTP Virtual Radio Interface

12. Contributors

This MIB module is based on contributions from Long Gao.

13. Acknowledgements

Thanks to David Harrington, Dan Romascanu, Abhijit Choudhury, Bert Wijnen, and David L. Black for helpful comments on this document and guiding some technical solutions.

The authors also thank the following friends and coworkers: Fei Fang, Xuebin Zhu, Hao Song, Yu Liu, Sachin Dutta, Ju Wang, Hao Wang, Yujin Zhao, Haitao Zhang, Xiansen Cai, and Xiaolan Wan.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2287] Krupczak, C. and J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

[RFC5415] Calhoun, P., Montemurro, M., and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, March 2009.

14.2. Informative References

[Err1832] RFC Errata, "Errata ID 1832", for RFC 5415, <http://www.rfc-editor.org>.

[IEEE.802-11.2007] "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Standard 802.11, 2007, <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC4118] Yang, L., Zerfos, P., and E. Sadot, "Architecture Taxonomy for Control and Provisioning of Wireless Access Points (CAPWAP)", RFC 4118, June 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC5416] Calhoun, P., Montemurro, M., and D. Stanley, "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11", RFC 5416, March 2009.

[RFC5834] Shi, Y., Ed., Perkins, D., Ed., Elliott, C., Ed., and Y. Zhang, Ed., "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding MIB for IEEE 802.11", RFC 5834, May 2010.

Authors' Addresses

Yang Shi (editor)
Hangzhou H3C Tech. Co., Ltd.
Beijing R&D Center of H3C, Digital Technology Plaza
NO. 9 Shangdi 9th Street, Haidian District
Beijing 100085
China

Phone: +86 010 82775276
EMail: rishyang@gmail.com

David T. Perkins (editor)
228 Bayview Dr.
San Carlos, CA 94070
USA

Phone: +1 408 394-8702
EMail: dperkins@dsperkins.com

Chris Elliott (editor)
1516 Kent St.
Durham, NC 27707
USA

Phone: +1 919-308-1216
EMail: chelliot@pobox.com

Yong Zhang (editor)
Fortinet, Inc.
1090 Kifer Road
Sunnyvale, CA 94086
USA

EMail: yzhang@fortinet.com