Aruba Mobility Controller and Access Point Series Security Target

Version 1.1
June 11, 2012

Prepared for:

Aruba Networks, Inc.

1344 Crossman Avenue
Sunnyvale, CA 94089-1113

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

6841 Benjamin Franklin Drive
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is a Wireless Local Area Network (WLAN) access system comprising Aruba Mobility Controllers, Access Points, and the ArubaOS. The Aruba Mobility Controllers are wireless switch appliances that provide a wide range of wireless and wired network mobility, security, centralized management, auditing, authentication, and remote access. The Aruba Access Point appliances service wireless clients[1] and can monitor radio frequency spectrums to detect intrusions, denial of service (DoS) attacks, and other vulnerabilities. The ArubaOS is a suite of mobility applications that runs on all Aruba controllers and APs and allows administrators to configure and manage the wireless and mobile user environment.

The Security Target contains the following additional sections:

- TOE Description (Section 2)

  This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.

- Security (Section 3)

  This section details the expectations of the environment, including the assumptions, organizational security policies, and threats that are countered by the TOE and TOE environment.

- Security Objectives (Section 4)

  This section details the security objectives of the TOE and TOE environment.

- IT Security Requirements (Section 5)

  The section presents the security functional requirements (SFRs) for the TOE, and details the assurance requirements for EAL4 augmented with ALC_FLR.2.

- TOE Summary Specification (Section 6)

  The section describes the security functions represented in the TOE that satisfy the security requirements.

1.1 RFC Conformance Statements

This section identifies, for the critical RFCs, the options supported by the TOE.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
EAP-TLS	RFC 2716	**TLS Version:** The TOE offers TLS v1.0 for the EAP TLS Authentication Protocol. **Conversation Restarts:** The TOE implements a restart capability and imposes a limit on the number of restarts. The TOE blacklists a client after a configurable number of authentication failures. **Identity Verification:** The TOE verifies that a claimed identity corresponds to the certificate presented by the client. The TOE uses the CN in the client certificate as the userID to authenticate.
EAP-TTLS	RFC 5281	The TOE does not implement an EAP-TTLS authentication server—it simply acts as a pass thru authenticator for all EAP-TTLS traffic between the supplicant and the authentication server in the operational environment.
HTTP over TLS	RFC 2818	**Connection Closure:** the TOE always initiates a close alert before closing a connection. The TOE does not perform an incomplete close and waits for the peer's closure alert.

[1] The wireless client is part of the IT environment.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
IPsec/IKE	RFC 2408/ RFC 2409	**General Message Processing:** The TOE ensures successive retransmissions of the same packet are separated by increasingly longer time intervals. **ISAKMP Header Format:** The TOE will not accept ISAKMP packets with a version number (comprising Major.Minor) in advance of its own. The TOE sets the encryption flag of ISAKMP header to 1in an outbound ISAKMP packet. **Security Association Establishment:** The TOE retains the Transform Number field in the Transform Payload in an outbound ISAKMP packet. **Security Association Modification:** The TOE will continue to support incoming traffic on an old SA until traffic is received on a newly created SA.
IPsec/ESP	RFC 4303	**Modes:** The TOE supports both transport and tunnel modes for ESP. **Services:** The TOE does not support confidentiality-only or integrity-only services—only the 'confidentiality and integrity' service is supported, in order to comply with FIPS requirements.
RADIUS	RFC 2138	The TOE supports challenge/response, and sends a new Access-Request in response to a valid Access-Challenge.
	RFC 2869	**EAP-Message:** The TOE silently discards Access-* messages without a Message-Authenticator attribute.
	RFC 3579	**Conflicting Messages:** The TOE sends EAP-success or EAP-failure to the client depending on if it receives Access-Accept or Access-Reject from the RADIUS server respectively. The Access-Accept packets have only one EAP-Message attribute in them, containing EAP Success; and Access-Reject packets have only one EAP-Message attribute in them, containing EAP Failure. **EAP-Message:** The TOE silently discards Access-* messages without a Message-Authenticator attribute. **Security Protocol/Security Issues:** The TOE supports IPSec requirements specified in the appropriate RFCs. It supports IPSec ESP to protect RADIUS frames between the TOE and RADIUS server with specified algorithms and IKE for key negotiation.
	RFC 3580	The TOE supports IPSec requirements specified in the appropriate RFCs. It supports IPSec ESP to protect RADIUS frames between the TOE and RADIUS server with specified algorithms and IKE for key negotiation.
SSH	RFC 4251	**Host Keys:** The TOE has one RSA and one DSA Host Key for SSH v2, which are generated on initial setup of the TOE. These keys are not shared with any other host and are unique to each TOE instance. The TOE presents the client with its host key fingerprint when the client is connecting to the TOE for the first time. When a client connects to the TOE for the first time, the TOE prompts the user to accept or deny the TOE's host key and hence connect successfully or disconnect. **Policy Issues:** The TOE has an uneditable policy specifying its supported encryption, integrity, and compression algorithms, as listed in section 6.1.2. It implements all mandatory algorithms and methods. The TOE can be configured to accept public key based authentication and/or password based authentication per admin user. The TOE does not allow port forwarding and sessions to clients. The TOE has no X11 libraries or applications and does not support X11 forwarding. **Confidentiality:** The TOE does not accept the "none" cipher.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
		Data Integrity: The TOE does not accept the "none" MAC. Note the TOE does not have RekeyLimit set in its configuration and hence does not initiate rekeying. However, client initiated rekeying is processed. **Denial of Service:** Once the SSH connection is terminated, the TOE does not pro-actively try to re-establish it. The TOE can be configured with ACLs to control the clients that are able to connect to it via SSH. **Ordering of Key Exchange Methods:** The TOE orders key exchange algorithms by strength. **Debug Messages:** The TOE requires debug messages to be turned on explicitly and the audit trail reflects the same.
SSH	RFC 4251	**End Point Security:** The TOE does not permit port forwarding. SSH is a CLI mechanism for TOE administrators and hence the TOE places no restrictions on user actions once authenticated. **Proxy Forwarding:** The TOE does not support proxy forwarding. **X11 Forwarding:** The TOE does not support X11 forwarding.
	RFC 4252	**Authentication Protocol:** The TOE does not accept "none" authentication method. The TOE disconnects a client after 30 seconds if authentication has not been completed. The TOE also allows authentication retries of 6 times before sending a disconnect to the client. **Authentication Requests:** The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed. The TOE just sends back a disconnect as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies "none" authentication method and replies with a list of permitted authentication methods. **Public Key Authentication Method:** The TOE allows "publickey" authentication. Authentication succeeds if the correct private key is used. Note the TOE does not require multiple authentications. **Password Authentication Method:** The TOE supports password authentication. The TOE does not allow an expired password to be used for authentication. Note the TOE does not support changing user passwords in the SSH session. **Host-Based Authentication:** The TOE does not support host-based authentication.
	RFC 4253	**Encryption:** The TOE offers only aes128-cbc,3des-cbc,aes192-cbc and aes256-cbc for encryption of SSH sessions. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the "none" algorithm for encryption. **Data Integrity:** The TOE permits negotiation of MAC algorithms in each direction. **Key Re-Exchange:** The TOE performs a re-exchange when SSH_MSG_KEXINIT is received. However, it does not initiate a key re-exchange itself.

- Protection Profile Claims (Section 0)

  This section presents any protection profile claims.

- Rationale (Section 8)

This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness, and suitability.

1.2 Security Target, TOE and CC Identification

ST Title – Aruba Mobility Controller and Access Point Series Security Target

ST Version – Version 1.1

ST Date – June 11, 2012

TOE Identification – Aruba Mobility Controller and Access Point Series, Version 3.4.4.0. Please see table below.

Product	Part Number(s)	Required Software Licenses	Firmware Version
Aruba 6000 Mobility Controller (FIPS)	• 6000-400-F1 • 6000-400-USF1 • 6000-400-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
Aruba 3200 Mobility Controller (FIPS)	• 3200-F1 • 3200-USF1 • 3200-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
Aruba 3400 Mobility Controller (FIPS)	• 3400-F1 • 3400-USF1 • 3400-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
Aruba 3600 Mobility Controller (FIPS)	• 3600-F1 • 3600-USF1 • 3600-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
Aruba 800 Mobility Controller (FIPS)	• 800-F1 • 800-USF1 • 800-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
Aruba 200 Mobility Controller (FIPS)	• 200-F1 • 200-USF1 • 200-ILF1	• Policy Enforcement Firewall • Wireless Intrusion Prevention • VPN Server	3.4.4.0-FIPS operated in FIPS mode
AP-60 Access Point	• AP-60-F1	N/A	3.4.4.0-FIPS operated in FIPS mode
AP-61 Access Point	• AP-61-F1	N/A	3.4.4.0-FIPS operated in FIPS mode

Product	Part Number(s)	Required Software Licenses	Firmware Version
AP-65 Access Point	• AP-65-F1	N/A	3.4.4.0-FIPS operated in FIPS mode
AP-70 Access Point	• AP-70-F1	N/A	3.4.4.0-FIPS operated in FIPS mode
AP-85 Access Point	• AP-85TX-F1 • AP-85FX-F1 • AP-85FX-EU-F1 • AP-85LX-F1 • AP-85LX-EU-F1	N/A	3.4.4.0-FIPS operated in FIPS mode
AP-124 Access Point	• AP-124-F1	N/A	3.4.4.0-FIPS operated in FIPS mode
AP-125 Access Point	• AP-125-F1	N/A	3.4.4.0-FIPS operated in FIPS mode

TOE Developer – Aruba Networks, Inc.

Evaluation Sponsor – Aruba Networks, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007

1.3 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 2, September 2007.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 2, September 2007.
  - Part 3 Conformant
  - EAL 4 Augmented
- Conformant to US Government Wireless Local Area Network (WLAN) Access System For Basic Robustness Environments Protection Profile, Version 1.1, 25 July 2007, National Security Agency

1.4 Conventions and Acronyms

This section specifies the formatting information used in the Security Target.

1.4.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1(1)

and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).

- Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details by ST author. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that refinements made by the PP author have not been indicated in the ST
- Extended Requirements are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the ST needs. To ensure these requirements are explicitly identified, the ending "_(EXT)" is appended to the newly created short name and the component.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4.2 Acronyms

AAA	Authentication, Authorization, and Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
AP	Access Point
CC	Common Criteria
CLI	Command Line Interface
CP	Control Plane
DP	Data Plane
DoS	Denial of Service
EAP	Extensible Authentication Protocol
FP	Fast Path
FPGA	Field Programmable Gate Array
FIPS	Federal Information Processing Standard
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
IKE	Internet Key Exchange
IPSec	Internet Protocol Security
LDAP	Lightweight Directory Access Protocol
MAC	Medium Access Control
MC	Mobility Controller
HMAC-MD5	Hashed Message Authentication Code – Message Digest 5

NAT	Network Address Translation
NTP	Network Time Protocol
PAPI	Programming Application Program Interface
PP	Protection Profile
RADIUS	Remote Authentication Dial In User Service
RF	Radio Frequency
RNG	Random Number Generator
SOS	SiByte Operating System
SP	Slow Path
SSH	Secure Shell
TACACS+	Terminal Access Controller Access-Control System +
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WIDS	Wireless Intrusion Detection System
WIP	Wireless Intrusion Protection
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

2. TOE Description

The Target of Evaluation (TOE) consists of Aruba Mobility Controller and AP Series appliances, running ArubaOS v3.4.4.0.

The TOE is a Wireless Local Area Network (WLAN) access system comprising Aruba Mobility Controllers, Access Points, and the ArubaOS. The WLAN PP defines this technology type as "one or more components that provide secure wireless access to a wired or wireless network". The Aruba Mobility Controllers are wireless switch appliances that provide a wide range of security services and features including wireless and wired network mobility, security, centralized management, auditing, authentication, and remote access. The Aruba Access Point appliances service wireless clients[2] and can monitor radio frequency spectrums to detect intrusions, denial of service (DoS) attacks, and other vulnerabilities. The ArubaOS is a suite of mobility applications that runs on all Aruba controllers and APs, and allows administrators to configure and manage the wireless and mobile user environment. Figure 1 shows an example of a WLAN Access System environment configuration[3]. Figure 2 shows an example of a WLAN Access System configuration. This configuration includes one AP and one MC. This should not be misconstrued as the only configuration as multiple MCs and APs can comprise the TOE. However, this is the minimum configuration required in the CC mode. The rest of this section will describe, at a high-level, an overview of the TOE architecture, define the scope of evaluation and the physical boundary of the TOE, and summarize the security functionality provided by the TOE.

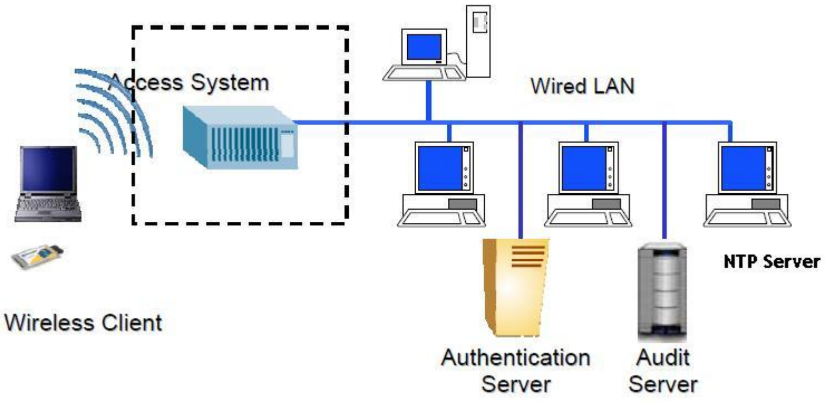

Figure 1: Example of WLAN Access System Environment

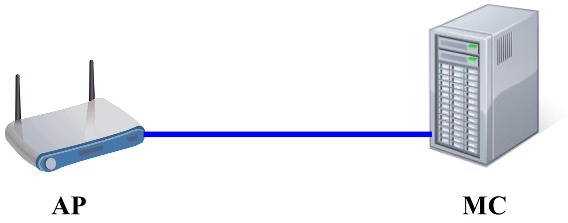

Figure 2: Example of WLAN Access System

The AP is connected to the Controller via wired Ethernet Local Area Network (LAN) or wired directly to the Controller. The connection is protected using HMAC-MD5, which is a non-FIPS approved algorithm. Therefore, no protection should be assumed other than the physical security assumption stated by A.PHYSICAL (for example, the network switches and TOE devices are protected in a secure room). The AP and MC use GRE as the tunneling protocol to encapsulate IEEE 802.11 traffic (data from wireless clients) over the IP wired network.

[2] Wireless client is not part of the TOE

[3] Other wireless configurations may exist and still meet requirements identified in the PP. In all cases, wireless traffic must be able to pass to the wired network via the wireless access system providing the necessary security.

In an encrypted WLAN, a wireless client first associates with an AP and then authenticates (IEEE 802.11i[4] or IPSec/IKE) using credentials to obtain access to the network (an IP address) and establish a session with the TOE. The authenticated wireless client is then assigned a role based on the configuration in the Mobility Controller. The ACLs defined for that role control traffic flows to and from the client.

The TOE's captive portal capability allows a wireless user to authenticate using a web-based portal. Captive portals are typically used in public access wireless hotspots or for hotel in-room Internet access. Using captive portal, a wireless client associates to the AP and establishes a session and IP address without authentication. The wireless client is assigned to a restricted role (by default but this is configurable) which permits the client to perform HTTPS authentication. When the wireless client launches a web browser, the client gets redirected to the TOE's captive portal login page over TLS. The wireless client authenticates using the right credentials and gets assigned a new role. The ACLs defined for that role control traffic flows to and from the client.

Each authenticated wireless client can also be placed into a VLAN. While all authenticated wireless clients can be placed into a single VLAN, the TOE (Mobility Controller) allows administrators to group wireless clients into separate VLANs. This enables separation and isolation of groups of wireless clients and their access to network resources. For example, administrators can place authorized employee clients into one VLAN and temporal clients, such as contractors or guests, into a separate VLAN.

2.1 TOE Overview

The TOE consists of the following components:

- Aruba Mobility Controllers
- Aruba Access Points
- ArubaOS.

In the CC evaluated configuration, the TOE (all components that make up the WLAN access system—at a minimum, one Controller and one AP) must be configured to operate in the FIPS 140-2 Approved mode of operation. In FIPS-Approved mode, various weak protocols and algorithms are disabled. Please reference the appropriate FIPS 140-2 Security Policy documents for each controller and access point for more details at http://csrc.nist.gov/groups/STM/cmvp/index.html.

2.2 TOE Architecture

At a high level, Aruba Mobility Controllers consist of two main components:

- Control Plane (CP)—implements functions which can be handled at lower speeds such as Mobility Controller system management (CLI and Web GUI), user authentication (e.g. 802.1X, RADIUS, LDAP, captive portal), Internet Key Exchange (IKE), auditing/logging (syslog), Wireless IDS (WIDS), and termination of protocols operating at the system level (e.g. SSH, TLS, NTP, etc.).
- Data Plane (DP)—implements functions that must be handled at high speeds such as high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging), termination of 802.11 associations/sessions, tunnel termination (GRE, IPSec), firewall/ACL and deep packet inspection functions, and cryptographic acceleration.

The CP runs the Linux OS, along with various custom user-space applications which provide CP functions. This suite of custom user-space applications, which constitutes ArubaOS, provides the following functions:

- Monitors and manages critical system resources, including processes, memory, and flash
- Sends and receives PAPI[5] protocol messages to and from managed APs as well as other mobility controllers
- Manages system configuration and licensing

[4] Implements 802.1X for wireless access points to address the security vulnerabilities found in WEP.
[5] PAPI is an Aruba-proprietary WLAN management protocol and provides no direct security.

- Manages an internal database used to store licenses, user authentication information, etc
- Provides network anomaly detection, hardware monitoring, mobility management, wireless management, and radio frequency management services
- Provides a Command Line Interface (CLI)
- Provides a web-based management UI for the mobility controller, as well as an optional captive portal login page for WLAN users
- Provides various WLAN station and AP management functions
- Provides authentication services for the system management interfaces (CLI, web GUI) as well as for WLAN users
- Provides IPSec key management services for remote APs, VPN users, and connections with other Aruba mobility controllers
- Provides network time protocol service for APs, point to point tunneling protocol services for users, layer 2 tunneling protocol services for users, file transfer protocol services for users, serial over Ethernet connection services for administration of directly-connected APs, SSH services for incoming management connections, SNMP client/agent services, and protocol independent multicast (routing) services for the controller
- Provides syslog services by sending logs to the operating environment.

The DP is further subdivided into two subcomponents: Fast Path (FP) and Slow[6] Path (SP). The FP implements high-speed packet forwarding based on various proprietary tables and sends the packets to SP. The SP manages (create, delete, and age entries) all DP tables such as user, ACL, station, tunnel, route, ARP cache, session, bridge, VLAN[7], and port. The SP also performs deep packet inspection and cryptographic processing.

The Linux OS running on the CP is MontaVista's Embedded Linux. Linux is a soft real-time, multi-threaded operating system that supports memory protection between processes. Only Aruba provided interfaces are used, and the CLI is a restricted command set. For the 3000 series, this OS implements both the control and data planes.

For all but the 3000 series, the data plane is primarily implemented on the SiByte[8]. There is a lightweight, Aruba-proprietary OS running on the SiByte, called SOS. SOS contains an Ethernet driver, a serial driver, a logging facility, semaphore support, a crypto driver. This OS is not general purpose operating system. In the MC 6000 controller, a FPGA is used to control and monitor the switch fabric, MACs, PHYs, SoE, and PoE and provide security functionality such as filtering.

The DP and CP run on different hardware platforms but the security functionality remains the same, regardless of the model. The differences in the platforms are in the processors, memory capacity, physical interfaces, FPGA implementation, etc., and are based on performance and scalability requirements. The table below shows the different models based on maximum number of APs and users supported.

Product	Max. # of APs	Max. # of Users	Typical Deployment
MC-6000	2,048	32,768	Headquarters/ Large Campus
MC-3000 Series	128	2,048	Medium/Large Enterprise/Campus
MC-800	16	256	Branch Office
MC-200	6	100	Small Offices/ Retail Store

[6] The entire DP (including both FP and SP elements) is a high-speed packet processor, so the SP designation should be understood to be relative.

[7] A VLAN has the same attributes as a physical LAN, but it allows for end devices to be grouped together even if they are not located on the same network switch. Network reconfiguration can be done through the Aruba software instead of physically relocating devices.

[8] SiByte refers to the hardware architecture of the TOE which includes Broadcom SiByte MIPS processors and cryptographic processor.

The Aruba AP is a wireless hardware device that is enclosed in a plastic encasing. The ArubaOS provides the security functionality for the APs. The ArubaOS runs on an embedded Linux kernel. Similar to the controllers, the security functionality of the different models is the same with differences in platforms based on performance and scalability requirements only. At a high level, Aruba Access Points consist of the following subsystems:

- Processor subsystem—performs the packet processing functions on the packet.
- Memory subsystem—contains memory which supports the Processor subsystem.
- Ethernet Controller (i.e., Network Interface Controller) subsystem—includes integrated Ethernet Media Access Control (MAC) for transfer of 10/100 Ethernet packets between the AP and the wired network.
- Radio Controller subsystem—there are two radio controllers, 802.11a (5 GHz range) and 802.11b/g (2.4 GHz range).
- Wireless Antenna subsystem—interface between the wireless world and the AP. The antenna handles both 5 GHz and 2.4 GHz ranges.
- PoE (Power over Ethernet) subsystem—receives 48V power over the Ethernet.
- SoE (Serial over Ethernet) subsystem—contains the RS-232 serial port circuitry for communicating with the switch connected to the 10/100 port.
- USB subsystem—the AP-70 supports one USB V2.0 compliant port (up to 480 Mbps). A PCI to USB 2.0 controller is used to interface to the system host.
- Serial subsystem—the AP-120 supports a serial console port on the front panel that utilizes a RJ45 jack and connects directly to serial port 0 via the RS232 transceiver.

2.2.1 Physical Boundaries

The TOE consists of the following components:

- Aruba Mobility Controllers: Aruba MC 200, MC 800, MC 3000 Series (comprising MC 3200, MC 3400, and MC 3600), and MC 6000.
- Aruba Access Points: Aruba AP 60, AP-61, AP-65, AP-70, AP-85, and AP-120 Series (comprising AP-120, AP-121, AP-124 and AP-125).
- ArubaOS version 3.4.4.0

The differences in the models include the number of ports, interfaces, throughput and processing speed, memory and storage. Although these models have different specifications (in terms of performance and capabilities), they all provide the same security functions described in the ST; therefore, they have been considered to be the same for the purposes of the ST description. There is no difference between the products and the TOE. Since the TOE is a WLAN access system, the physical boundary of each product that comprises the WLAN is the hard steel or plastic encasing.

The ArubaOS consists of a base software package with add-on software modules that can be activated by installing the appropriate license key. Three SFR-enforcing software modules are required to be licensed and installed in the CC evaluated configuration. The other software modules are optional and are described in Appendix A for completeness only. The base ArubaOS software includes the following functions:

- Centralized configuration and management of APs
- Wireless client authentication to an external authentication server or to the controller's internal database
- Encryption
- Mobility with fast roaming
- RF management and analysis tools.

The following table summarizes the required software modules.

Required Software Module	Description
Policy Enforcement Firewall	Provides identity-based security for wired and wireless clients. Stateful firewall enables classification based on client identity, device type, location, and time of day, and provides differentiated access for different classes of users.
Wireless Intrusion Protection	Detects, classifies and limits designated wireless security threats such as rogue APs, DoS attacks, malicious wireless attacks, impersonations, and unauthorized intrusions. Eliminates need for separate system of RF sensors and security appliances.
VPN Server	Enables controllers to provide Virtual Private Networks (VPN) tunnel termination to local and remote clients. Provides site-to-site VPN tunnels between controllers and third-party VPN concentrators.

The wireless client can be any laptop that uses a wireless network card that is Wi-Fi Certified. Specifically, it must be WPA/WPA2 compliant to support the cryptographic and authentication (e.g., certificate) requirements of the TOE. Note that WPA/WPA2 compliant is a specific subset of the Wi-Fi certification. For more information, please see http://certifications.wi-fi.org/wbcs_certified_products.php?lang=en

The TOE relies on third-party software and hardware components in the operating environment. The TOE can utilize an external audit server (support syslog) to store audit records and external authentication server (support RADIUS, LDAP, TACACS+) to authenticate users. In addition, the TOE uses an external Time server (support NTP) to obtain reliable time stamps and external SNMP server to capture SNMP traps. For security reasons, only SNMPv3 is allowed in the evaluated configuration. The remote administrator can use a web browser to access the Web GUI interface and/or use SSH client to access the CLI. The local administrator can use the serial port to access the CLI. Neither the web browser or SSH client is part of the TOE. Note that Telnet cannot be used to access the CLI in the CC evaluated configuration.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE, comprising:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- Resource Utilization
- TOE access.
- Trusted Paths/Channels

The TOE protects itself from tampering and bypass through several mechanisms implemented by the TOE and the operating environment. The underlying operating system separates processes into separate domains and prevents one process from accessing memory space of another process. The operating system is non-modifiable and the interfaces are strictly limited. The TOE relies on physical security to protect data transmitted between the TOE components from unauthorized modification. Remote administration used by administrators to manage the TOE is secured through TLS (Web GUI) or SSH (CLI). All administrators must be identified and authenticated either by the TOE or an external authentication server. Inactive sessions are terminated after an administrator-specified time period. In addition to authentication, the TOE also verifies that users are authorized to perform management functions based on their roles. The external NTP server provides the reliable time source and external syslog server stores and protects the audit trail from tampering.

Version 3.4.4.0 of the TOE includes some enhancements relative to the previously evaluated 3.4.2.3 version. None of those enhancements have been subject to evaluation and as such should either not be used ot should be used with the understanding the understanding they have not been subject to analysis and testing in the context of a Common Criteria evaluation. Of those features the following are security related:

Kerberos authentication support – This enhancement involves new configuration commands to enable and configure a Kerberos profile. Once enabled, the product is able to 'sniff' Kerberos sessions and then use an external server for role derivation. Since this feature allows authentication of users via other than evaluated means, this feature should generally not be used in the evaluated configuration.

Management password policy enhancement – This enhancement implements password length and composition (specific number of letters, numbers, and special character and maximum number of repeating characters) requirements as well as a maximum number of failed login attempts before being locked out for a period. While this feature has not been evaluated, its use should not impact any claims made in this Security Target.

NAT-T changes – This enhancement changes the behavior of NAT to support MAC clients via NAT-T operating on specific ports (UDP500 and UDP4500). This enhancement enables MAC clients to connect using IPSec VPN tunnels and as such is security related. While this feature has not been evaluated, its use should not impact any claims made in this Security Target though it should not be assumned that MAC client IPSec connections have been subject to evaluation as has been done for toher supported clients.

The sections below summarize the security functions provided by the TOE.

2.2.2.1 Security Audit

The TOE is capable of auditing security relevant events such as logins, administrator actions, use of trusted channel and path, cryptographic operations, resource limitation exceeded, etc. Each audit event includes the date and time of the event, the type of event, the subject identity (if applicable), and the outcome of the event. The administrator can include and exclude events to be audited based on specific criteria. The TOE can detect security event/violation based on signature and perform the appropriate action such as alerting the administrator or denying access.

The TOE relies on the NTP server to provide a reliable timestamp and syslog server to store and protect the audit trail. The administrator is provided an interface in the operating environment to read audit logs and that interface is restricted.

2.2.2.2 Cryptographic Support

The TOE has been certified as a FIPS 140-2 cryptographic module (FIPS 140-2 certified Certificates #:1297, #1116, #1109, #1077, and #1075). The FIPS overall level is 2.The logical interfaces used for the input and output of plaintext cryptographic key components, authentication data, and CSPs are logically separated from all other interfaces using a trusted path where "trusted path" is interpreted to include a communications channel established using a FIPS 140-2 Level 2 cryptographic module and the HTTPS protocol between the cryptomodule and the external IT entity. The cryptographic module only employs FIPS-Approved RNG, key generation, establishment, zeroization, encryption, digital signature, and hashing algorithms as specified by the FCS requirements.

2.2.2.3 User Data Protection

The TOE provides both policies and ACLs to control information flow. Firewall policies identify specific characteristics about a data packet and specify the action to take based on that identification. Access control lists (ACLs) provide a method of restricting certain types of traffic on a physical port based on IP address, port, protocol, etc. Administrators can apply firewall policies to user roles and give differential treatment to different users on the same network, or to physical ports and apply the same policy to all traffic through the port. The TOE can also group wireless clients into separate VLANs. The TOE protects the data between itself and the wireless clients using AES or TDES but encryption can be enabled or disabled by administrator. In the evaluated configuration, the administrator must not disable encryption. The TOE ensures any previous information content is made unavailable upon the allocation of a memory buffer to a network packet.

2.2.2.4 Identification and Authentication

The TOE can maintain administrator and user attributes, including credentials such as username and password for administrators and session key and role for remote authenticated users (username and password are stored in the internal database or authentication server). The TOE requires identification and authentication (either locally or remotely through external authentication server, internally, or both) of administrators managing the TOE. Wireless clients are identified and authenticated by different authentication mechanisms such as Captive portal, 802.1X, etc. More detailed information is provided in section 6.1.4. After an administrator-specified number of failed attempts, the user account is locked out. In addition, the password mechanism can be configured to have a minimum length of six characters.

2.2.2.5 Security Management

The TOE provides the capability to manage auditing, cryptographic operations, intrusion protection functions, password minimum length enforcement, user accounts, policies & ACLs rules, advisory banner, and timeout (inactivity threshold) value. The management functions are restricted to an administrator role. The role must have the appropriate access privileges or access will be denied. The wireless user role has no access to the management interfaces. The information flow policy is denied by default and only administrators can change the default values. The FIPS-certified TOE ensures that only secure values are accepted for security attributes.

2.2.2.6 Protection of the TSF

The TOE provides integrity protection for all communication between its components. This prevents unauthorized modification of TSF data during transmission. The TOE also provides self-tests to ensure the correct operation of the cryptographic functions and TSF hardware. There is an option for the administrator to verify the integrity of stored TSF executable code. The communication between the TOE and another trusted IT product (e.g., NTP, syslog, RADIUS) is protected through a trusted channel. The communication between the TOE and remote administrator is protected through a trusted path.

2.2.2.7 Resource Utilization

The TOE can enforce maximum usage quotas on the number of concurrent sessions available to a defined group of users (role).

2.2.2.8 TOE Access

The TOE allows administrators to configure a period of inactivity for a user’s session. Once that time period has been reached while the session has no activity, the session is terminated. A warning banner is displayed at the management interfaces (Web GUI and CLI) to advise users on appropriate use and penalty for misuse of system.

2.2.2.9 Trusted Path/Channels

The TOE provides an encrypted channel between itself and third-party trusted IT entities in the operating environment. The TOE also provides a protected communication path between itself and wireless users.

2.3 TOE Documentation

Aruba Networks offers a series of documents that describe the installation and configuration of Mobility Controllers and Access Points as well as guidance for subsequent use and administration of the applicable security features. The documentation is available online for registered users at http://www.arubanetworks.com/support.php.

3. Security Problem Definition

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed. The statement of the TOE security environment defines the following:

- Organizational Policies that the TOE and the environment of the TOE fulfill
- Threats that the TOE and the environment of the TOE counter
- Assumptions made about the operational environment and the intended method of use for the TOE

Furthermore, the TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL 4 augmented with ALC_FLR.2 as defined in the CC.

3.1 Organizational Policies

P.ACCESS_BANNER The TOE shall display an initial banner for administrator logins describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.

P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.

P.CRYPTOGRAPHY The TOE shall provide cryptographic functions for its own use, including encryption/decryption operations.

P.CRYPTOGRAPHY_VALIDATED Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).

P.ENCRYPTED_CHANNEL The TOE shall provide the capability to encrypt/decrypt wireless network traffic between the TOE and those wireless clients that are authorized to join the network.

P.NO_AD_HOC_NETWORKS In accordance with the DOD Wireless Policy, there will be no ad hoc 802.11 or 802.15 networks allowed.

3.2 Threats

T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

T.ACCIDENTAL_CRYPTO_COMPROMISE A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.

T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

T.POOR_DESIGN Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.

T.POOR_IMPLEMENTATION Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.

T.POOR_TEST The developer or tester performs insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may occur, resulting in incorrect TOE behavior being undiscovered leading to flaws that may be exploited by a mischievous user or program.

T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

T.TSF_COMPROMISE A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).

T.UNATTENDED_SESSION A user may gain unauthorized access to an unattended session.

T.UNAUTHORIZED_ACCESS A user may gain access to services (either on the TOE or by sending data through the TOE) for which they are not authorized according to the TOE security policy.

T.UNAUTH_ADMIN_ACCESS An unauthorized user or process may gain access to an administrative account.

3.3 Assumptions

A.NO_EVIL Administrators are non-hostile, appropriately trained and follow all administrator guidance.

A.NO_GENERAL_PURPOSE There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

A.TOE_NO_BYPASS Wireless clients are configured so that information cannot flow between a wireless client and any other wireless client or host networked to the TOE without passing through the TOE.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

4.1 Security Objectives for the TOE

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security-relevant events associated with users.

O.CORRECT_TSF_OPERATION

The TOE will provide the capability to verify the correct operation of the TSF.

O.CRYPTOGRAPHY

The TOE shall provide cryptographic functions to maintain the confidentiality and allow for detection of modification of user data that is transmitted between physically separated portions of the TOE, or outside of the TOE.

O.CRYPTOGRAPHY_VALIDATED

The TOE will use NIST FIPS 140-1/2 validated cryptomodules for cryptographic services implementing NIST-approved security functions and random number generation services used by cryptographic functions.

O.DISPLAY_BANNER

The TOE will display an advisory warning prior to establishing an administrator session regarding use of the TOE prior to permitting the use of any TOE services that requires authentication.

O.MANAGE

The TOE will provide functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

O.MEDIATE

The TOE must mediate the flow of information to and from wireless clients communicating via the TOE in accordance with its security policy.

O.RESIDUAL_INFORMATION

The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.

O.SELF_PROTECTION

The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.

O.TIME_STAMPS

The TOE shall obtain reliable time stamps.

O.TOE_ACCESS

The TOE will provide mechanisms that control a user's logical access to the TOE.

O.INTRUSION

The TOE will detect wireless intrusion attacks, alert administrators and, where possible, prevent or contain the intrusion attempts.

O.ADMIN_GUIDANCE

The TOE will provide administrators with the necessary information for secure management.

O.CONFIGURATION_
IDENTIFICATION

The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.

O.DOCUMENTED_ DESIGN

The design of the TOE is adequately and accurately documented.

O.PARTIAL_
FUNCTIONAL_TESTING

The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.

O.VULNERABILITY_ ANALYSIS The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Environment

OE.AUDIT_PROTECTION The IT Environment will provide the capability to protect audit information and the authentication credentials.

OE.AUDIT_REVIEW The IT Environment will provide the capability to selectively view audit information.

OE.MANAGE The TOE IT environment will augment the TOE functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

OE.NO_EVIL Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.

OE.NO_GENERAL_PURPOSE There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

OE.PHYSICAL The environment provides physical security commensurate with the value of the TOE and the data it contains.

OE.PROTECT_MGMT_COMMS The environment shall protect the transport of audit records to the audit server, remote network management, and authentication server communications with the TOE and time service in a manner that is commensurate with the risks posed to the network.

OE.RESIDUAL_INFORMATION The TOE IT environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.

OE.SELF_PROTECTION The environment will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.

OE.TIME_STAMPS The TOE IT environment shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.

OE.TOE_ACCESS The environment will provide mechanisms that support the TOE in providing a user's logical access to the TOE.

OE.TOE_NO_BYPASS Wireless clients are configured so that information cannot flow between a wireless client and any other wireless client or host networked to the TOE without passing through the TOE.

5. IT Security Requirements

Most of the security requirements for the TOE have been drawn from Parts 2 and 3 of the Common Criteria. A few security requirements have been extended as specified in the PP. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE and as required by the PP, while the assurance requirements have been selected to offer a moderate to high degree of assurance that those security functions are properly realized.

5.1 Extended Components Definition

There are extended security requirements defined within this Security Target. But all extended security requirements were taken from the PP verbatim. There are no additional extended security requirements that were not defined in the PP.

5.2 TOE Security Functional Requirements

The following table describes the SFRs that are satisfied by Aruba Mobility Controller and Access Point Series.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_ARP.1: Security Alarms
	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User Identity Association
	FAU_SAA.3: Simple Attack Heuristics
	FAU_SEL.1: Selective Audit
FCS: Cryptographic Support	FCS_BCM_(EXT).1: Extended – Baseline Cryptographic Module
	FCS_CKM.1(1): Symmetric Cryptographic Key Generation
	FCS_CKM.1(2): Asymmetric Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Distribution
	FCS_CKM_(EXT).2: Extended – Cryptographic Key Handling and Storage
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (Data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (Digital Signature)
	FCS_COP.1(3): Cryptographic Operation (Hashing)
	FCS_COP.1(4): Cryptographic Operation (Key Agreement)
	FCS_COP_(EXT).1: Extended – Random Number Generation
FDP: User Data Protection	FDP_IFC.1(1): Subset Information Flow Control
	FDP_IFF.1(1): Simple Security Attributes
	FDP_IFC.1(2): Subset Information Flow Control
	FDP_IFF.1(2): Simple Security Attributes
	FDP_PUD_(EXT).1: Extended – Protection of User Data
	FDP_RIP.1: Subset Residual Information Protection

Requirement Class	Requirement Component
FIA: Identification and Authentication	FIA_AFL.1: Authentication Failure Handling
	FIA_ATD.1(1): Administrator Attribute Definition
	FIA_ATD.1(2): User Attribute Definition
	FIA_SOS.1: Verification of Secrets
	FIA_UAU.1: Timing of Authentication
	FIA_UAU_(EXT).5: Extended – Multiple Authentication Mechanisms
	FIA_UID.2: User Identification Before Any Action
	FIA_USB.1: User-subject Binding
FMT: Security Management	FMT_MOF.1(1): Management of Cryptographic Security Functions Behavior
	FMT_MOF.1(2): Management of Audit Security Functions Behavior
	FMT_MOF.1(3): Management of Authentication Security Functions Behavior
	FMT_MOF.1(4): Management of Wireless Intrusion Protection Security Functions Behavior
	FMT_MSA.1(1): Management of Security Attributes
	FMT_MSA.1(2): Management of Security Attributes
	FMT_MSA.2: Secure Security Attributes
	FMT_MSA.3(1): Static Attribute Initialization
	FMT_MSA.3(2): Static Attribute Initialization
	FMT_MTD.1(1): Management of Audit Data
	FMT_MTD.1(2): Management of Authentication Data (Administrator)
	FMT_MTD.1(3): Management of Authentication Data (User)
	FMT_MTD.1(4): Management of TSF Data (Policy/ACL Rulesets)
	FMT_MTD.1(5): Management of TSF Data (Session Quota)
	FMT_SMF.1(1): Specification of Management Functions (Cryptographic Functions)
	FMT_SMF.1(2): Specification of Management Functions (TOE Audit Record Generation)
	FMT_SMF.1(3): Specification of Management Functions (Cryptographic Key Data)
	FMT_SMF.1(4): Specification of Management Functions (Wireless Intrusion Protection)
	FMT_SMF.1(5): Specification of Management Functions (TOE Authentication Data)
	FMT_SMF.1(6): Specification of Management Functions (Policy/ACL Rulesets)

Requirement Class	Requirement Component
	FMT_SMF.1(7): Specification of Management Functions (Session Quota)
	FMT_SMR.1: Security Roles
FPT: Protection of the TSF	FPT_STM_(EXT).1: Extended – Reliable Time Stamps
	FPT_TST_(EXT).1: Extended – TSF Testing
	FPT_TST.1(1): TSF Testing of Cryptographic Modules
	FPT_TST.1(2): TSF Testing of Cryptographic Key Generation
FRU: Resource Utilization	FRU_RSA.1: Maximum Quotas
FTA: TOE Access	FTA_SSL.3: TSF-initiated Termination
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted Path/Channels	FTP_ITC_(EXT).1: Extended – Inter-TSF Trusted Channel
	FTP_TRP.1: Trusted Path

Table 1 TOE Security Functional Components

5.2.1 Security Audit (FAU)

5.2.1.1 Security Alarms (FAU_ARP.1)

FAU_ARP.1.1 The TSF shall take **[the actions specified in column three of Table "WIP signature events, information and actions"]** upon detection of a potential security violation.

Application Note: The Wireless Intrusion Protection software module must be licensed and installed for this capability.

5.2.1.2 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [***minimum***] level of audit; and

c) [**additional auditable events specified in Table 2**].

Requirement	Minimal Auditable Events	Additional Audit Record Contents
FAU_ARP.1	Actions taken due to potential security violations.	None
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_SAA.3	Enabling and disabling of any of the analysis mechanisms Automated responses performed by the tool	None
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the Administrator performing the function
FCS_BCM_(EXT).1	None	None

Requirement	Minimal Auditable Events	Additional Audit Record Contents
FCS_CKM.1(1), (2)	Generation of a key	The identity of the Administrator performing the function
FCS_CKM.2	Success and failure of the activity	None
FCS_CKM_(EXT).2	Error(s) detected during cryptographic key transfer	If available – the ~~authentication credentials~~ **identity** of subjects with which the invalid key is shared[9].
FCS_CKM.4	Destruction of a cryptographic key	If available – The identity of the Administrator performing the function
FCS_COP.1(1),(2),(3),(4)	None	None
FCS_COP_(EXT).1	None	None
FDP_IFC.1(1)	None	None
FDP_IFC.1(2)	None	None
FDP_IFF.1(1)	Decisions to permit requested information flows	None
FDP_IFF.1(2)	Decisions to permit requested information flows	None
FDP_PUD.1_(EXT)	Enabling or disabling TOE encryption of wireless traffic	The identity of the Administrator performing the function.
FDP_RIP.1	None	None
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal)	None
FIA_ATD.1(1),(2)	None	None
FIA_SOS.1	Rejection by the TSF of any tested secret	None
FIA_UAU.1	Use of the authentication mechanism (success or failure)	User identity - the TOE **SHALL NOT** record invalid passwords the audit log.
FIA_UAU_(EXT).5	Failure to receive a response from the remote authentication server	Identification of the Authentication server that did not reply
FIA_UID.2	None	None
FIA_USB.1	Unsuccessful binding of user security attributes to a subject	None
FMT_MOF.1(1)	Changing the TOE encryption algorithm including the selection not to encrypt communications	Encryption algorithm selected (or none)
FMT_MOF.1(2)	Start or Stop of audit record generation	None

[9] Cryptographic keys are not shared among different users

Requirement	Minimal Auditable Events	Additional Audit Record Contents
FMT_MOF.1(3)	Changes to the TOE remote authentication settings; Changes to the threshold of failed authentication attempts; Changes to the session lock timeframe	The identity of the Administrator performing the function.
FMT_MOF.1(4)	None	None
FMT_MSA.1(1)	None	None
FMT_MSA.1(2)	None	None
FMT_MSA.2	All offered and rejected values for security attributes	None
FMT_MSA.3(1)	None	None
FMT_MSA.3(2)	None	None
FMT_MTD.1(1)	Changes to the set of rules used to pre-select audit events.	None
FMT_MTD.1(2), FMT_MTD.1.(3)	Changing the TOE authentication credentials	None – the TOE **SHALL NOT** record authentication credentials in the audit log.
FMT_MTD.1(4)	Changes to the set of Policy/ACL rules	The identity of the Administrator performing the function.
FMT_MTD.1(5)	None	None
FMT_SMF.1(1)-(7)	Use of the management functions	The identity of the Administrator performing the function.
FMT_SMR.1	Modifications to the group of users that are part of a role	None
FPT_STM_(EXT).1	Changes to the time	None
FPT_TST_EXP.1	Execution of the self test	Success or Failure of test
FPT_TST.1(1), (2)	Execution of the self test	Success or Failure of test
FRU_RSA.1	Rejection of allocation operation due to resource limits	None
FTA_SSL.3	TSF Initiated Termination	Termination of an interactive session by the session locking mechanism.
FTA_TAB.1	None	None
FTP_ITC_(EXT).1	Initiation/Closure of a trusted channel;	Identification of the remote entity with which the channel was attempted/created; Success or failure of the event
FTP_TRP.1	Initiation of a trusted path	Identification of the remote entity with which the path was attempted/created; Success or failure of the event

Table 2 TOE Auditable Events

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**information specified in column three of Table 2**].

5.2.1.3 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.4 Simple Attack Heuristics (FAU_SAA.3)

FAU_SAA.3.1 The TSF shall be able to maintain an internal representation of the following signature events **[the subset of system events specified in column one of Table "WIP signature events, information and actions"]** that may indicate a violation of the enforcement of the SFRs.

FAU_SAA.3.2 The TSF shall be able to compare the signature events against the record of system activity discernible from an examination of **[the information specified in column two of Table "WIP signature events, information and actions"]**.

FAU_SAA.3.3 The TSF shall be able to indicate a potential violation of the enforcement of the SFRs when a system event is found to match a signature event that indicates a potential violation of the enforcement of the SFRs.

Signature Event	Information Used to Detect Event	Action on Event
Rogue AP detection	Wireless and wired traffic characteristics	Generate and send alarm of the presence of AP If configured by an administrator, automatically disable the AP
Denial of Service Attack	Anomalous rates of 802.11 management frames	Generate and send alarm
MAC address spoofing	802.11 MAC address sequence analysis	Generate and send alarm
Station disconnection detection	Anomalies in the 802.11 sequence number on specific 802.11 management frames	Generate and send alarm
EAP handshake flood DOS attack	Floods of EAPOL messages requesting 802.1x authentication	Generate and send alarm
Sequence Number analysis	Anomalies in 802.11 MAC sequence numbers of received frames	Generate and send alarm
Null-Probe-Response signature detection	Received SSID element of 0 length in a probe response frame	Generate and send alarm
AirJack signature detection	Received SSID of "AirJack" in beacon frame	Generate and send alarm
NetStumbler Generic signature detection	802.11 data packets with specific patterns in the payload	Generate and send alarm
NetStumbler Version 3.3.0x signature detection	802.11 data packets with specific patterns in the payload	Generate and send alarm
Deauth-Broadcast signature detection	802.11 deauthentication frames with the broadcast MAC as the destination address	Generate and send alarm

Signature Event	Information Used to Detect Event	Action on Event
Misconfigured AP detection	An AP advertising capabilities that do not match known valid AP characteristics	Generate and send alarm If configured by an administrator, deny access by AP

Table 2-1 WIP signature events, information and actions

Application Note: The Wireless Intrusion Protection software module must be licensed and installed for this capability.

5.2.1.5 Selective Audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) ***user identity, event type;***

b) [**device interface, wireless client identity**].

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Extended: Baseline Cryptographic Module (FCS_BCM_(EXT).1)

FCS_BCM_(EXT).1.1 All FIPS-approved cryptographic functions implemented by the TOE shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions implemented by the TOE.

FCS_BCM_(EXT).1.2 **Refinement:** All cryptographic modules implemented in the TOE [***as a combination of hardware and software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS PUB 140-2,Level 3 for the following:***

- ~~***Cryptographic Module Ports and Interfaces;***~~
- ***Roles, Services and Authentication;***
- ~~***Cryptographic Key Management;***~~ ***and***
- ***Design Assurance***

The logical interfaces used for the input and output of plaintext cryptographic key components, authentication data, and CSPs shall be logically separated from all other interfaces using a trusted path or AS02.16 must be satisfied; where "trusted path" is interpreted to include a communications channel established using a FIPS 140-2 Level 2 cryptographic module and the HTTPS protocol between the cryptomodule and the external IT entity].

Application Note: This refinement was required based on the precedent decision PD-0164.

5.2.2.2 Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(1))

FCS_CKM.1.1(1) The TSF shall generate symmetric cryptographic keys using a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and provide integrity protection to generated symmetric keys in accordance with NIST SP 800-57 "Recommendation for Key Management" Section 6.1.

5.2.2.3 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(2))

FCS_CKM.1.1(2) The TSF shall generate asymmetric cryptographic keys in accordance with the mathematical specifications of the FIPS-approved or NIST-recommended standard [**ANSI X9.31-1998**], using a domain parameter generator and [

(1) a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and

*(2) **a prime number generator as specified in ANSI X9.80 "Prime Number Generation, Primality Testing, and Primality Certificates" using random integers with deterministic tests, or constructive generation methods***]

in a cryptographic key generation scheme that meets the following:

- The TSF shall provide integrity protection and assurance of domain parameter and public key validity to generated asymmetric keys in accordance with NIST SP 800-57 "Recommendation for Key Management" Section 6.1.
- Generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates.

5.2.2.4 Cryptographic Key Distribution (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [

1) Manual (Physical) Method, and

2) ***Automated (electronic) Method***]

that meets the following:

- NIST Special Publication 800-57, "Recommendation for Key Management" Section 8.1.5.
- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

5.2.2.5 Extended: Cryptographic Key Handling and Storage (FCS_CKM_(EXT).2)

FCS_CKM_(EXT).2.1 The TSF shall perform a key error detection check on each transfer of key (internal, intermediate transfers).

FCS_CKM_(EXT).2.2 The TSF shall store persistent secret and private keys when not in use in encrypted form or using split knowledge procedures.

FCS_CKM_(EXT).2.3 The TSF shall destroy non-persistent cryptographic keys after a cryptographic administrator-defined period of time of inactivity.

FCS_CKM_(EXT).2.4 The TSF shall prevent archiving of expired (private) signature keys.

5.2.2.6 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a cryptographic key zeroization method that meets the following:

a) Key zeroization requirements of FIPS PUB 140-2, "Security Requirements for Cryptographic Modules"

b) Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete.

c) The TSF shall zeroize each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/critical cryptographic security parameter to another location.

d) For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time.

e) For volatile memory and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify.

5.2.2.7 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1.1(1) **Refinement:** The cryptomodule shall perform encryption and decryption using the FIPS-Approved Security Function AES algorithm **(required for trusted path) and TDES algorithm** operating in [***CBC, CCM (AES only)***] mode(s) supporting key sizes of [***128 bits, 168bits (TDES only), 192 bits, and 256 bits***].

5.2.2.8 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1.1(2) **Refinement:** The TSF shall perform cryptographic signature services using the FIPS-approved security function [

RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [1024 or 2048 bits]

]

that meets NIST Special Publication 800-57, "Recommendation for Key Management."

Application Note: RSA 2048 actually provides 112 bits rather than 128 bits of security.

5.2.2.9 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1.1(3) **Refinement:** The TSF shall perform cryptographic hashing services using the FIPS-approved security function Secure Hash Algorithm and message digest size of [***160 bits, 256 bits (required for trusted path)***].

5.2.2.10 Cryptographic Operation (for cryptographic key agreement) (FCS_COP.1(4))

FCS_COP.1.1(4) **Refinement:** The TSF shall perform cryptographic key agreement services using the FIPS-approved security function as specified in NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" [

[Diffie-Hellman] and cryptographic key sizes (modulus) of [1024 or 2048 bits]]

that meets NIST Special Publication 800-57, "Recommendation for key Management".

5.2.2.11 Cryptographic Operation (for TLS Protocol) (FCS_COP.1(5))

FCS_COP.1.1(5) The TSF shall perform [**Transport Level Security v1.0**] in accordance with ~~a~~ **the** specified cryptographic algorithm**s** [**AES, TDES, SHA, RSA, Diffie-Hellman**] and cryptographic key sizes [**128, 168 (TDES), 192, and 256 bits (AES), 1024 or 2048 bits (RSA), not applicable for SHA, and 1024 or 2048 bits (Diffie-Hellman)**] that meet the following:[**AES or TDES (see FCS_COP.1(1)), RSA (see FCS_COP.1(2)), SHA (see FCS_COP.1(3)), Diffie-Hellman (see FCS_COP.1(4)), and TLSv1.0 (RFC 2246)**].

5.2.2.12 Cryptographic Operation (for SSH Protocol) (FCS_COP.1(6))

FCS_COP.1.1(6) The TSF shall perform [**Secure Shell v2.0**] in accordance with ~~a~~ **the** specified cryptographic algorithm**s** [**AES, SHA, RSA, Diffie-Hellman**] and cryptographic key sizes [**128, 192, and 256 bits (AES), 2048 bits (RSA), not applicable for SHA, and 2048 bits (Diffie-Hellman)**] that meet the following:[**AES (see FCS_COP.1(1)), RSA (see FCS_COP.1(2)), SHA (see FCS_COP.1(3)), Diffie-Hellman (see FCS_COP.1(4)), and SSHv2.0 (Transport Layer - RFC 4253, User Authentication Layer – RFC 4252, Connection Layer – RFC 4255)**].

5.2.2.13 Cryptographic Operation (for IKE/IPSec Protocols) (FCS_COP.1(7))

FCS_COP.1.1(7) The TSF shall perform [**IKE/IPSec**] in accordance with ~~a~~ **the** specified cryptographic algorithm**s** [**AES, TDES, SHA, Diffie-Hellman**] and cryptographic key sizes [**128, 168 (TDES),192, and 256 bits (AES), not applicable for SHA, and 1024 or 2048 bits (Diffie-Hellman)**] that meet the following:[**AES (see FCS_COP.1(1)), SHA (see FCS_COP.1(3)), Diffie-Hellman (see FCS_COP.1(4)), and IKE/IPSec (Internet Key Exchange – RFC 2409, Encapsulating Security Payload – RFC 4303)**].

Application Note: The TOE does not implement or support the Authentication Header (AH) protocol.

5.2.2.14 Extended: Random Number Generation (FCS_COP_(EXT).1)

FCS_COP_(EXT).1.1 The TSF shall perform all random number generation (RNG) services in accordance with a FIPS-approved RNG [**ANSI X9.31-1998 algorithm**] seeded by [

(1) one or more independent hardware-based entropy sources, and/or

(2) one or more independent software-based entropy sources, and/or]

FCS_COP_(EXT).1.2 The TSF shall defend against tampering of the random number generation (RNG)/ pseudorandom number generation (PRNG) sources.

5.2.3 User Data Protection (FDP)

5.2.3.1 Subset Information Flow Control (FDP_IFC.1(1))

FDP_IFC.1.1(1) The TSF shall enforce the **[Aruba Information Flow Policy]** on:

a.) **[subjects: wireless clients that send information to (or through) the TOE interface;**

b.) **information: network packets; and**

c.) **operation: send**]

Application Note: The Policy Enforcement Firewall software module must be licensed and installed for this capability.

5.2.3.2 Simple Security Attributes (FDP_IFF.1(1))

FDP_IFF.1.1(1) The TSF shall enforce the **[Aruba Information Flow Policy]** based on the following types of subject and information security attributes:

a.) **[subject security attributes:**

i. IP address of wireless client

b.) **information security attributes:**

i. **Source address**

ii. **Source port identifier (TCP or UDP source port number)**

iii. **Destination address**

iv. **Destination port identifier (TCP or UDP destination port number)**

v. **Protocol**

vi. **Service]**

FDP_IFF.1.2(1) The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[**

i. For each TSF operation (except NAT), if the Firewall Policy rules match the specific characteristics (information security attributes) about a data packet passing through the Aruba controller, then the specified action (one of Permit, Drop, Reject) will be performed, or

ii. For each TSF operation (except NAT), if the ACL rules match the specific characteristics (information security attributes) about a data packet passing through the Aruba controller, then the specified action (one of Permit, Drop, Reject) will be performed, or

iii. **For NAT[10] operations, subjects on a network can cause information to flow through the TOE to another connected network if:**

- **all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;**
- **the address of the source subject, in the information, translates to an internal network address (src-NAT, dual-NAT);**
- **and the address of the destination subject, in the information, translates to an address on the other connected network (dst-NAT, dual-NAT).]**

FDP_IFF.1.3(1) The TSF shall enforce the **[no additional rules]**.

FDP_IFF.1.4(1) The TSF shall explicitly authorize an information flow based on the following rules: **[no additional rules]**.

FDP_IFF.1.5(1) The TSF shall explicitly deny an information flow based on the following rules: **[no additional rules]**.

Application Note: The Policy Enforcement Firewall software module must be licensed and installed to enforce these filtering capabilities.

5.2.3.3 Subset Information Flow Control (FDP_IFC.1(2))

FDP_IFC.1.1(2) The TSF shall enforce the **[VLAN Information Flow Policy]** on:

a.) **[subjects: wireless clients that send information through the TOE interface;**
b.) **information: network packets; and**
c.) **operation: send**]

5.2.3.4 Simple Security Attributes (FDP_IFF.1(2))

FDP_IFF.1.1(2) The TSF shall enforce the **[VLAN Information Flow Policy]** based on the following types of subject and information security attributes:

a.) **[subject security attributes:**
  i. **Role**
  ii. **Authentication method**
b.) **information security attributes:**
  i. **Source Port**
  ii. **Destination Port]**

FDP_IFF.1.2(2) The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[**

Client information can flow to and from a VLAN only if that client and port is assigned to that VLAN (if not, traffic is dropped).

i. **After client authentication, the client will be assigned to the VLAN configured for a specific role or authentication method, such as 802.1x or VPN. Role takes precedence over authentication method.**

ii. **The port is configured to carry traffic for that specific VLAN or VLANs.**]

FDP_IFF.1.3(2) The TSF shall enforce the **[If no rule in FDP_IFF.1.2 matches, the client will be assigned to the default[11] VLAN.]**.

FDP_IFF.1.4(2) The TSF shall explicitly authorize an information flow based on the following rules: **[no additional rules]**.

[10] NAT is not supported for IPv6.
[11] The default VLAN is the VLAN configured for the WLAN.

FDP_IFF.1.5(2) The TSF shall explicitly deny an information flow based on the following rules: [**no additional rules**].

5.2.3.5 Extended: Protection of User Data (FDP_PUD_(EXT).1)

FDP_PUD_(EXT).1.1 When the administrator has enabled encryption, the TSF shall:

- encrypt authenticated user data transmitted to a wireless client from the radio interface of the wireless access system using the cryptographic algorithm(s) specified in FCS_COP.1(1);
- decrypt authenticated user data received from a wireless client by the radio interface of the wireless access system using the cryptographic algorithm(s) specified in FCS_COP.1(1);

5.2.3.6 Subset Residual Information Protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [**network packet objects**].

5.2.4 Identification and Authentication (FIA)

5.2.4.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [***an administrator configurable positive integer within the range of [0-255]***] ~~of~~ unsuccessful authentication attempts occur related to [**wireless clients and remote administrators logging on to the WLAN access system**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [***surpassed***], the TSF shall [**prevent login by wireless clients ~~administrators~~ until an action is taken by a local Administrator**].

Application Note: In addition to blacklisting the users, the TOE will log the event and send a SNMP trap.

5.2.4.2 Administrator Attribute Definition (FIA_ATD.1(1))

FIA_ATD.1.1(1) The TSF shall maintain the following minimum list of security attributes belonging to individual administrators: [**password, [username]]**.

5.2.4.3 User Attribute Definition (FIA_ATD.1(2))

FIA_ATD.1.1(2) The TSF shall maintain the following minimum list of security attributes belonging to individual remotely authenticated users: [**session key, role, username, password]**.

Application Note: Username and password are maintained by the TOE when the authentication mechanism is provided by the TOE.

5.2.4.4 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following:** [

a) **For each attempt to use the password authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000.**

b) **For multiple attempts to use the password authentication mechanism during a one-minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000].**

5.2.4.5 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [**user identification as stated in FIA_UID.2**] on behalf of users to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.4.6 Extended: Multiple Authentication Mechanisms (FIA_UAU_(EXT).5)

FIA_UAU_(EXT).5.1 The TSF shall provide local authentication, and a remote authentication mechanism to perform user authentication.

FIA_UAU_(EXT).5.2 The TSF shall, at the option of the administrator, invoke the remote authentication mechanism for administrators and wireless LAN users.

5.2.4.7 User Identification Before Any Action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.4.8 User-subject Binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**username**].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**none**].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**none**].

5.2.5 Security Management (FMT)

5.2.5.1 Management of Cryptographic Security Functions Behaviour (FMT_MOF.1(1))

FMT_MOF.1.1(1) The TSF shall restrict the ability to [***modify the behaviour of***] the cryptographic functions [

- Crypto: load a key
- Crypto: delete/zeroize a key
- Crypto: set a key lifetime
- Crypto: set the cryptographic algorithm
- Crypto: set the TOE to encrypt or not to encrypt wireless transmissions
- Crypto: execute self tests of TOE hardware and the cryptographic functions]

to [**administrators**].

5.2.5.2 Management of Audit Security Functions Behaviour (FMT_MOF.1(2))

FMT_MOF.1.1(2) The TSF shall restrict the ability to [***enable, disable, and modify the behaviour of***] the functions [

- Audit: pre-selection of the events which trigger an audit record,
- Audit: start and stop of the audit function]

to [**administrators**].

5.2.5.3 Management of Authentication Security Functions Behaviour (FMT_MOF.1(3))

FMT_MOF.1.1(3) The TSF shall restrict the ability to [***modify the behaviour of***] the Authentication functions [

- Auth: allow or disallow the use of an authentication server
- Auth: set the number of authentication failures that must occur before the TOE takes action to disallow future logins
- Auth: set the length of time a session may remain inactive before it is terminated]

to [**administrators**].

5.2.5.4 Management of Wireless Intrusion Protection Security Functions Behaviour (FMT_MOF.1(4))

FMT_MOF.1.1(4) The TSF shall restrict the ability to [***determine the behavior of, enable, disable, and modify the behaviour of***] the functions [

- WIP: signature events (wireless intrusion profiles)
- WIP: actions to be taken upon detection of a potential security violation]

to [**administrators**].

5.2.5.5 Management of Security Attributes (FMT_MSA.1(1))

FMT_MSA.1.1(1) The TSF shall enforce the [**Aruba Information Flow Policy**] to restrict the ability to [***modify***] the **information** security attributes [**referenced in the FDP_IFF.1(1).1 and operations referenced in the FDP_IFC.1(1).1**] to [**authorized administrator**].

5.2.5.6 Management of Security Attributes (FMT_MSA.1(2))

FMT_MSA.1.1(2) The TSF shall enforce the [**VLAN Information Flow Policy**] to restrict the ability to [***modify***] the security attributes [**referenced in the FDP_IFF.1(2).1**] to [**authorized administrator**].

5.2.5.7 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: FMT_MSA.2 was included to meet dependencies in cryptography-related requirements FCS_CKM.1, FCS_CKM.2, FCS_CKM.4 and FCS_COP.1. The security attributes are cryptographic key values. Cryptographic keys are considered secure if generated, distributed and managed by a FIPS 140-2 validated cryptographic module.

5.2.5.8 Static Attribute Initialization (FMT_MSA.3(1))

FMT_MSA.3.1(1) The TSF shall enforce the [**Aruba Information Flow Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(1) The TSF shall allow the [**authorized administrator**] to specify alternative initial values to override the default values when an object or information is created.

5.2.5.9 Static Attribute Initialization (FMT_MSA.3(2))

FMT_MSA.3.1(2) The TSF shall enforce the [**VLAN Information Flow Policy**] to provide [***permissive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(2) The TSF shall allow the [**authorized administrator**] to specify alternative initial values to override the default values when an object or information is created.

5.2.5.10 Management of Audit Pre-selection Data (FMT_MTD.1(1))

FMT_MTD.1.1(1) The TSF shall restrict the ability to [***query, modify, clear, [create]***] the [**set of rules used to pre-select audit events**] to [**the administrator**].

5.2.5.11 Management of Authentication Data (Administrator) (FMT_MTD.1(2))

FMT_MTD.1.1(2) The TSF shall restrict the ability to [***query, modify, delete, clear, [create]***] the [**authentication credentials, user identification credentials**] to [**administrators**].

5.2.5.12 Management of Authentication Data (User) (FMT_MTD.1(3))

FMT_MTD.1.1(3) The TSF shall restrict the ability to [***modify***] the [**user authentication credentials**] to [~~**TOE users**~~ **administrators**].

Application Note: This refinement is made because only administrators can change user credentials.

5.2.5.13 Management of TSF Data (Policy/ACL Rulesets) (FMT_MTD.1(4))

FMT_MTD.1.1(4) The TSF shall restrict the ability to [***query, modify, delete, [create]***] the [**Aruba Information Flow Policy Rules and VLANs**] to [**administrators**].

5.2.5.14 Management of TSF Data (Sessions Quota) (FMT_MTD.1(5))

FMT_MTD.1.1(5) The TSF shall restrict the ability to [***modify***] the [**the maximum number of sessions (per user) setting**] to [**administrators**].

5.2.5.15 Specification of Management Functions (Cryptographic Function) (FMT_SMF.1(1))

FMT_SMF.1.1(1) The TSF shall be capable of performing the following security management functions: [**query and set the encryption/decryption of network packets (via FCS_COP.1(1)) in conformance with the administrators configuration of the TOE**].

5.2.5.16 Specification of Management Functions (TOE Audit Record Generation) (FMT_SMF.1(2))

FMT_SMF.1.1(2) The TSF shall be capable of performing the following security management functions: [**query, enable or disable Security Audit**].

5.2.5.17 Specification of Management Functions (Cryptographic Key Data) (FMT_SMF.1(3))

FMT_SMF.1.1(3) The TSF shall be capable of performing the following security management functions: [**~~query, ~~set, modify, and delete the cryptographic keys and key data in support of the Wireless Client Policy ~~FDP_PUD_(EXT) and enable/disable verification of cryptographic key testing~~**].

Application Note: This refinement is made because of PD-0145.

5.2.5.18 Specification of Management Functions (Wireless Intrusion Protection) (FMT_SMF.1(4))

FMT_SMF.1.1(4) The TSF shall be capable of performing the following security management functions: [**determine the behavior of, enable, disable, and modify the behavior of WIP signature events and actions in support of FAU_SAA.3 and FAU_ARP.1**].

5.2.5.19 Specification of Management Functions (TOE Authentication Data) (FMT_SMF.1(5))

FMT_SMF.1.1(5) The TSF shall be capable of performing the following security management functions: [**query, modify, delete, clear and create the administrator authentication credentials and user identification credentials, modify the user authentication credentials**].

5.2.5.20 Specification of Management Functions (Policy/ACL Rulesets) (FMT_SMF.1(6))

FMT_SMF.1.1(6) The TSF shall be capable of performing the following security management functions: [**query, modify, delete and create the information flow Policy/ACL rulesets and VLANs**].

5.2.5.21 Specification of Management Functions (Sessions Quota) (FMT_SMF.1(7))

FMT_SMF.1.1(7) The TSF shall be capable of performing the following security management functions: [**modify the maximum number of sessions (per user) setting**].

5.2.5.22 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**administrator, non-administrator, wireless user**].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 Extended: Reliable Time Stamps (FPT_STM_(EXT).1)

FPT_STM_(EXT).1.1 The TSF shall be able to provide reliable time stamps, synchronized via an external time source, for its own use.

5.2.6.2 Extended: TSF Testing (FPT_TST_(EXT).1)

FPT_TST_(EXT).1.1 The TSF shall run a suite of self tests during the initial start-up and also either periodically during normal operation, or at the request of an authorized administrator to demonstrate the correct operation of the TSF.

FPT_TST_(EXT).1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.

5.2.6.3 TSF Testing (for cryptography) (FPT_TST.1(1))

FPT_TST.1.1(1) The TSF shall run a suite of self tests in accordance with FIPS PUB 140-2 and Appendix C of this profile [***during initial start-up (on power on), at the request of the* cryptographic administrator *(on demand)***, ***under various conditions [defined in section 4.9.1 of FIPS 140-2, and periodically (at least once a day)]***] to demonstrate the correct operation of the following cryptographic functions: [

- ***key error detection;***
- ***cryptographic algorithm;***
- ***RNG/PRNG***].

FPT_TST.1.2(1) The TSF shall provide authorized ~~cryptographic~~ administrators with the capability to verify the integrity of [***TSF data related to the cryptography by using TSF-provided cryptographic functions***].

FPT_TST.1.3(1) The TSF shall provide authorized ~~cryptographic~~ administrators with the capability to verify the integrity of stored TSF executable code related to the cryptography by using TSF-provided cryptographic functions.

5.2.6.4 TSF Testing (for key generation components) (FPT_TST.1(2))

FPT_TST.1.1(2) The TSF shall perform self tests immediately after generation of a key to demonstrate the correct operation of [***each key generation component. If any of these tests fail, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140-2 for failing a self-test, and this event will be audited***].

FPT_TST.1.2(2) The TSF shall provide authorized ~~cryptographic~~ administrators with the capability to verify the integrity of [***TSF data related to the key generation by using TSF-provided cryptographic functions***].

FPT_TST.1.3(2) The TSF shall provide authorized ~~cryptographic~~ administrators with the capability to verify the integrity of stored TSF executable code related to the key generation by using TSF-provided cryptographic functions.

5.2.7 Resource Utilization (FRU)

5.2.7.1 Maximum Quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [**number of sessions**] that [*defined group of users*] can use [***simultaneously***].

5.2.8 TOE Access (FTA)

5.2.8.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate ~~an~~ a local interactive or wireless session after ~~a~~ an [**administrator configurable time interval of user inactivity**].

5.2.8.2 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

5.2.9 Trusted Path/Channels (FTP)

5.2.9.1 Extended: Inter-TSF Trusted Channel (FTP_ITC_(EXT).1)

FTP_ITC_(EXT).1.1 The TOE shall provide an encrypted communication channel between itself and entities in the TOE IT Environment that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC_(EXT).1.2 The TSF shall permit [***the TSF,***] or the IT Environment entities to initiate communication via the trusted channel.

FTP_ITC_(EXT).1.3 The TSF shall initiate communication via the trusted channel for [**all authentication functions, remote logging, time, [*none*]**].

Application Note: The VPN Server software module must be licensed and installed for this capability.

5.2.9.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TOE shall provide a communication path between itself and wireless users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***modification, disclosure, [replay]***].

FTP_TRP.1.2 The TSF shall permit [***wireless client devices]*** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***wireless user authentication, [none]***].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security Architecture Description
	ADV_FSP.4: Complete Functional Specification
	ADV_IMP.1: Implementation Representation of the TSF
	ADV_TDS.3: Basic Modular Design
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.4: Production Support, Acceptance Procedures and Automation
	ALC_CMS.4: Problem Tracking CM Coverage
	ALC_DEL.1: Delivery Procedures
	ALC_DVS.1: Identification of Security Measures
	ALC_FLR.2: Flaw Reporting Procedures
	ALC_LCD.1: Developer Defined Life-cycle Model
	ALC_TAT.1: Well-defined Development Tools

Requirement Class	Requirement Component
ATE: Tests	ATE_COV.2: Analysis of Coverage
	ATE_DPT.2: Testing: Security Enforcing Modules
	ATE_FUN.1: Functional Testing
	ATE_IND.2: Independent Testing - Sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused Vulnerability Analysis

Table 2-2 EAL 4 augmented with ALC_FLR.2 Assurance Components

5.3.1 Development (ADV)

5.3.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Complete Functional Specification (ADV_FSP.4)

ADV_FSP.4.1d The developer shall provide a functional specification.

ADV_FSP.4.2d The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.4.1c The functional specification shall completely represent the TSF.

ADV_FSP.4.2c The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3c The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4c The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.4.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 Implementation Representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2d The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

ADV_IMP.1.1c The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2c The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3c The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

ADV_IMP.1.1e The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

5.3.1.4 Basic Modular Design (ADV_TDS.3)

ADV_TDS.3.1d The developer shall provide the design of the TOE.

ADV_TDS.3.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.3.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2c The design shall describe the TSF in terms of modules.

ADV_TDS.3.3c The design shall identify all subsystems of the TSF.

ADV_TDS.3.4c The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5c The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6c The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.

ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.3.9c The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10c The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_TDS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

5.3.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d The developer shall provide operational user guidance.

AGD_OPE.1.1c The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1c The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle Support (ALC)

5.3.3.1 Production Support, Acceptance Procedures and Automation (ALC_CMC.4)

ALC_CMC.4.1d The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.4.2d The developer shall provide the CM documentation.
ALC_CMC.4.3d The developer shall use a CM system.
ALC_CMC.4.1c The TOE shall be labelled with its unique reference.
ALC_CMC.4.2c The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.4.3c The CM system shall uniquely identify all configuration items.
ALC_CMC.4.4c The CM system shall provide automated measures such that only authorised changes are made to the configuration items.
ALC_CMC.4.5c The CM system shall support the production of the TOE by automated means.
ALC_CMC.4.6c The CM documentation shall include a CM plan.
ALC_CMC.4.7c The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.4.8c The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ALC_CMC.4.9c The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.4.10c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.
ALC_CMC.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.2 Problem Tracking CM Coverage (ALC_CMS.4)

ALC_CMS.4.1d The developer shall provide a configuration list for the TOE.
ALC_CMS.4.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.
ALC_CMS.4.2c The configuration list shall uniquely identify the configuration items.
ALC_CMS.4.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2d The developer shall use the delivery procedures.
ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.4 Identification of Security Measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.

ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.3.5 Flaw Reporting Procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.6 Developer Defined Life-cycle Model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.7 Well-defined Development Tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of each development tool.

ALC_TAT.1.1c Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3c The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Tests (ATE)

5.3.4.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 Testing: Security Enforcing Modules (ATE_DPT.2)

ATE_DPT.2.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.

ATE_DPT.2.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.2.3c The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.

ATE_DPT.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.3 Functional Testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.

ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.4 Independent Testing - Sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.

ATE_IND.2.1c The TOE shall be suitable for testing.

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3e The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.5 Vulnerability Assessment (AVA)

5.3.5.1 Focused Vulnerability Analysis (AVA_VAN.3)

AVA_VAN.3.1d The developer shall provide the TOE for testing.

AVA_VAN.3.1c The TOE shall be suitable for testing.

AVA_VAN.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions and mechanisms.

6.1 TOE Security Functions

6.1.1 Security Audit

The TOE has an audit generation mechanism to record security and non-security relevant events at a minimum level of audit. There are several types of category for audit logs including Network, System, Security, Wireless, and User. The Network log category can include all network packets, protocol packet dump, mobility, and DHCP events. The System log category can include all system, configuration, SNMP, and web server events. The Security log category can include all security, AAA, firewall, packet trace, VPN, 802.1x, and IKE events. The Wireless log category can include all wireless events. The User log category can include all user, captive portal, VPN, 802.1X, and RADIUS user events.

The administrator can turn on or off (include or exclude) auditable events based on specific criteria. The administrator can configure the logging level (event type) for each of the modules of the ArubaOS. The inclusion and exclusion of audited events for the event type is performed by using the "logging" command at the CLI. The Web GUI provides similar functionality through the *Monitoring->Management->Logging* panel. For user identity, device interface, and wireless client, this is done by using combinations of the 'user-debug', 'process authmgr', and 'process aaa' options of the 'logging level' CLI command.. There are a total of eight syslog logging levels and the default logging level for all categories is Warning. The logging levels are defined as followed:

Emergency	- Panic conditions that occur when the system becomes unusable.
Alert	- Any condition requiring immediate attention and correction.
Critical	- Any critical conditions such as a hard drive error.
Errors	- Error conditions
Warning	- Warning messages
Notice	- Significant events of a non-critical and normal nature.
Informational	- Messages of general interest to system users.
Debug	- Messages containing information useful for debugging.

The TOE generates audit records for auditable events. The audit function is integrated into each module of ArubaOS. In particular, when an auditable event occurs, the module executes a logging API call that records event information to the external audit (syslog) server. The audit records are transmitted to the audit server over a trusted channel and are stored and protected by the audit server. At the audit server, the administrator is provided with an interface (part of operating environment) to read audit logs. Though not required by PP, the TOE also stores audit records locally and provides CLI and WebUI capabilities to view the contents of the audit trail. For each audit event, the following information is recorded:

Event Type The logging level

Subject Identity The identity of the subject involved in the event. For identified users, the subject identity is represented by the username. For other subjects, the subject identity is represented by the IP address for wired network subjects, and by the MAC address for wireless network subjects.

Date and Time The date and time when each event occurred. The time can be obtained internally or from a trusted NTP server in the operating environment.

Outcome Success or failure of the event

The following minimum events are audited:

1. Actions taken due to security violations
2. Automated responses performed by the tool

3. Enabling and disabling of the WIP signature-based detection
4. All modification to audit configuration
5. Generation of key
6. Destruction of a cryptographic key
7. Success and failure of key distribution
8. Decision to permit information flow
9. Enabling and disabling encryption of wireless traffic
10. The reaching of the threshold and subsequent lockout
11. Rejection of the password due to minimum length
12. Use of authentication mechanism
13. Failure to receive a response from the authentication server
14. unsuccessful binding of user security attribute to subject
15. Changing the encryption algorithm
16. Errors detected during cryptographic key transfer
17. Start or stop audit function
18. Modifying to authentication setting, threshold of failed authentication attempts, session lockout
19. Restoration to the normal state (if applicable)
20. All offered and rejected values for security attributes
21. Changes to set of rules used to pre-select audit events
22. Changing authentication credentials
23. Changes to Policy/ACL rules
24. User of management functions
25. Modification to group of users part of role
26. Changes to the time
27. Execution of self test
28. Rejection due to maximum session allowed
29. Session termination
30. Initiation and closure of a trusted channel
31. Initiation of a trusted path

Wireless Intrusion Protection involves analysis of wired and wireless traffic to detect anomalous activity, based on a variety of information including the configuration of valid APs and the characteristics of received wireless frames. Many WLAN intrusion and attack tools generate characteristic signatures that can be detected by the TOE. The Controller is pre-configured with several known signatures, and also includes the ability for administrator to create new signatures. To create new signature, the administrator creates signature rules that match the attribute to a value. The administrator can configure the Aruba Mobility Controller to detect and, where possible, protect the managed networks from specific types of network intrusion attacks.

Attributes	Description
BSSID	BSSID field in the 802.11 frame header.
Destination MAC address	Destination MAC address in 802.11 frame header.
Source MAC address	Source MAC address of the 802.11 frame.
Frame Type	Type of 802.11 frame. For each type of frame further details can be specified to filter and detect only the required frames. It can be one of the following: • association • auth • beacon • control (all control frames) • data (all data frames) • deauth • deassoc • management (all management frames) • probe-request • probe-response
SSID	For beacon, probe-request, and probe-response frame types, specify the SSID as either a string or hex pattern.
SSID-length	For beacon, probe-request, and probe-response frame types, specify the SSID length. Maximum length is 32 bytes.
Payload	Pattern at a fixed offset in the payload of a 802.11 frame. Specify the pattern to be matched as a string or hex pattern. Maximum length is 32 bytes.
Offset	When a payload pattern is configured, specify the offset in the payload where the pattern is expected to be found in the frame.
Sequence Number	Sequence number of the frame.
EAP rate threshold	Number of EAP handshakes that must be received within the EAP Rate Time Interval to trigger an alarm. Default: 60
EAP rate time interval	Time, in seconds, during which the configured number of EAP handshakes must be received to trigger an alarm. Default: 3 seconds

The TOE (MC and AP) can generate and send SNMP traps to the operating environment (SNMP server) to alert the administrators of potential problems, misconfiguration, or attacks. The administrators must monitor the SNMP server regularly for the alarms sent by the TOE. The following are key SNMP traps generated by the TOE: user entry created/deleted/authenticated/de-authenticated/authentication failed, user table full, ACL table full, adhoc networks detected, AP interference, SSID misconfiguration, OUI misconfiguration, Short Preamble misconfiguration, valid SSID violation, weak or repeat WEP-IV violation, address spoofing, bandwidth rate exceeded, frame retry rate exceeded, station policy violation, station impersonation, reserved channel impersonation, signature match detected, and unsecure AP detected. Administrator can enable a policy to automatically disable APs that are classified as a rogue APs by the TOE. Administrator can also configure a list of parameters that defines their characteristics of a misconfigured AP. These parameters can include preamble type, WEP configuration, OUI of valid MAC addresses, valid channels, and ESSID. The system can also be configured to detect an AP using a weak

WEP key. If an AP is detected as misconfigured, the system will deny access to the misconfigured AP if protection is enabled. This is achieved by sending spoofed deauthentication management frames to the misconfigured AP to terminate the authentication association.

The Aruba Mobility Controller can be configured to detect and, where possible, protect the managed networks from the following types of network intrusion attacks:

> **Rogue AP**, using the *Configuration-> WLAN Intrusion Detection-> Rogue AP* panel of the Web GUI
>
> **Denial of Service**, using the *Configuration-> WLAN Intrusion Detection-> Denial of Service* panel of the Web GUI, including: (1) Rate Analysis
>
> **Man-in-the-Middle**, using the *Configuration-> WLAN Intrusion Detection-> Man-in-the-Middle* panel of the Web GUI, including: (1) MAC Spoofing (2) Station Disconnection Detection (3) EAP Handshake Analysis (4) Sequence Number Analysis
>
> **Signature Detection**, using the *Configuration-> WLAN Intrusion Detection-> Signatures* panel of the Web GUI, including: (1) Null-Probe-Response (2) AirJack (3) NetStumbler Generic (4) NetStumbler Version 3.3.0x (5) Deauth-Broadcast
>
> **WLAN Policies**, using the *Configuration-> WLAN Intrusion Detection-> Policies* panel of the Web GUI, including: (1) Misconfigured AP Protection.

The Security Audit function is designed to satisfy the following security functional requirements:

- FAU_ARP.1: The TOE can generate the appropriate alarms (using SNMP traps), disable AP, and/or deny access by AP if a potential security violation is detected.
- FAU_GEN.1: The TOE generates audit events for various purposes such as security and trouble shooting. The events include startup and shutdown of audit function, all authentication attempts, all administrative actions, and all required auditable events as specified in Table 2. At a minimum, each event includes date and time, logging level (event type), subject identity, and outcome of event.
- FAU_GEN.2: The TOE associates user id to the appropriate audit event. In other words, the user is identified by the username in the audit record.
- FAU_SAA.3: The TOE can be configured to detect using signature events and, where possible, protect managed networks from intrusion attacks.
- FAU_SEL.1: The TOE provides administrators the capability to include or exclude audit events based on event type, user id (username), device interface (e.g., VLAN0, ETH1), and wireless client identity (IP address).

6.1.2 Cryptographic Support

The TOE meets FIPS 140-2 requirements by allowing the administrator to enable a FIPS operating mode. The CC evaluated configuration of the TOE requires the use of this FIPS operating mode. In this mode, only FIPS-approved algorithms are allowed for cryptographic services (e.g., encryption, hashing, digital signature, etc.). All use of cryptographic services (e.g., TLSv1, IPSec/IKE, SSHv2, etc.) can only utilize FIPS-approved algorithms for the underlying algorithms. All models are FIPS-certified at overall Level 2. This ensures that tamper-evident seals are placed around the enclosure (as specified by FIPS 140-2 requirements) to detect any tampering. In addition, at Level 2, any ventilation holes or slots must be small or obstructed to prevent probing of the inside. The Cryptographic security function is described in the context of how it satisfies the cryptographic security requirements.

The Cryptographic Support function is designed to satisfy the following security functional requirements:

- FCS_BCM_(EXT).1: The TOE is implemented as a combination of software and hardware. Cryptographic processing occurs only in software on some platforms, while others use commercial or custom ASIC's to accelerate cryptographic operations. The TOE meets FIPS 140-2 requirements by allowing the administrator to enable a FIPS operating mode. When operating in FIPS mode, ArubaOS offers a restricted set of configuration options and enforces a more restrictive set of security practices.

The TOE has been FIPS 140-2 validated with an overall level 2 and a minimum of level 3 in the following areas:

- Roles, Services and Authentication;
- Design Assurance; and

The certificate numbers for the completed FIPS validation are FIPS 140-2 Certificates: #1297, #1116, #1109, #1077, and #1075. The FIPS 140-2 Level 2 cryptographic module implements the algorithms and protocols (identified below) used to secure and protect the trusted paths and trusted channels between the remote administrators and trusted external IT entity, respectively.

- FCS_CKM.1(1): The cryptomodule only generates symmetric keys using ANSI.X9.31-1998 and provide integrity protection to generated keys in accordance with NIST SP 800-57. The TSF employs a FIPS 140-2 approved software RNG that complies with the ANSI X9.31 standard.
- FCS_CKM.1(2): The TSF generates 1024 or 2048-bit DH keys and 1024 or 2048-bit RSA keys, which are equivalent or greater in strength to 80 or 112 bit symmetric keys. Prime numbers are generated using a method that complies with ANSI X9.80. The FIPS-approved RSA algorithm is defined by ANSI X9.31-1998. Support for distribution of symmetric keys is in accordance with NIST SP 800-57.
- FCS_CKM.2: The TSF supports the manual and electronic distribution of cryptographic keys. Support for distribution of symmetric keys is in accordance with NIST SP 800-57 and 800-56A.
- FCS_CKM_(EXT).2: A parity check is performed whenever a key is internally transferred. All keys are encrypted when not in use. Administrators can define a period of inactivity, after which the ArubaOS will destroy non-persistent cryptographic keys. When no longer needed, memory space used by a key is overwritten three times using a variable bit pattern. ArubaOS does not provide a mechanism to archive expired private signature keys.
- FCS_CKM.4: The TSF supports a key zeroization command that destroys all keying material, overwriting it three times with varying bit patterns. In addition, this command resets the device to the factory default configuration. Further, freed key storage memory is always zeroized whenever the key is moved, copied or deleted.
- FCS_COP.1(1): The cryptomodule supports a FIPS-approved implementation of AES-CBC, using 128, 192 and 256 bit keys and TDES using 168 bit keys. The AES-CBC encryption algorithm can be used to encrypt data in TLS, SSH, and IPSec communication sessions. The TDES-CBC encryption algorithm can be used to encrypt data in TLS and IPSec communication sessions.
- FCS_COP.1(2): The TSF supports the digital signature algorithm RSA with a key size (modulus) of 1024 or 2048 bits. The RSA algorithm is used to perform digital signature generation and verification.
- FCS_COP.1(3): The TSF supports cryptographic hashing via the SHA-1 or SHA-256 algorithm. The cryptographic hash algorithm is used by the TOE to provide integrity protection of the TSF data and code.
- FCS_COP.1(4): The TSF supports cryptographic key agreement via Diffie-Hellman. The TOE generates Diffie-Hellman session parameters used during the TLS and IPSec/IKE handshakes. IPSec/IKE pre-shared keys are not generated by the TOE. They are set by the administrator using the management interface. The Diffie-Hellman provides a key agreement algorithm for finite field-based key agreement algorithm and cryptographic key sizes (modulus) of 1024 or 2048 bits, and meets NIST PUB 800-57.
- FCS_COP.1(5): The TSF implements TLS v1.0 protocol as specified in RFC 2246 (vendor affirmed). The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. The TLS Record Protocol provides connection security by using symmetric cryptography for data encryption (AES and TDES) and message integrity check for data integrity (SHA). The TLS Handshake Protocol, allows the server and client to authenticate each other (RSA), to negotiate an encryption and digest algorithms and cryptographic keys, and to generate unique encryption keys for each session (Diffie-hellman). A CipherSuite defines a cipher specification supported in TLS Version1.0. The following CipherSuite options are allowed:

  - TLS_DHE_RSA_WITH_AES_256_CBC_SHA
  - TLS_RSA_WITH_AES_256_CBC_SHA
  - TLS_DHE_RSA_WITH_AES_128_CBC_SHA
  - TLS_RSA_WITH_TDES_168_CBC_SHA
  - TLS_DHE_RSA_WITH_TDES_168_CBC_SHA
  - Other CipherSuite options that use FIPS Approved algorithms only

- FCS_COP.1(6): The TSF implements SSH v2.0 as specified in Transport Layer - RFC 4253, User Authentication Layer – RFC 4252, Connection Layer – RFC 4255 (vendor affirmed). The SSH-2 protocol has a internal architecture (defined in RFC 4251) with well-separated layers. The transport layer handles initial key exchange (Diffie-Hellman) and server authentication and sets up encryption (AES), compression and integrity verification (SHA). The user authentication layer handles client authentication and provides a number of authentication methods (password and/or public key RSA method is used by the TOE). The connection layer defines the concept of channels, channel requests and global requests using which SSH services are provided. The following algorithms options may be allowed:
  - Kex_algorithms: diffie-hellman-group1024-exhange-sha256, diffie-hellman-group2048-exhange-sha256, diffie-hellman-group14-exhange-sha1[12]
  - Encryption_algorithms_client_to_server: aes128-cbc, aes192-cbc, aes256-cbc.
  - Encryption_algorithms_server_to_client: aes128-cbc, aes192-cbc, aes256-cbc.
  - Mac_algorithms_client_to_server: sha256
  - Mac_algorithms_server_to_client: sha256
  - Compression_algorithms_client_to_server: none
  - Compression_algorithms_server_to_client: none
  - Other algorithms options that are FIPS Approved algorithms only
- FCS_COP.1(7): The TSF implements IKE and IPSEC (Internet Key Exchange – RFC 2409, Encapsulating Security Payload – RFC 4303) and they are vendor affirmed. The Diffie-Hellman key exchange will use group mode that shall include the private group 2 or 14, or 1024 or 2048-bit MOD P. The authentication will be performed using pre-shared keys (not RSA) and the pre-shared keys will be transmitted using a secure out-of-band method (outside the scope of the evaluation). Only the Encapsulating Security Payload protocol is allowed and all algorithms configured in the one-way security associations must correspondent to FCS_COP.1*. IPSec can be implemented in a host-to-host transport mode, as well as in a network tunnel mode. Both modes are allowed in the CC evaluated configuration.
- FCS_COP_(EXP).1: The TSF performs random number generation via the ANSI X9.31-1998 algorithm, using multiple software-generated and hardware-generated seed values. Physical security is used to prevent tampering of the random number generation/pseudorandom number generation sources.

6.1.3 User Data Protection

The TOE's firewall policy consists of one or more rules that define the source, destination, protocol, and service type for specific traffic and whether the Mobility Controller should permit, deny, or perform other actions on the traffic that matches the rule (stateful inspection type). The TOE's ACLs, on the other hand, provide a common way for restricting certain types of traffic on a physical port (packet-filtering type). Firewall policies[13] differ from ACLs in the following ways:

[12] Group 1 provides 768 bits of keying strength. They are not recommended and should not be used in the CC evaluated configuration.

[13] Since firewall policies provide greater function than standard and extended ACLs, it is recommended to be used instead of ACLs.

- Firewall policies are *stateful*, meaning that they recognize flows in a network and keep track of the state of sessions. For example, if a firewall policy permits ftp traffic from a client, the policy also recognizes that inbound traffic associated with that session should be allowed.
- Firewall policies are *bi-directional*, meaning that they keep track of data connections traveling into or out of the network. ACLs are normally applied to either traffic inbound to an interface or outbound from an interface.
- Firewall policies are *dynamic*, meaning that address information in the policy rules can change as the policies are applied to users. For example, the alias *user* in a policy automatically applies to the IP address assigned to a particular user. ACLs typically require static IP addresses in the rule.

ArubaOS provides the following types of ACLs:

- Standard ACLs permit or deny traffic based on the source IP address of the packet. Standard ACLS can be either named or numbered, with valid numbers in the range of 1-99 and 1300-1399. Standard ACLs use a bitwise mask to specify the portion of the source IP address to be matched.
- Extended ACLs permit or deny traffic based on source or destination IP address, source or destination port number, or IP protocol. Extended ACLs can be named or numbered, with valid numbers in the range 100-199 and 2000-2699.
- MAC ACLs are used to filter traffic on a specific source MAC address or range of MAC addresses. MAC ACLs allow filtering of non-IP traffic and is different from MAC based authentication. MAC ACLs can be either named or numbered, with valid numbers in the range of 700-799 and 1200-1299.
- Ethertype ACLs are used to filter based on the Ethertype field in the frame header. Ethertype ACLs can be either named or numbered, with valid numbers in the range of 200-299.These ACLs can be used to permit IP while blocking other non-IP protocols, such as IPX or AppleTalk.

When creating a firewall policy rule, four fields must be provided: Source, Destination, Service, and Action. For the required fields, the following parameters are provided and listed below:

Field	Parameter	Description
Source/Destination	Any	Acts as a wildcard and applies to any source address.
	User	This refers to traffic from the wireless client. This is mapped to subject security attribute: IP address[14].
	Host	This refers to traffic from a specific host. When this option is chosen, must configure the IP address of the host. This is mapped to subject security attribute: IP address.
	MAC	This refers to a specific source MAC address or range of MAC addresses.
	Network	This refers to a traffic that has a source IP from a subnet of IP addresses. When this option is chosen, must configure the IP address and network mask of the subnet.
Service	Any	This option specifies that this rule applies to any type of traffic.
	TCP	Using this option, you configure a range of TCP port(s) to match for the rule to be applied.
	UDP	Using this option, you configure a range of UDP port(s) to match for the rule to be applied.

[14] The parameter “User” is any IP address in the TOE’s user-table. Parameter “Host” refers to a specific IP address. The TOE maintains a table of all users called “user-table”. A user-table entry is created for a wireless client once it obtains network access and IP traffic. All wireless clients are listed in the user-table.

Field	Parameter	Description
	Service	Using this option, you use one of the pre-defined services (common protocols such as HTTPS, HTTP, and others) as the protocol to match for the rule to be applied.
	Protocol	Using this option, you specify a different layer 4 protocol (other than TCP/UDP) by configuring the IP protocol value.
Action	Permit	Permits traffic matching this rule.
	Drop	Drops packets matching this rule without any notification.
	Reject	Drops the packet and sends an ICMP notification to the traffic source.
	Src-NAT	Performs network address translation (NAT) on packets matching the rule.
	Dst-NAT	Redirects traffic to the configured IP address and destination port.
	Dual-NAT	Performs both source and destination NAT on packets matching the rule.

The following fields are optional and may not be considered security relevant (e.g., for QoS purpose):

Field	Description
Log*	Logs a match to this rule. This is recommended when a rule indicates a security breach, such as a data packet on a policy that is meant only to be used for voice calls.
Mirror	Mirrors session packets to datapath or remote destination.
Queue	The queue in which a packet matching this rule should be placed. For example, select **High** for higher priority data, such as voice, and **Low** for lower priority traffic.
Time Range	Time range for which this rule is applicable.
Blacklist*	Automatically blacklists a client that is the source or destination of traffic matching this rule. This option is recommended for rules that indicate a security breach where the blacklisting option can be used to prevent access to clients that are attempting to breach the security.
TOS	Value of type of service (TOS) bits to be marked in the IP header of a packet matching this rule when it leaves the controller.
802.1p Priority	Value of 802.1p priority bits to be marked in the frame of a packet matching this rule when it leaves the controller.

* - Security-relevant actions

All the information security attribute values are compared against the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator. If there is a match, the TOE will perform the specified action on the packet whose information security attributes match the rule. Firewall policies and ACLs can be applied to user roles to give differential treatment to different users on the same network, or to physical ports to apply the same policy to all traffic through the port.

The administrator can place all authenticated wireless users into a single VLAN or into different VLANs, depending upon the network. A client can be assigned to a VLAN based on user role or authentication method (role will take precedence if there is a conflict). If none is defined, the client is automatically assigned to the default VLAN (referred to as VLAN: 1). The wireless clients can only send information to the VLAN that they are assigned to and only if the port(s) are configured to carry traffic associated with that VLAN(s). If not, the traffic will be dropped by the TOE. By default, a port carries traffic only for the VLAN to which it is assigned. Administrator can optionally configure a port to operate as a trunk port that can carry traffic for multiple VLANs. For a trunk port, administrator can specify whether the port will carry traffic for all VLANs configured on the controller or for specific VLANs. For more information on the VLAN configuration and management, please see section 6.1.5.

The TOE will encrypt user data transmitted to wireless users using authentication 802.1X . The administrator sets the protocol to be used by the TOE. The protocol utilizes AES for encryption. In addition to that, once the wireless user is authenticated to the TOE and is connected to the IP network, the administrator has an option to enforce additional Layer 3 encryption using the IPSec/IKE protocol, which also employs AES or TDES algorithm. The 802.11i protocol may use for encryption the session keys derived by the authentication server using EAP-TLS, EAP-TTLS or PEAP protocols during the wireless user authentication phase.

IPSec/IKE uses session keys which are derived during the IKE handshake using the Diffie-Hellman key agreement algorithm. The IKE handshake uses a pre-shared key set by the administrator. The IPSec/IKE protocol with pre-shared keys represents a probabilistic/permutational security mechanism.

The wireless user may authenticate without using 802.1X. In this case, an IPSec/IKE VPN is set up prior to user authentication. The user then authenticates using a username and password. The authentication and subsequent data traffic are protected by the VPN. As above, IPSec/IKE uses a pre-shared key and represents a probabilistic/permutational security mechanism.

Network packets are received in memory buffers pre-allocated at boot time. The buffers are populated in the network interface receive ring. When the CPU receives network packets from the network interface, the CPU allocates a free buffer from the preallocated pool and replenishes the receive ring. When the CPU has finished packet processing, the CPU adds the memory buffer associated with this network packet to the free buffer pool. Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required.

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_IFC.1(1): The Aruba Information Flow Policy applies to packets traffic to, from, and through the network interface on the TOE. The table above defines the subjects and operation (send) covered by the scope of this requirement.
- FDP_IFF.1(1): The Aruba Information Flow Policy is enforced on information flows matching an access policy defined by the administrator. The access policy may be configured to pass, drop, reject, or perform the specified action on the information matching the policy. The policy may be individually assigned to a user role or a physical port.
- FDP_IFC.1(2): The VLAN Information Flow Policy applies to packets traffic through the network interface on the TOE. The requirement defines the subjects (wireless clients) and operation (send) covered by the scope of this requirement.
- FDP_IFF.1(2): The VLAN Information Flow Policy is enforced on information flows matching the VLANs defined and configured by the administrator. The policy may be configured to pass or drop traffic from clients based on VLANs and ports assigned. The client can be assigned to the VLAN configured based on user role or authentication method.
- FDP_PUD_(EXT).1: The TOE will encrypt user data transmitted to wireless users using authentication 802.11i or IPSec/IKE VPN. All of these protocols utilize AES or TDES for encryption. The FIPS module also supports Bypass mode in which no encryption is employed as part of FIPS mode. Only the Crypto Officer (administrator) can enable Bypass mode in the module.
- FDP_RIP.1: Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required.

6.1.4 Identification and Authentication

The TOE supports role-based authentication. Wireless users (or clients, term used interchangeably) can authenticate to an external authentication server or to the Controller’s internal database. The administrator can create a user account in the internal database and assign a predefined role to that account. When that user logs in to the Controller using the configured username and password, he or she is restricted based on that assigned role. In this case, the authentication mechanism is provided by the TOE and the credentials are maintained in the internal database. The administrator can also configure the TOE so that wireless users are authenticated using an external authentication server[15]. The TOE supports the RADIUS, LDAP, and TACACS+ servers. A trusted channel is established between the TOE and the authentication server. For wireless users using 802.1X authentication, when a user client connects to the TOE, the TOE passes authentication protocol messages between the client and the authentication server, until the user is authenticated, or authentication is denied. As a part of the initial handshake, the authentication server presents to the client a TLS server certificate. Communications between the client and the server are then encrypted by AES or TDES. The following authentication protocols are supported: EAP-TLS, EAP-TTLS, PEAP.

For EAP-TTLS and PEAP protocols, the user will authenticate to the server over a TLS encrypted connection using a username and password. For EAP-TLS, the user will use a TLS client certificate[16] to authenticate. The certificate will contain the username of the user, and may contain other user-specific information. The authentication server will maintain a list of trusted certification authorities to verify the client certificate. If the authentication fails, the authentication server will communicate the authentication failure to the TOE. Otherwise, the authentication server will communicate the authentication success to the TOE and send to the TOE the session key, which was derived during the EAP-TLS/EAP-TTLS/PEAP handshake, as well as the user role attribute. The session key may be used by the TOE to encrypt further communications with a wireless client.

For wireless users using an open system connection with VPN, the IPSec/IKE VPN is established between the TOE and the wireless client prior to the user authentication using pre-shared keys. The user then authenticates to the external authentication server using a username and password. The external authentication server communicates success or failure of the authentication to the TOE. Captive portal allows a wireless user to authenticate using a web-based portal. Captive portals are typically used in public access wireless hotspots or for hotel in-room Internet access. After a user associates to the wireless network, their device is assigned an IP address. The user must start a web browser and pass an authentication check before access to the network is granted. The username/password exchange is encrypted using standard TLS encryption.

MAC address authentication is examining the MAC address of an associated device, comparing it to an internal or external database, and changing the user role to an authenticated state if there is a match. MAC address authentication is not a secure form of authentication as the MAC address of a network interface card (NIC) can be easily changed in software. In addition, the MAC address cannot be associated with an individual user, only a device. Therefore, the use of MAC address authentication is NOT allowed in the CC evaluated mode.

When a wireless user exceeds the configured authentication threshold, the user is automatically blacklisted by the controller, an event is logged, and an SNMP trap is sent (optional SNMP server in the operating environment must be set up to capture SNMP traps). By default, the maximum authentication failure threshold is set to 0 (but can be set as high as 255), which means that there is no limit to the number of times a user can attempt to authenticate. When users are blacklisted because they exceed the authentication failure threshold, they are blacklisted indefinitely by default. Administrator can configure the duration of the blacklisting.

Remote administrators are configured as users who have privileges to access the CLI and Web GUI administration interfaces. Remote administrators are authenticated as users using an external authentication server. Following user authentication, the remote administrators must authenticate further as an administrator in order to perform administrative tasks. A trusted channel is established between the TOE and the authentication server. The remote administrators authenticate as users using a username and password via Web GUI. The Web GUI interface provides a trusted path to connect to the TOE via HTTPS. The HTTPS interface uses a server RSA certificate which is stored on the TOE.

The further authentication as an administrator , which is required to perform administrative tasks, uses an internal database on the TOE, which stores a list of administrators and their passwords[17]. Local administrators are authenticated as administrators using the internal database using a username and password. Wireless users and administrators are identified by using their usernames. Wireless user has no access to the protected network until he/she is authenticated. The administrator has no access to management functions until he or she is authenticated.

The TOE maintains a username and password for each administrator, set for each new user using the *Configuration-> Management-> Administration* panel of the Web GUI, and clicking the Add button. A new password for a

[15] The Controller accepts the user credentials and send the credentials to the authentication server. The wireless clients never communicate directly with the authentication server.

[16] When authentication server is used, the credentials (password or certificate) are maintained outside the TOE.

[17] Administrator passwords cannot contain whitespace or question mark.

particular local administrator can be set by using the *Configuration-> Management-> Administration* panel of the Web GUI, and clicking the Edit button next to the administrator name. A similar functionality is provided by the "mgmt-user" command of the CLI. The TOE maintains remotely authenticated wireless user attributes for the duration of each remotely authenticated user session, including session key and role (username and password may be stored in internal database if TOE authentication is used). The TSF associates the username of a logged on user with the user's sessions and any processes acting on behalf of that user.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1: After an administrator specified numbers of failed attempts, the TOE will lockout (blacklist) the wireless client, log the event, and send a SNMP trap. The duration of the lockout can be configured by administrators. FIA_AFL.1 is enforced even when using external authentication server.
- FIA_ATD.1(1): The TOE's authentication mechanism uses the embedded database (the internal database) to store information about the administrators. The following information is associated with each administrator account: username and password.
- FIA_ATD.1(2): The TOE's authentication mechanism uses the embedded database (the internal database) to store information about the users. The following information is associated with each remotely authenticated user account: session key[18] and role. Note that if wireless users and remote administrators are authenticated using an external authentication server, the credentials are maintained in operating environment.
- FIA_SOS.1: The TOE authentication mechanism provides configuration for minimum password length. The administrator should at a minimum, requires password to be at least 6 characters long. The following calculation is based on the following facts:
  - Password is case-sensitive
  - A-Z, a-z, 0-9, !@#$%^&*()_+, and extended characters
  - Password minimum length is set to 6

  Passwords must be at least six characters long. Numeric, alphabetic (upper and lower case), and keyboard/extended characters can be used, which gives a total of 95 characters to choose from. A six character password using all characters has a total possible combination of 95^6 (=735,091,890,625). The probability for a random attempt to succeed is therefore less than one in 1,000,000.
- FIA_UAU.1: The TOE will not allow the wireless user or the administrator to perform any TSF-mediated actions except identification before the authentication process completes successfully.
- FIA_UAU_(EXT).5: The TOE, at a minimum, provides a local administrator authentication mechanism and a remote user authentication mechanism. The administrator can configure TOE to provide the same or different authentication mechanism (local, remote) for wireless users and administrators. The TOE shall invoke the correct authentication mechanism as configured by the administrator.
- FIA_UID.2: The TOE requires each wireless user or administrator to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
- FIA_USB.1: The TOE associates the username of a logged on user with the user's sessions and any processes acting on behalf of that user. FIA_USB.1 applies to all authenticated users.

6.1.5 Security Management

The TOE provides the administrator role the capability to enable the management of security attributes, TSF data and security functions. The administrator can configure TOE security settings and policies using the Web Graphical User Interface via HTTPS, or the command line interface via serial console locally or remotely using SSH[19]. The

[18] Session keys are not stored in the internal database for security reasons. They are maintained in a table in memory and are associated with the correct remote administrator account.

[19] CLI can also be accessed using Telnet but this is not allowed in the CC evaluated mode. Telnet access is disabled by default.

Web GUI is a just a front-end to the CLI (i.e., calls the CLI internally). It is provided as a user-friendly interface for administrator to manage the TOE. Every function that can be performed on the Web GUI, can also be performed using the CLI but not vice versa. However, every security management function claimed can be done either using the Web GUI or CLI.

The TOE supports role-based authentication. There are three types of roles: administrator[20] role, non-administrator role, and wireless user role. Administrators can manage the TOE using HTTPS Web GUI or command line interface. Wireless clients cannot access the TOE through the Web GUI or CLI interfaces and, therefore, do not have access to the management functionalities of the TOE. Non-administrator role can perform non-security tasks which include the following:

- Read-only role - This role permits access to CLI "show" commands or Web GUI monitoring pages only. It does not allow user to perform any action such as copying files or rebooting the controller.
- Network operations role - This role permits access to Monitoring, Reports, and Events pages in the Web GUI that are useful for monitoring the controller. This role can log into the CLI; however, user can only use a subset of CLI commands to monitor the controller.

The remote administrator or non-administrator authenticates with a username and password to the internal authentication database, via an HTTPS connection. The local administrator or non-administrator authenticates with a username and password to the internal authentication database, via the interactive command line. Once the administrator is authenticated, the TOE provides management interfaces which can be used by the administrator to configure the TOE security functions. Local administrators can also use the CLI via a serial console (direct) connection to the TOE. Remote administrators may use the Web GUI interface from the browser or may also use the CLI interface via an SSH protocol connection from an SSH client. An administrator or non-administrator can log on to the administrative interface (Web GUI or SSH) while connected to the network over wireless.

The TOE provides to the administrator capabilities to modify, set initial value of, or manage the:

a) Audit server IP address, using the *Configuration -> Management->Logging ->Servers* panel of the Web GUI.

b) Authentication server IP address, using the *Security->AAA Servers -> Radius Servers* panel of the Web GUI. Set the authentication failure threshold using *Configuration->Security->Authentication->Profiles* panel of the Web GUI or using CLI "aaa authentication" command.

c) NTP server IP address, using the *Configuration->Switch->General* panel of the Web GUI.

d) SNMP server IP address, using the *Configuration->Management ->SNMP* panel of the Web GUI or CLI "snmp-server" command.

e) TLS Web GUI server certificate used for TOE administration, using the *Wireless Network Management -> Import Certificate For Web Server* panel of the Web GUI.

f) Wireless security protocols, including choosing the 802.11i for wireless data encryption. This functionality is provided by the *WLAN->Network* panel of the Web GUI.

g) VPN server, including setting up IPSec/IKE connections and setting the pre-shared keys. This functionality is provided by the *Security-> VPN Settings -> IPSec* panel of the Web GUI.

h) WIP, including setting signature events (wireless intrusion profiles) and actions to be taken upon detection of a potential security violation using the *Configuration->AP Configuration* and CLI "ids signature-matching-profile" command.

i) Identification and Authentication. This includes configuring administrator usernames and passwords (which is performed by utilizing the *Management->Access Control* panel of the Web GUI), creating a user role and assigning the firewall policies, VLAN ID, and maximum session quota (which is performed by utilizing the *Configuration->Access Control->User Roles* panel of the Web GUI or using CLI "user-role" command), setting the CLI idle timeout for administrators (which is performed utilizing the CLI "loginsession timeout" command) and setting the idle timeout for wireless users (which is configured by

[20] Some Aruba documents may refer to as the "root" and/or "crypto officer" role.

utilizing the *Security->AAA Servers-> Internal Servers-> General* panel of the Web GUI for local administrators and the *Security->AAA Servers* panel of the Web GUI or the CLI “config aaa timers” command).

j) Auditing. This includes starting and stopping of audit event logging, setting the IP address of the audit server and setting the types of audit events to selectively log. The configuration is performed by utilizing the *Management->Logging* panel of the Web GUI.

k) Policies. This includes creating and configuring the policies using the *Configuration->Security->Access Control->Policies* panel of the Web GUI or using CLI “ip access-list” command. By default, if no policies are defined, information is not allowed to flow.

l) Blacklist Duration. This includes configuring the blacklist duration using the *Configuration->Wireless->AP Configuration* panel of the Web GUI or using CLI “wlan virtual-ap auth-failure-blacklist-time <seconds> blacklist-time <seconds>” command.

m) Users Management. This includes managing users and administrators account maintained in the internal database using *Configuration -> Management -> Administration* panel of the Web GUI or using CLI “mgmt-user” command.

n) FIPS mode on Controller, using the *Configuration->Network->Controller ->System Setting* panel of the Web GUI. Check the “FIPS Mode for Mobility Controller Enable” box.

o) FIPS mode on AP, using *Configuration -> Wireless -> AP Configuration -> AP* Group panel of the Web GUI. Find the appropriate AP group and select the AP system profile. Check the “Fips Enable” box.

p) VLAN Configuration. This includes creating a new VLAN and configuring ports to that VLAN using *Configuration > Network > VLANs* and *Configuration > Network > Ports* using the Web GUI or using CLI “interface vlan <id>” and “switchport trunk allowed vlan <id>,<id>” commands.

Note that the list above is not meant to be all-inclusive (for example, administrator can also set the audit server IP address using command “logging <ipaddr>” or set logging level using command “logging level <level> <category> [subcat <subcategory>]” as well). For more information about the management interfaces, please refer to the ArubaOS User Guide documentation. To found out more information about the cryptographic functionalities, please refer to the FIPS 140-2 Security Policies.

The security attributes are cryptographic keys. The keys are considered secure if they are generated, distributed and managed by a FIPS 140-2 approved cryptographic module. In particular, running the TOE in FIPS mode of operation enforces this requirement.

The Security Management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1(1): The TOE provides and restricts the capability to manage the cryptographic functions identified in FMT_MOF.1(1). The cryptographic keys and algorithms will be managed by the administrator as prescribed in the FIPS 140-2 standard. The Cryptographic Module Security Policy for the Aruba Mobility Controllers and APs (certificates #1297, #1116, #1109, #1077, and #1075) provides more information on cryptographic functions managements.
- FMT_MOF.1(2): The TOE provides and restricts the capability to manage the security audit functions identified in FMT_MOF.1(2).
- FMT_MOF.1(3): The TOE provides and restricts the capability to manage the authentication functions identified in FMT_MOF.1(3).
- FMT_MOF.1(4): The TOE provides and restricts the capability to manage the wireless intrusion protection functions identified in FMT_MOF.1(4).
- FMT_MSA.1(1): The TOE provides and restricts the capability to manage the information security attributes and operations.
- FMT_MSA.1(2): The TOE provides and restricts the capability to manage the security attributes.
- FMT_MSA.2: The TOE only utilizes FIPS-Approved functions to generate, distribute, and mange cryptographic keys. The TOE has been FIPS-certified which means that self-tests have been implemented

to verify the correct operations of the algorithms and keys generated. Self-tests include known answer test, software integrity test, pair-wise consistency test, continuous random number generation test, etc.

- FMT_MSA.3(1): By default, all information flow are denied unless explicitly allowed by administrator.
- FMT_MSA.3(2): By default, if no VLAN is configured or defined then all wireless clients will be assigned to the default VLAN.
- FMT_MTD.1(1): The TOE provides and restricts the capability to manage the selection of audit events.
- FMT_MTD.1(2): The TOE provides and restricts the capability to modify the authentication credentials and user identification credentials. The TOE provides the administrator capabilities to query, modify, delete, clear and create administrators, including usernames and passwords (stored in internal database), using the *Configuration -> Management -> Administration* panel of the Web GUI or CLI.
- FMT_MTD.1(3): The TOE provides and restricts the capability to modify the user authentication credentials to administrators. Users do not have access to the management interface and cannot change their passwords.
- FMT_MTD.1(4): The TOE provides and restricts the capability to manage the firewall policies, VLANS, and ACLs rulesets.
- FMT_MTD.1(5): The TOE provides and restricts the capability to modify the maximum session quota per user setting.
- FMT_SMF.1(1): The TOE provides interfaces to manage the cryptographic function.
- FMT_SMF.1(2): The TOE provides interfaces to manage the audit generation function.
- FMT_SMF.1(3): The TOE provides interfaces to manage the cryptographic key functions.
- FMT_SMF.1(4): The TOE provides interfaces to manage the Wireless Intrusion Protection function.
- FMT_SMF.1(5): The TOE provides interfaces to manage the identification and authentication credentials of user and administrator.
- FMT_SMF.1(6): The TOE provides interfaces to manage the Policy, VLANs, and ACL rulesets.
- FMT_SMF.1(7): The TOE provides interfaces to manage the maximum session quota per user setting.
- FMT_SMR.1: The TOE supports role-based authentication. There are two types of roles: administrator role and wireless user role.

6.1.6 Protection of the TSF

The Mobility Controller has an internal hardware clock that provides reliable time stamps used for auditing. The internal clock is synchronized with a time signal obtained from an external NTP server. The Mobility Controller and AP runs a suite of self tests during power-up which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. An administrator can choose to reboot the TOE to perform power-up self test. The Mobility Controller and AP runs the suite of FIPS 140-2 validated cryptographic module self tests during start-up or on request from the administrator, including immediately after generation of a key (FIPS self-tests, including the continuous RNG test).

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_STM_(EXT).1: The TOE provides its own time and/or relies on an external trusted time server for this function.
- FPT_TST_(EXT).1: The TOE offers a suite of self-tests to verify the correct operation of the TSF and integrity of TSF executable.
- FPT_TST.1(1), (2): The TOE offers a suite of self-tests for administrator (in FIPS terminology, the Crypto Officer which is mapped to the administrator role) to verify the correct operation of the key generation and static TSF cryptographic data.

6.1.7 Resource Utilization

The TOE can create a user role and assign several attributes to that role such as firewall policies and maximum sessions. The 'Max Sessions' field configures a maximum number of sessions per role as a whole. The default value is 65535. Administrators can configure any value between 0-65535.

The Resource Utilization function is designed to satisfy the following security functional requirement:

> FRU_RSA.1: The TOE enforces the maximum quota on number of sessions a role can have opened simultaneously. A user session is identified by a 5-tuple – Source IP, Destination IP, Protocol, Source Port, Destination Port. Each user is allocated a role and each role in turn has a maximum number of permitted sessions. As long as the 5-tuple doesn't change, the session remains the same and doesn't have to be reauthenticated once the client has been successfully authenticated and assigned a role. When a wireless client moves its association from one AP to another AP, none of the 5-tuple changes, hence no new user session is generated—the user simply roams to the other AP.

6.1.8 TOE Access

The TOE terminates a wireless user session or an administrator CLI session (for a local or remote administrator) after the inactivity time exceeds a configurable session idle timeout. The session idle timeout is the maximum amount of time a wireless user or an administrator may remain idle. The TOE terminates an administrator Web GUI session for a remote administrator after the inactivity time exceeds a session idle timeout of 15 minutes. Timeout does not apply to the following screens below which are auto refreshed. These screens are used for monitoring purposes only and do not provide any security management interface. The information on these screens are updated constantly; therefore, the screens are never idle (timeout does not apply).

- *Monitoring > Network > All Access Points*
- *Monitoring > Network > All Air Monitor*
- *Monitoring > Network > Wired Access Points*
- *Monitoring > Network > All WLAN Clients*
- *Monitoring > Controller > Access Points*
- *Monitoring > Controller > Air Monitor*
- *Monitoring > Controller > Wired Access Points*
- *Monitoring > Controller > Clients*
- *Monitoring > WLAN > [ESSID_NAME] > Access Points*
- *Monitoring > WLAN > [ESSID_NAME] > Clients*
- *Monitoring > Debug > Local Clients*
- *Monitoring > Debug > Process Logs*
- *Maintenance > WLAN > Program AP*
- *Maintenance > WLAN > Reboot AP*

The TOE assesses wireless user inactivity as the cessation of network traffic arriving from the wireless client. It should be noted that processes acting on behalf of the user may send protocol network packets to the mobility controller, even when the user is not interacting directly, e.g. pressing keys.

To change the administrator CLI session idle timeout, the administrator can use the "loginsession timeout" command of the CLI. The default value of the administrator session idle timeout is 15 minutes.

The TOE Access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: By default, the TOE will terminate inactive user session after 15 minutes and require users to login again. The timeout period can be changed only by administrator. Management sessions through the serial port, SSH, or the Web GUI will time out after 15 minutes.
- FTA_TAB.1: The TSF displays an advisory warning banner regarding use of the TOE prior to establishing an administrator session. The administrator can configure the warning message displayed in the banner.

6.1.9 Trusted Path/Channels

The TOE provides trusted paths for remote administrator authentication as a wired user and for wireless user authentication using an open system connection. The TOE provides a trusted channel between itself and the operating environment authentication, audit and NTP servers.

For remote administrators, the TOE provides a TLS based trusted path from the TOE to the remote administrators for authentication as users to the external authentication server. SSH[21] is also used to provide secure remote command line administration interface.

For wireless users using an open system connection, the TOE provides an IPSec/IKE VPN trusted path from the TOE to the wireless users for authentication of the wireless users. There is no certificate involved with IPSec/IKE as pre-shared key is used instead. A pre-shared key is transmitted using a out-of-band method and is the basic for initial authentication. The user then authenticates to the external authentication server using a username and password (see section 6.1.4).

The TOE uses the IPSec/IKE protocol with pre-shared keys to establish a trusted channel between itself and the external authentication, logging, and NTP servers. To configure the channels the administrator uses the *Security -> VPN Settings -> IPSec* panel of the Web GUI to create the host-to-host IPSec/IKE connections. The administrator then sets a pre-shared IPSec/IKE key for each IPSec/IKE connection using the *Security -> VPN Settings -> IPSec -> Add IKE Secret* panel of the Web GUI. All configuration settings must specify FIPS-certified AES/TDES and SHA-256/SHA-1 algorithms as specified by the FCP_COP.1 requirements.

The Trusted Path/Channels function is designed to satisfy the following security functional requirements:

- FTP_ITC_(EXT).1: The TOE uses the IPSec/IKE protocol with pre-shared keys to establish a trusted channel between itself and the external authentication, logging, and NTP servers.
- FTP_TRP.1: The IPSec/IKE protocol is also used by the TOE to secure network traffic between the TOE and wireless network clients, providing a trusted path for wireless user authentication in the case 802.11i user authentication is not used. If 802.11i authentication is used, it will secure the network traffic to and from wireless users at Layer 2. SSH or TLS is also used to provide secure remote command line administration interface.

Security Function	Mobility Controller	Access Point
Security Audit	Generate audit events and implement all other audit functions. Perform the wireless intrusion detection functionality and generate alarms.	Generate audit events and perform wireless intrusion detection functionality and generate alarms.
User Data Protection	Perform all access control decisions and enforcement.	None
Cryptographic Support	Provide the use of only FIPS-certified algorithms.	Provide the use of only FIPS-certified algorithms.

[21] The correct implementations of SSH, TLS, and IPSec/IKE will be vendor affirmed.

Security Function	Mobility Controller	Access Point
Identification & Authentication	Perform all identification and authentication functionality including lockout, password length enforcement, storage of credentials, etc.	None
Security Management	Provide all security management functionality	None
Protection of the TSF	Provide FIPS self-tests and timestamp for auditing	Provide FIPS self-tests
Resource Utilization	Implement the quota on sessions	None
TOE Access	Implement the session timeout and the warning banner	None
Trusted Paths/Channels	Provide TLS and SSH for trusted servers and remote administrator	None

6.2 RFC Conformance Statements

This section identifies, for the critical RFCs, the options supported by the TOE.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
EAP-TLS	RFC 2716	**TLS Version:** The TOE offers TLS v1.0 for the EAP TLS Authentication Protocol. **Conversation Restarts:** The TOE implements a restart capability and imposes a limit on the number of restarts. The TOE blacklists a client after a configurable number of authentication failures. **Identity Verification:** The TOE verifies that a claimed identity corresponds to the certificate presented by the client. The TOE uses the CN in the client certificate as the userID to authenticate.
EAP-TTLS	RFC 5281	The TOE does not implement an EAP-TTLS authentication server—it simply acts as a pass thru authenticator for all EAP-TTLS traffic between the supplicant and the authentication server in the operational environment.
HTTP over TLS	RFC 2818	**Connection Closure:** the TOE always initiates a close alert before closing a connection. The TOE does not perform an incomplete close and waits for the peer's closure alert.
IPsec/IKE	RFC 2408/ RFC 2409	**General Message Processing:** The TOE ensures successive retransmissions of the same packet are separated by increasingly longer time intervals. **ISAKMP Header Format:** The TOE will not accept ISAKMP packets with a version number (comprising Major.Minor) in advance of its own. The TOE sets the encryption flag of ISAKMP header to 1in an outbound ISAKMP packet. **Security Association Establishment:** The TOE retains the Transform Number field in the Transform Payload in an outbound ISAKMP packet. **Security Association Modification:** The TOE will continue to support incoming traffic on an old SA until traffic is received on a newly created SA.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
IPsec/ESP	RFC 4303	**Modes:** The TOE supports both transport and tunnel modes for ESP. **Services:** The TOE does not support confidentiality-only or integrity-only services—only the 'confidentiality and integrity' service is supported, in order to comply with FIPS requirements.
RADIUS	RFC 2138	The TOE supports challenge/response, and sends a new Access-Request in response to a valid Access-Challenge.
	RFC 2869	**EAP-Message:** The TOE silently discards Access-* messages without a Message-Authenticator attribute.
	RFC 3579	**Conflicting Messages:** The TOE sends EAP-success or EAP-failure to the client depending on if it receives Access-Accept or Access-Reject from the RADIUS server respectively. The Access-Accept packets have only one EAP-Message attribute in them, containing EAP Success; and Access-Reject packets have only one EAP-Message attribute in them, containing EAP Failure. **EAP-Message:** The TOE silently discards Access-* messages without a Message-Authenticator attribute. **Security Protocol/Security Issues:** The TOE supports IPSec requirements specified in the appropriate RFCs. It supports IPSec ESP to protect RADIUS frames between the TOE and RADIUS server with specified algorithms and IKE for key negotiation.
	RFC 3580	The TOE supports IPSec requirements specified in the appropriate RFCs. It supports IPSec ESP to protect RADIUS frames between the TOE and RADIUS server with specified algorithms and IKE for key negotiation.
SSH	RFC 4251	**Host Keys:** The TOE has one RSA and one DSA Host Key for SSH v2, which are generated on initial setup of the TOE. These keys are not shared with any other host and are unique to each TOE instance. The TOE presents the client with its host key fingerprint when the client is connecting to the TOE for the first time. When a client connects to the TOE for the first time, the TOE prompts the user to accept or deny the TOE's host key and hence connect successfully or disconnect. **Policy Issues:** The TOE has an uneditable policy specifying its supported encryption, integrity, and compression algorithms, as listed in section 6.1.2. It implements all mandatory algorithms and methods. The TOE can be configured to accept public key based authentication and/or password based authentication per admin user. The TOE does not allow port forwarding and sessions to clients. The TOE has no X11 libraries or applications and does not support X11 forwarding. **Confidentiality:** The TOE does not accept the "none" cipher. **Data Integrity:** The TOE does not accept the "none" MAC. Note the TOE does not have RekeyLimit set in its configuration and hence does not initiate rekeying. However, client initiated rekeying is processed. **Denial of Service:** Once the SSH connection is terminated, the TOE does not pro-actively try to re-establish it. The TOE can be configured with ACLs to control the clients that are able to connect to it via SSH. **Ordering of Key Exchange Methods:** The TOE orders key exchange algorithms by strength. **Debug Messages:** The TOE requires debug messages to be turned on explicitly and the audit trail reflects the same.

Protocol	RFCs	TOE Handling of Security-Related Protocol Options
SSH	RFC 4251	**End Point Security:** The TOE does not permit port forwarding. SSH is a CLI mechanism for TOE administrators and hence the TOE places no restrictions on user actions once authenticated. **Proxy Forwarding:** The TOE does not support proxy forwarding. **X11 Forwarding:** The TOE does not support X11 forwarding.
	RFC 4252	**Authentication Protocol:** The TOE does not accept "none" authentication method. The TOE disconnects a client after 30 seconds if authentication has not been completed. The TOE also allows authentication retries of 6 times before sending a disconnect to the client. **Authentication Requests:** The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed. The TOE just sends back a disconnect as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies "none" authentication method and replies with a list of permitted authentication methods. **Public Key Authentication Method:** The TOE allows "publickey" authentication. Authentication succeeds if the correct private key is used. Note the TOE does not require multiple authentications. **Password Authentication Method:** The TOE supports password authentication. The TOE does not allow an expired password to be used for authentication. Note the TOE does not support changing user passwords in the SSH session. **Host-Based Authentication:** The TOE does not support host-based authentication.
	RFC 4253	**Encryption:** The TOE offers only aes128-cbc,3des-cbc,aes192-cbc and aes256-cbc for encryption of SSH sessions. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the "none" algorithm for encryption. **Data Integrity:** The TOE permits negotiation of MAC algorithms in each direction. **Key Re-Exchange:** The TOE performs a re-exchange when SSH_MSG_KEXINIT is received. However, it does not initiate a key re-exchange itself.

7. Protection Profile Claims

The TOE conforms to the U.S. Government Wireless Local Area Network (WLAN) Access System Protection Profile For Basic Robustness Environments, Version 1.1, National Security Agency, 25 July 2007.

This Security Target includes all of the assumptions, threats, and policy statements described in the PP, verbatim.

This Security Target includes all of the Security Objectives from the PP, verbatim.

This ST has an additional Security Objective, O.INTRUSION, which is implemented by the TOE's Wireless Intrusion Protection features. This added Security Objective does not affect the PP conformance because it requires additional security functionality used to protect the WLAN. Thus, the ST statement of TOE security objectives is stricter than the PP statement of TOE security objectives.

This Security Target includes all of the Security Functional and Security Assurance Requirements from the PP verbatim, except as noted below:

> The St Author has added sixteen new security functional requirements (as required by CCEVS policy #13) and modified three existing PP security functional requirements. The table below identifies which requirements were added ("**Addition**") or refined ("**Refinement**"), which requirements were changed ("PP Tailored"), and which requirements were copied word for word ("PP Verbatim"). The rationale for any addition or modification is provided in column 3.

Requirement Component	PP Conformance	Addition/Modification Rationale
FAU_ARP.1: Security Alarms	**Addition**	Added because the TOE has the capability to detect wireless intrusion based on signature and perform the specified action.
FAU_GEN.1: Audit Data Generation	PP Tailored	Change the wording in the two completed assignment operations to reference "Table 2". Add the required auditable events for the 'minimum' level for the newly added security functional requirements. FTP_TRP.1 incorrectly references trusted channel instead of trusted path. This is corrected in table 2. A correction to the audit event for FCS_CKM_(EXT).2 was also made
FAU_GEN.2: User Identity Association	PP Verbatim	Not applicable
FAU_SAA.3: Simple Attack Heuristics	**Addition**	Added because the TOE has signature-based intrusion detection capability.
FAU_SEL.1: Selective Audit	PP Verbatim	Not applicable
FCS_BCM_(EXT).1: Extended – Baseline Cryptographic Module	**Refinement**	This SFR has been refined based on PD-0164, "WLAN PP Places FIPS 140-2 section level Requirements on Crypto Module that are not always attainable."
FCS_CKM.1(1): Cryptographic Key Generation	PP Verbatim	Not applicable
FCS_CKM.1(2): Cryptographic Key Generation	PP Verbatim	Complete the required assignment operation.

Requirement Component	PP Conformance	Addition/Modification Rationale
FCS_CKM.2: Cryptographic Key Distribution	PP Verbatim	Not applicable
FCS_CKM_(EXT).2: Extended – Cryptographic Key Handling and Storage	PP Verbatim	Not applicable
FCS_CKM.4: Cryptographic Key Destruction	PP Verbatim	Not applicable
FCS_COP.1(1): Cryptographic Operation (Data encryption/decryption)	**Refinement**	Complete the required assignment and selection operations and added FIPS-approved TDES. The FCS requirements copied directly from the Medium Robustness PP were considered too restrictive for a Basic Robustness PP. CCEVS allows for exception to only requiring FIPS-approved AES for remote administration only.
FCS_COP.1(2): Cryptographic Operation (Digital Signature)	**Refinement**	Complete the required selection operation and added 1024 bits. The FCS requirements copied directly from the Medium Robustness PP were considered too restrictive for a Basic Robustness PP. CCEVS allows for exception of 1024 bits.
FCS_COP.1(3): Cryptographic Operation (Hashing)	**Refinement**	Complete the required selection operation and added FIPS-Approved SHA-1. The FCS requirements copied directly from the Medium Robustness PP were considered too restrictive for a Basic Robustness PP. CCEVS allows for exception to only requiring FIPS-approved SHA-256 for remote administration only.
FCS_COP.1(4): Cryptographic Operation (Key Agreement)	**Refinement**	Complete the required selection and assignment operations and added 1024 bits. The FCS requirements copied directly from the Medium Robustness PP were considered too restrictive for a Basic Robustness PP. CCEVS allows for exception of 1024 bits.
FCS_COP.1(5); Cryptographic Operation (for TLS Protocol)	**Addition**	Added because the TOE has trusted path capability.
FCS_COP.1(6): Cryptographic Operation (for SSH Protocol)	**Addition**	Added because the TOE has trusted path capability.
FCS_COP.1(7): Cryptographic Operation (for IKE/IPSec Protocol)	**Addition**	Added because the TOE has trusted path capability.
FCS_COP_(EXT).1: Extended – Random Number Generation	PP Verbatim	Complete the required assignment operation.
FDP_IFC.1(1): Subset Information Flow Control	**Addition**	Added because the TOE has information flow control capability.

Requirement Component	PP Conformance	Addition/Modification Rationale
FDP_IFF.1(1): Simple Security Attributes	**Addition**	Added because the TOE has information flow control capability.
FDP_IFC.1(2): Subset Information Flow Control	**Addition**	Added because the TOE has information flow control capability.
FDP_IFF.1(2): Simple Security Attributes	**Addition**	Added because the TOE has information flow control capability.
FDP_PUD_(EXT).1: Extended – Protection of User Data	PP Tailored	The FCS_COP_(EXT).2 reference is incorrect (does not exist). It should be FCS_COP.1(1) which specifies the encryption algorithm. Requirement is corrected to reference FCS_COP.1(1).
FDP_RIP.1: Subset Residual Information Protection	PP Verbatim	Complete the required selection operation.
FIA_AFL.1: Authentication Failure Handling	**Refinement**	Update the format to be consistent with CC Version 3.1 Revision 2. Add selection operation to be consistent with CC Version 3.1 Revision 2.
FIA_ATD.1(1): Administrator Attribute Definition	PP Verbatim	Complete the required assignment operation.
FIA_ATD.1(2): User Attribute Definition	PP Verbatim	Complete the required assignment operation.
FIA_SOS.1: Verification of Secrets	**Addition**	Added because the TOE can enforce the minimum password length on passwords.
FIA_UAU.1: Timing of Authentication	PP Verbatim	Complete the required assignment operation.
FIA_UAU_(EXT).5: Extended – Multiple Authentication Mechanisms	PP Verbatim	Not applicable
FIA_UID.2: User Identification Before Any Action	PP Verbatim	Not applicable
FIA_USB.1: User-subject Binding	PP Verbatim	Complete the required assignment operation.
FMT_MOF.1(1): Management of Cryptographic Security Functions Behavior	PP Verbatim	Not applicable
FMT_MOF.1(2): Management of Audit Security Functions Behavior	PP Verbatim	Not applicable

Requirement Component	PP Conformance	Addition/Modification Rationale
FMT_MOF.1(3): Management of Authentication Security Functions Behavior	PP Verbatim	Not applicable
FMT_MOF.1(4): Management of Wireless Intrusion Protection Security Functions Behavior	**Addition**	Added because the TOE restricts the interface to manage the signature detection and response mechanism.
FMT_MSA.1(1): Management of Security Attributes	**Addition**	Added because of dependency on FMT_MSA.3(1).
FMT_MSA.1(2): Management of Security Attributes	**Addition**	Added because of dependency on FMT_MSA.3(2).
FMT_MSA.2: Secure Security Attributes	PP Verbatim	Not applicable
FMT_MSA.3(1): Static Attribute Initialization	**Addition**	Added because of dependency on FDP_IFF.1(1).
FMT_MSA.3(2): Static Attribute Initialization	**Addition**	Added because of dependency on FDP_IFF.1(2).
FMT_MTD.1(1): Management of Audit Data	PP Verbatim	Not applicable
FMT_MTD.1(2): Management of Authentication Data (Administrator)	PP Verbatim	Not applicable
FMT_MTD.1(3): Management of Authentication Data (User)	**Refinement**	The ST refines this requirement to make it stricter. In the PP, users are allowed to change their passwords. The refinement removed this capability so that only administrators can change user passwords.
FMT_MTD.1(4): Management of TSF Data (Policy/ACL Rulesets)	**Addition**	Added because the TOE restricts the capability to manage the policies and ACLs rules.
FMT_MTD.1(5): Management of TSF Data (Sessions Quota)	**Addition**	Added because the TOE restricts the capability to manage the maximum number of sessions a user can have concurrently opened (also see FRU_RSA.1).
FMT_SMF.1(1): Specification of Management Functions (Cryptographic Functions)	PP Tailored	The FCS_COP_(EXT).2 reference is incorrect (does not exist). It should be FCS_COP.1(1) which specifies the encryption algorithm. Requirement is corrected to reference FCS_COP.1(1).

Requirement Component	PP Conformance	Addition/Modification Rationale
FMT_SMF.1(2): Specification of Management Functions (TOE Audit Record Generation)	PP Verbatim	Not applicable
FMT_SMF.1(3): Specification of Management Functions (Cryptographic Key Data)	**Refinement**	This refinement is needed as specified in PD-0145. Basically, in FIPS mode, the self-test cannot be disabled.
FMT_SMF.1(4): Specification of Management Functions (Wireless Intrusion Protection)	**Addition**	Added because this management function exists (also see FAU_ARP.1 and FAU_SAA.3).
FMT_SMF.1(5): Specification of Management Functions (TOE Authentication Data)	**Addition**	Added because this management function exists (also see FMT_MTD.1(2) and FMT_MTD.1(3)).
FMT_SMF.1(6): Specification of Management Functions (Policy/ACL Rulesets)	**Addition**	Added because this management function exists (also see FDP_IFC.1* and FDP_IFF.1*).
FMT_SMF.1(7): Specification of Management Functions (Sessions Quota)	**Addition**	Added because this management function exists (also see FRU_RSA.1).
FMT_SMR.1: Security Roles	PP Verbatim	Not applicable
FPT_STM_(EXT).1: Extended – Reliable Time Stamps	PP Verbatim	Not applicable
FPT_TST_(EXT).1: Extended – TSF Testing	PP Verbatim	Not applicable
FPT_TST.1(1): TSF Testing of Cryptographic Modules	**Refinement**	Remove the term "cryptographic" because FMT_SMR.1 does not identify such role. This is an inconsistency problem in the PP.
FPT_TST.1(2): TSF Testing of Cryptographic Key Generation	**Refinement**	Remove the term "cryptographic" because FMT_SMR.1 does not identify such role. This is an inconsistency problem in the PP.
FRU_RSA.1: Maximum Quotas	**Addition**	Added because the TOE can enforce the maximum number of session a role can have opened concurrently.
FTA_SSL.3: TSF-initiated Termination	PP Verbatim	Not applicable

Requirement Component	PP Conformance	Addition/Modification Rationale
FTA_TAB.1: Default TOE Access Banners	PP Verbatim	Not applicable
FTP_ITC_(EXT).1: Extended – Inter-TSF Trusted Channel	PP Verbatim	Not applicable
FTP_TRP.1: Trusted Path	PP Tailored	Note: Selection operations are not performed in the PP but are reproduced in ST. Update the format to be consistent with CC version 3.1 Revision 2. Add selection operation to be consistent with CC Version 3.1 Revision 2.

This ST does not include the SFRs for the operating environment as it is not required in CC version 3.1. However, the operating environment SFRs are identified and described in section 5.2 of the PP. Any security requirement required in the operating environment is covered by the Security Objective in the environment as specified in the PP. Please refer to the PP for more information.

The Protection Profile specifies EAL2 augmented with, ALC_FLR.2 (Flaw Remediation) but the ST claims a higher assurance level of EAL4 augmented with, ALC_FLR.2 (Flaw Remediation). The table below shows the difference between the assurance requirement levels. See section 8.3 for the rationale for selecting a higher evaluation assurance level.

Assurance Class	EAL2 Assurance Requirement Components	EAL4 Assurance Requirement Components
ADV: Development	ADV_ARC.1	ADV_ARC.1
	ADV_FSP.2	ADV_FSP.4
	ADV_TDS.1	ADV_TDS.3
		ADV_IMP.1
AGD: Guidance Documents	AGD_OPE.1	AGD_OPE.1
	AGD_PRE.1	AGD_PRE.1
ALC: Life Cycle Support	ALC_CMC.2	ALC_CMC.4
	ALC_CMS.2	ALC_CMS.4
	ALC_DEL.1	ALC_DEL.1
	ALC_FLR.2	ALC_FLR.2
		ALC_DVS.1
		ALC_LCD.1
		ALC_TAT.1
ATE: Tests	ATE_COV.1	ATE_COV.2
		ATE_DPT.2
	ATE_FUN.1	ATE_FUN.1
	ATE_IND.2	ATE_IND.2
AVA: Vulnerability Assessment	AVA_VAN.2	AVA_VAN.3

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies;
- TOE Summary Specification; and
- PP Claims

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	P.ACCESS_BANNER	P.ACCOUNTABILITY	P.CRYPTOGRAPHY	P.CRYPTOGRAPHY_VALIDATED	P.ENCRYPTED_CHANNEL	P.NO_AD_HOC_NETWORKS	T.ACCIDENTAL_ADMIN_ERROR	T.ACCIDENTAL_CRYPTO_COMPROMISE	T.MASQUERADE	T.POOR_DESIGN	T.POOT_IMPLEMENTATION	T.POOR_TEST	T.RESIDUAL_DATA	T.TSF_COMPROMISE	T.UNATTENDED_SESSION	T.UNAUTHORIZED_ACCESS	T.UNAUTH_ADMIN_ACCESS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.TOE_NO_BYPASS
O.AUDIT_GENERATION		X																			
O.CORRECT_TSF_OPERATION												X									
O.CRYPTOGRAPHY			X	X	X																
O.CRYPTOGRAPHY_VALIDATED				X	X																
O.DISPLAY_BANNER	X																				
O.INTRUSION																X					
O.MANAGE		X					X							X		X	X				
O.MEDIATE					X	X										X					
O.RESIDUAL_INFORMATION			X					X					X	X							
O.SELF_PROTECTION								X						X		X					
O.TIME_STAMPS		X																			
O.TOE_ACCESS		X							X						X	X	X				
O.ADMIN_GUIDANCE							X										X				
O.CONFIGURATION_IDENTIFICATION										X	X										
O.DOCUMENTED_DESIGN										X											
O.PARTIAL_FUNCTIONAL_TESTING											X										
O.VULNERABILITY_ANALYSIS										X	X										
OE.AUDIT_PROTECTION		X																			
OE.AUDIT_REVIEW		X																			
OE.MANAGE		X												X		X	X				
OE.NO_EVIL							X										X	X			
OE.NO_GENERAL_PURPOSE							X												X		
OE.PHYSICAL																				X	
OE.PROTECT_MGMT_COMMS					X																
OE.RESIDUAL_INFORMATION								X					X	X							
OE.SELF_PROTECTION								X						X		X					
OE.TIME_STAMPS		X																			
OE.TOE_ACCESS		X							X							X	X				
OE.TOE_NO_BYPASS						X			X							X					X

Table 6 Environment to Objective Correspondence

8.1.1.1 P.ACCESS_BANNER

The TOE shall display an initial banner for administrator logins describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.

This Organizational Policy is satisfied by ensuring that:

- O.DISPLAY_BANNER: satisfies this policy by ensuring that the TOE displays an administrator configurable banner that provides all users with a warning about unauthorized use of the TOE. A banner will be presented for all TOE services that allow direct access to the TOE. In other words, it will be required for all administrative actions.

8.1.1.2 P.ACCOUNTABILITY

The authorized users of the TOE shall be held accountable for their actions within the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.AUDIT_GENERATION: addresses this policy by providing the Administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).
- OE.AUDIT_PROTECTION: provides protected storage of TOE and IT environment audit data in the environment.
- O.MANAGE: ensures that access to administrative functions and management of TSF data is restricted to the administrator.
- OE.AUDIT_REVIEW: Further supports accountability by providing mechanisms for viewing and sorting the audit logs
- OE.MANAGE: ensures that the administrator can manage audit functionality in the TOE IT environment.
- O.TIME_STAMPS: plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.
- OE.TIME_STAMPS: ensures that the TOE IT environment provides time services.
- O.TOE_ACCESS: support this policy by controlling logical access to the TOE and its resources. This objective ensures that users are identified and authenticated so that their actions may be tracked by the administrator.
- OE.TOE_ACCESS: See previous (O.TOE_ACCESS)

8.1.1.3 P.CRYPTOGRAPHY

The TOE shall provide cryptographic functions for its own use, including encryption/decryption operations.

This Organizational Policy is satisfied by ensuring that:

- O.CRYPTOGRAPHY: satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE.

- O.RESIDUAL_INFORMATION: satisfies this policy by ensuring that cryptographic data are cleared according to FIPS 140-1/2.

8.1.1.4 P.CRYPTOGRAPHY_VALIDATED

Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).

This Organizational Policy is satisfied by ensuring that:

- O.CRYPTOGRAPHY: satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE.
- O.CRYPTOGRAPHY_VALIDATED: satisfies this policy by requiring that all cryptomodules for cryptographic services be NIST 140-1/2 validated. This will provide assurance that the NIST-approved security functions and random number generation will be in accordance with NIST and validated according the FIPS 140-1/2.

8.1.1.5 P.ENCRYPTED_CHANNEL

The TOE shall provide the capability to encrypt/decrypt wireless network traffic between the TOE and those wireless clients that are authorized to join the network.

This Organizational Policy is satisfied by ensuring that:

- O.CRYPTOGRAPHY: satisfy this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to wireless clients that are authorized to join the network.
- O.CRYPTOGRAPHY_VALIDATED: See previous (O.CRYPTOGRAPHY).
- O.MEDIATE: further allows the TOE administrator to set a policy to encrypt all wireless traffic.
- OE.PROTECT_MGMT_COMMS: provides that the audit records, remote network management information and authentication data will be protected by means of a protected channel in the environment.

8.1.1.6 P.NO_AD_HOC_NETWORKS

In accordance with the DOD Wireless Policy, there will be no ad hoc 802.11 or 802.15 networks allowed.

This Organizational Policy is satisfied by ensuring that:

- O.MEDIATE: works to support this policy by ensuring that all network packets that flow through the TOE are subject to the information flow policies.
- OE.TOE_NO_BYPASS: supports this policy by ensuring that wireless clients must be configured to use the wireless access system for all information flowing between a wireless client and any other host on the network. If the clients are properly configured, any information passing through the TOE will be inspected to ensure it is authorized by TOE polices.

8.1.1.7 T.ACCIDENTAL_ADMIN_ERROR

An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is satisfied by ensuring that:

- O.ADMIN_GUIDANCE: helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.

- O.MANAGE: also contributes to mitigating this threat by providing administrators the capability to view and manage configuration settings. For example, if the administrator made a mistake when configuring the set of permitted users' authentication credentials, providing the capability to view the lists of authentication credentials affords them the ability to review the list and discover any mistakes that might have been made.
- OE.NO_EVIL: contributes to mitigating this threat by ensuring that the administrators are non-hostile and are trained to appropriately manage and administer the TOE.
- OE.NO_GENERAL_PURPOSE: also helps to mitigate this threat by ensuring that there can be no accidental errors due to the introduction of unauthorized software or data, by ensuring that there are no general-purpose or storage repository applications available on the TOE.

8.1.1.8 T.ACCIDENTAL_CRYPTO_COMPROMISE

A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION: contributes to the mitigation of this threat by ensuring that any residual data is removed from network packet objects and ensuring that cryptographic material is not accessible once it is no longer needed.
- OE.RESIDUAL_INFORMATION: See previous (O.RESIDUAL_INFORMATION).
- O.SELF_PROTECTION: ensures that the TOE will have adequate protection from external sources and that all TSP functions are invoked.
- OE.SELF_PROTECTION: ensures that the TOE IT environment will have protection similar to that of the TOE.

8.1.1.9 T.MASQUERADE

A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

This Threat is satisfied by ensuring that:

- O.TOE_ACCESS: mitigates this threat by controlling logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE. Finally, the TOE includes requirements that ensure protected channels are used to authenticate wireless users and to communicate with critical portions of the TOE IT environment.
- OE.TOE_ACCESS: supports TOE authentication by providing an authentication server in the TOE IT environment. The environment also includes requirements that ensure protected channels are used to communicate with critical portions of the TOE IT environment.
- OE.TOE_NO_BYPASS: contributes to mitigating this threat by ensuring that wireless clients must be configured for all information flowing between a wireless client and another client or other host on the network without passing through the TOE.

8.1.1.10 T.POOR_DESIGN

Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.

This Threat is satisfied by ensuring that:

- O.CONFIGURATION_IDENTIFICATION: plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design documentation and the ability to report and resolve security flaws.
- O.DOCUMENTED_DESIGN: counters this threat, to a degree, by requiring that the TOE be developed using sound engineering principles. The use of a high level design and the functional specification ensure that developers responsible for TOE development understand the overall design of the TOE. This in turn decreases the likelihood of design flaws and increases the chance that accidental design errors will be discovered. ADV_RCR.1 ensures that the TOE design is consistent across the High Level Design and the Functional Specification.
- O.VULNERABILITY_ANALYSIS_TEST: ensures that the TOE has been analyzed for obvious vulnerabilities and that any vulnerabilities found have been removed or otherwise mitigated, this includes analysis of any probabilistic or permutational mechanisms incorporated into a TOE claiming conformance to this PP.

8.1.1.11 T.POOR_IMPLEMENTATION

Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.

This Threat is satisfied by ensuring that:

- O.CONFIGURATION_IDENTIFICATION: plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. This ensures that changes to the TOE are performed in structure manner and tracked.
- O.PARTIAL_FUNCTIONAL_TESTING: ensures that the developers provide evidence and demonstration that all security functions perform as specified through independent sample testing.
- O.VULNERABILITY_ANALYSIS_TEST: ensures that the TOE has been analyzed and tested to demonstrate that it is resistant to obvious vulnerabilities.

8.1.1.12 T.POOR_TEST

The developer or tester performs insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may occur, resulting in incorrect TOE behavior being undiscovered leading to flaws that may be exploited by a mischievous user or program.

This Threat is satisfied by ensuring that:

- O.CORRECT_TSF_OPERATION: provides assurance that the TSF continues to operate as expected in the field.
- Also see section 6 of the "US Government Wireless Local Area Network (WLAN) Access System Protection Profile for Basic Robustness Environments."

8.1.1.13 T.RESIDUAL_DATA

A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION: contributes to the mitigation of this threat by ensuring that any residual data is removed from network packet objects and ensuring that cryptographic material is not accessible once it is no longer needed.
- OE.RESIDUAL_INFORMATION: See previous (O.RESIDUAL_INFORMATION).

8.1.1.14 T.TSF_COMPROMISE

A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).

This Threat is satisfied by ensuring that:

- O.MANAGE: mitigate this threat by restricting access to administrative functions and management of TSF data to the administrator.
- OE.MANAGE: ensures that the administrator can view security relevant audit events.
- O.RESIDUAL_INFORMATION: contributes to the mitigation of this threat by ensuring that any residual data is removed from network packet objects and ensuring that cryptographic material is not accessible once it is no longer needed.
- OE.RESIDUAL_INFORMATION: See previous (O.RESIDUAL_INFORMATION).
- O.SELF_PROTECTION: requires that the TOE environment be able to protect itself from tampering and that the security mechanisms in the TOE cannot be bypassed. Without this objective, there could be no assurance that users could not view or modify TSF data or TSF executables.
- OE.SELF_PROTECTION: ensures that the TOE IT environment will have protection similar to that of the TOE.

8.1.1.15 T.UNATTENDED_SESSION

A user may gain unauthorized access to an unattended session.

This Threat is satisfied by ensuring that:

- O.TOE_ACCESS: helps to mitigate this threat by including mechanisms that place controls on administrator sessions. Administrator sessions are dropped after an Administrator defined time period of inactivity. Dropping the connection of a session (after the specified time period) reduces the risk of someone accessing the machine where the session was established, thus gaining unauthorized access to the session.

8.1.1.16 T.UNAUTHORIZED_ACCESS

A user may gain access to services (either on the TOE or by sending data through the TOE) for which they are not authorized according to the TOE security policy.

This Threat is satisfied by ensuring that:

- O.INTRUSION: works to mitigate this threat by detecting, informing the administrator and, if possible, containing wireless intrusion attacks.
- O.MEDIATE: works to mitigate this threat by ensuring that all network packets that flow through the TOE are subject to the information flow policies.
- O.TOE_ACCESS: The TOE requires authentication prior to gaining access to certain services on or mediated by the TOE.
- OE.TOE_ACCESS: See previous (O.TOE_ACCESS).

- O.SELF_PROTECTION: The TSF and its environment must ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services.
- OE.SELF_PROTECTION: See previous (OE.SELF_PROTECTION).
- O.MANAGE: The TOE and its environment restrict the ability to modify the security attributes associated with the TOE to the administrator. These objectives ensure that no other user can modify the information flow policy to bypass the intended TOE security policy.
- OE.MANAGE: See previous (O.MANAGE).
- OE.TOE_NO_BYPASS: contributes to mitigating this threat by ensuring that wireless clients must be configured to use the wireless access system for all information flowing between a wireless client and any other host on the network. If the clients are properly configured, any information passing through the TOE will be inspected to ensure it is authorized by TOE polices.

8.1.1.17 T.UNAUTH_ADMIN_ACCESS

An unauthorized user or process may gain access to an administrative account.

This Threat is satisfied by ensuring that:

- O.ADMIN_GUIDANCE: helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is not secure.
- O.MANAGE: mitigate this threat by restricting access to administrative functions and management of TSF data to the administrator.
- OE.MANAGE: See previous (O.MANAGE).
- O.TOE_ACCESS: helps to mitigate this threat by including mechanisms to authenticate TOE administrators and place controls on administrator sessions.
- OE.TOE_ACCESS: See previous (O.TOE_ACCESS).
- OE.NO_EVIL: helps to mitigate this threat by ensuring that the TOE administrators have guidance that instructs them in how to administer the TOE in a secure manner.

8.1.1.18 A.NO_EVIL

Administrators are non-hostile, appropriately trained and follow all administrator guidance.

This Assumption is satisfied by ensuring that:

- OE.NO_EVIL: Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.

8.1.1.19 A.NO_GENERAL_PURPOSE

There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

This Assumption is satisfied by ensuring that:

- OE.NO_GENERAL_PURPOSE: There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

8.1.1.20 A.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: The environment provides physical security commensurate with the value of the TOE and the data it contains.

8.1.1.21 A.TOE_NO_BYPASS

Wireless clients are configured so that information cannot flow between a wireless client and any other wireless client or host networked to the TOE without passing through the TOE.

This Assumption is satisfied by ensuring that:

- OE.TOE_NO_BYPASS: Wireless clients are configured so that information cannot flow between a wireless client and any other wireless client or host networked to the TOE without passing through the TOE.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that table below indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFRs) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.AUDIT_GENERATION	**O.CORRECT_TSF_OPERATION**	**O.CRYPTOGRAPHY**	**O.CRYPTOGRAPHY_VALIDATED**	**O.DISPLAY_BANNER**	**O.MANAGE**	**O.MEDIATE**	**O.RESIDUAL_INFORMATION**	**O.SELF_PROTECTION**	**O.TIME_STAMPS**	**O.TOE_ACCESS**	**O.INTRUSION**
FAU_ARP.1												X
FAU_GEN.1	X											
FAU_GEN.2	X											
FAU_SAA.3												X
FAU_SEL.1	X											
FCS_BCM_(EXT).1			X	X								
FCS_CKM.1(1)			X	X								
FCS_CKM.1(2)			X	X								
FCS_CKM.2			X	X								
FCS_CKM_(EXT).2			X	X				X				
FCS_CKM.4			X	X				X				
FCS_COP.1(1)			X	X								
FCS_COP.1(2)			X	X								
FCS_COP.1(3)			X	X								
FCS_COP.1(4)			X	X								
FCS_COP.1(5)			X	X								

	O.AUDIT_GENERATION	O.CORRECT_TSF_OPERATION	O.CRYPTOGRAPHY	O.CRYPTOGRAPHY_VALIDATED	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.RESIDUAL_INFORMATION	O.SELF_PROTECTION	O.TIME_STAMPS	O.TOE_ACCESS	O.INTRUSION
FCS_COP.1(6)			X	X								
FCS_COP.1(7)			X	X								
FCS_COP_(EXT).1			X	X								
FDP_IFC.1(1)							X					
FDP_IFF.1(1)							X					
FDP_IFC.1(2)							X					
FDP_IFF.1(2)							X					
FDP_PUD_(EXT).1							X					
FDP_RIP.1								X				
FIA_AFL.1											X	
FIA_ATD.1(1)											X	
FIA_ATD.1(2)											X	
FIA_SOS.1											X	
FIA_UAU.1							X				X	
FIA_UAU_(EXT).5							X				X	
FIA_UID.2							X				X	
FIA_USB.1	X											
FMT_MOF.1(1)						X						
FMT_MOF.1(2)						X						
FMT_MOF.1(3)						X						
FMT_MOF.1(4)						X						
FMT_MSA.1(1)						X						
FMT_MSA.1(2)						X						
FMT_MSA.2						X						
FMT_MSA.3(1)						X						
FMT_MSA.3(2)						X						
FMT_MTD.1(1)						X						
FMT_MTD.1(2)						X						
FMT_MTD.1(3)						X						
FMT_MTD.1(4)						X						
FMT_MTD.1(5)						X						
FMT_SMF.1(1)						X						
FMT_SMF.1(2)						X						
FMT_SMF.1(3)						X						
FMT_SMF.1(4)						X						
FMT_SMF.1(5)						X						
FMT_SMF.1(6)						X						
FMT_SMF.1(7)						X						
FMT_SMR.1						X						
FPT_STM_(EXT).1	X									X		

	O.AUDIT_GENERATION	O.CORRECT_TSF_OPERATION	O.CRYPTOGRAPHY	O.CRYPTOGRAPHY_VALIDATED	O.DISPLAY_BANNER	O.MANAGE	O.MEDIATE	O.RESIDUAL_INFORMATION	O.SELF_PROTECTION	O.TIME_STAMPS	O.TOE_ACCESS	O.INTRUSION
FPT_TST_(EXT).1		X										
FPT_TST.1(1)		X										
FPT_TST.1(2)		X										
FRU_RSA.1											X	
FTA_SSL.3											X	
FTA_TAB.1					X							
FTP_ITC_(EXT).1	X										X	
FTP_TRP.1											X	
ADV_ARC.1									X			

Table 7 Objective to Requirement Correspondence

8.2.1.1 O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security-relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.
- FAU_GEN.2: ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID. In all other cases, the association is based on the source network identifier, which is presumed to be the correct identity, but cannot be confirmed since these subjects are not authenticated.
- FAU_SEL.1: allows for the selection of events to be audited. This requires that the criteria used for the selection of auditable events to be defined. For example, the user identity can be used as selection criterion for the events to be audited.
- FIA_USB.1: plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authorized users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the subject that causes an audit record to be generated (e.g., presumed network address of an unauthenticated user may be a spoofed address).
- FPT_STM_(EXT).1: supports the audit functionality by ensuring that the TOE is capable of obtaining a time stamp for use in recording audit events.

- FTP_ITC_(EXT).1: provides a trusted channel for services provided by the TOE IT environment (the audit server and the time server).

8.2.1.2 O.CORRECT_TSF_OPERATION

The TOE will provide the capability to verify the correct operation of the TSF.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST_(EXT).1: is necessary to ensure the correctness of the TSF software and TSF data. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies.
- FPT_TST.1(1): The FPT_TST.1(1) for crypto and FPT_TST.1(2) for key generation functional requirement has been included to address the critical nature and specific handling of the cryptographic related TSF data. Since the cryptographic TSF data has specific FIPS PUB requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements.
- FPT_TST.1(2): See previous.

8.2.1.3 O.CRYPTOGRAPHY

The TOE shall provide cryptographic functions to maintain the confidentiality and allow for detection of modification of user data that is transmitted between physically separated portions of the TOE, or outside of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FCS_BCM_(EXT).1: Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules implemented in hardware, in software, or in hardware/software combinations [FCS_BCM_(EXT).1]. The cryptographic services offered by this baseline capability are augmented and customized in the TOE to support medium robustness environments. These TOE services are based primarily upon functional security requirements in the areas of key management and cryptographic operations. In the area of key management there are functional requirements that address the generation of symmetric keys [FCS_CKM.1 (1)], and the generation of asymmetric keys [FCS_CKM.1 (2)]; methods of manual and automated cryptographic key distribution [FCS_CKM.2]; cryptographic key destruction [FCS_CKM.4]; techniques for cryptographic key validation and packaging [FCS_CKM.1]; and cryptographic key handling and storage [FCS_CKM_(EXT).2]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; cryptographic key agreement [FCS_COP.1 (4)]; and improved random number generation [FCS_COP_(EXT).1].
- FCS_CKM.1(1): See FCS_BCM_(EXT).1.
- FCS_CKM.1(2): See FCS_BCM_(EXT).1.
- FCS_CKM.2: See FCS_BCM_(EXT).1.
- FCS_CKM_(EXT).2: See FCS_BCM_(EXT).1.
- FCS_CKM.4: See FCS_BCM_(EXT).1.
- FCS_COP.1(1): See FCS_BCM_(EXT).1.
- FCS_COP.1(2): See FCS_BCM_(EXT).1.
- FCS_COP.1(3): See FCS_BCM_(EXT).1.
- FCS_COP.1(4): See FCS_BCM_(EXT).1.
- FCS_COP.1(5): See FCS_BCM_(EXT).1.
- FCS_COP.1(6): See FCS_BCM_(EXT).1.
- FCS_COP.1(7): See FCS_BCM_(EXT).1.
- FCS_COP_(EXT).1: See FCS_BCM_(EXT).1.

8.2.1.4 O.CRYPTOGRAPHY_VALIDATED

The TOE will use NIST FIPS 140-1/2 validated cryptomodules for cryptographic services implementing NIST-approved security functions and random number generation services used by cryptographic functions.

This TOE Security Objective is satisfied by ensuring that:

- FCS_BCM_(EXT).1: Baseline cryptographic services are provided in the TOE by FIPS PUB 140-2 compliant modules implemented in hardware, in software, or in hardware/software combinations [FCS_BCM_(EXT).1]. The cryptographic services offered by this baseline capability are augmented and customized in the TOE to support medium robustness environments. These TOE services are based primarily upon functional security requirements in the areas of key management and cryptographic operations. In the area of key management there are functional requirements that address the generation of symmetric keys [FCS_CKM.1 (1)], and the generation of asymmetric keys [FCS_CKM.1 (2)]; methods of manual and automated cryptographic key distribution [FCS_CKM.2]; cryptographic key destruction [FCS_CKM.4]; techniques for cryptographic key validation and packaging [FCS_CKM.1]; and cryptographic key handling and storage [FCS_CKM_(EXT).2]. Specific functional requirements in the area of cryptographic operations address data encryption and decryption [FCS_COP.1 (1)]; cryptographic signatures [FCS_COP.1 (2)]; cryptographic hashing [FCS_COP.1 (3)]; cryptographic key agreement [FCS_COP.1 (4)]; and improved random number generation [FCS_COP_(EXT).1].
- FCS_CKM.1(1): See FCS_BCM_(EXT).1.
- FCS_CKM.1(2): See FCS_BCM_(EXT).1.
- FCS_CKM.2: See FCS_BCM_(EXT).1.
- FCS_CKM_(EXT).2: See FCS_BCM_(EXT).1.
- FCS_CKM.4: See FCS_BCM_(EXT).1.
- FCS_COP.1(1): See FCS_BCM_(EXT).1.
- FCS_COP.1(2): See FCS_BCM_(EXT).1.
- FCS_COP.1(3): See FCS_BCM_(EXT).1.
- FCS_COP.1(4): See FCS_BCM_(EXT).1.
- FCS_COP.1(5): See FCS_BCM_(EXT).1.
- FCS_COP.1(6): See FCS_BCM_(EXT).1.
- FCS_COP.1(7): See FCS_BCM_(EXT).1.
- FCS_COP_(EXT).1: See FCS_BCM_(EXT).1.

8.2.1.5 O.DISPLAY_BANNER

The TOE will display an advisory warning prior to establishing an administrator session regarding use of the TOE prior to permitting the use of any TOE services that requires authentication.

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: meets this objective by requiring that the TOE display an administrator defined banner before a user can establish an authenticated session. This banner is under complete control of the administrator, who can specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire. The only time that it is envisioned that an authenticated session would need to be established is for the performance of TOE administration. Bannering is not necessary prior to use of services that pass network traffic through the TOE.

8.2.1.6 O.MANAGE

The TOE will provide functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1(1): FMT_MOF.1(1)(2) and (3) ensure that the administrator has the ability manage the cryptographic, audit, and authentication functions.
- FMT_MOF.1(2): See FMT_MOF.1(1).

- FMT_MOF.1(3): See FMT_MOF.1(1).
- FMT_MOF.1(4): ensures only administrator has the ability to manage the WIP signature events and actions taken upon detection of security violation.
- FMT_MSA.1(1): ensures only administrator has the ability to modify the information security attributes in FDP_IFF.1(1) and operations in FDP_IFC.1(1) that are used to control information flow.
- FMT_MSA.1(2): ensures only administrator has the ability to modify the security attributes in FDP_IFF.1(2) that are used to control information flow.
- FMT_MSA.2: provides the administrator the ability to accept only secure values and modify security attributes.
- FMT_MSA.3(1): ensures that by default, information flow is denied, unless explicitly allow only by administrator.
- FMT_MSA.3(2): ensures that by default, information flow is permissive, unless explicitly allow only by administrator.
- FMT_MTD.1(1): FMT_MTD.1(1) (2) and (3) ensure that the administrator can manage TSF data. This PP specifically identifies audit preselection, identification, and authentication data. An ST author, may use additional iterations to address TSF data that has not already been specified by other requirements. This is necessary because the ST author may add TSF data in assignments that cannot be addressed a priori by the PP authors.
- FMT_MTD.1(2): See FMT_MTD.1(1).
- FMT_MTD.1(3): See FMT_MTD.1(1).
- FMT_MTD.1(4): ensures that only administrator can manage the policies and ACLs rulesets to control the information flow.
- FMT_MTD.1(5): ensures that only administrator can manage the maximum number of session (per user) setting.
- FMT_SMF.1(1): FMT_SMF.1(1), (2), and (3) support this objective by identifying the management functions for cryptographic data, audit records, and cryptographic key data.
- FMT_SMF.1(2): See FMT_SMF.1(1).
- FMT_SMF.1(3): See FMT_SMF.1(1).
- FMT_SMF.1(4):identifies the function for management of WIP signature events and actions in support of FAU_SAA.3 and FAU_ARP.1.
- FMT_SMF.1(5): identifies the function for management of administrator and user identification and authentication credentials.
- FMT_SMF.1(6): identifies the function for management of information flow Policy and ACL rulesets.
- FMT_SMF.1(7): identifies the function for management of the maximum number of session a user can have opened concurrently.
- FMT_SMR.1: defines the specific security roles to be supported.

8.2.1.7 O.MEDIATE

The TOE must mediate the flow of information to and from wireless clients communicating via the TOE in accordance with its security policy.

This TOE Security Objective is satisfied by ensuring that:

- FDP_IFC.1(1): defines the scope of the information flow policy including sending and receiving subjects, information that is being sent and received, and the operations performed on that information.
- FDP_IFF.1(1): defines the security attributes and rules that are used by the TOE in accordance with its security policy to control and mediate the information flow.
- FDP_IFC.1(2): defines the scope of the information flow policy including sending and receiving subjects, information that is being sent and received, and the operations performed on that information.
- FDP_IFF.1(2): defines the security attributes and rules that are used by the TOE in accordance with its security policy to control and mediate the information flow.

- FDP_PUD_(EXT).1: allows the administrator to control whether or not unencrypted data will be allowed to pass through the TOE.
- FIA_UAU.1: ensures that the TOE has the ability to mediate packet flow based upon the authentication credentials of the wireless user.
- FIA_UAU_(EXT).5: See FIA_UAU.1.
- FIA_UID.2: See FIA_UAU.1.

8.2.1.8 O.RESIDUAL_INFORMATION

The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM_(EXT).2: places requirements on how cryptographic keys are managed within the TOE. This requirement places restrictions in addition to FDP_RIP.1, in that when a cryptographic key is moved from one location to another (e.g., calculated in some scratch memory and moved to a permanent location) that the memory area is immediately cleared as opposed to waiting until the memory is reallocated to another subject.
- FCS_CKM.4: applies to the destruction of cryptographic keys used by the TSF. This requirement specifies how and when cryptographic keys must be destroyed. The proper destruction of these keys is critical in ensuring the content of these keys cannot possibly be disclosed when a resource is reallocated to a user.
- FDP_RIP.1: is used to ensure the contents of resources are not available once the resource is reallocated. For this TOE it is critical that the memory used to build network packets is either cleared or that some buffer management scheme be employed to prevent the contents of a packet being disclosed in a subsequent packet (e.g., if padding is used in the construction of a packet, it must not contain another user’s data or TSF data).

8.2.1.9 O.SELF_PROTECTION

The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.

This TOE Security Objective is satisfied by ensuring that:

- ADV_ARC.1: provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs**.** Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.

8.2.1.10 O.TIME_STAMPS

The TOE shall obtain reliable time stamps.

This TOE Security Objective is satisfied by ensuring that:

- FPT_STM_(EXT).1: requires that the TOE be able to obtain reliable time stamps for its own use and therefore, partially satisfies this objective. Time stamps include date and time and are reliable in that they are always available to the TOE, and the clock must be monotonically increasing.

8.2.1.11 O.TOE_ACCESS

The TOE will provide mechanisms that control a user’s logical access to the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: FIA_AFL.1 ensures that the TOE can protect itself and its users from brute force attacks on their authentication credentials.

- FIA_ATD.1(1): FIA_ATD.1(1) and (2) Management requirements provide additional control to supplement the authentication requirements.
- FIA_ATD.1(2): See FIA_ATD.1(1).
- FIA_SOS.1: FIA_SOS.1 contributes to this objective by ensuring that the password length is long enough that a brute force attack cannot crack them in a reasonable amount of time.
- FIA_UAU.1: FIA_UAU.1 and FIA_UAU_(EXT).5 contribute to this objective by ensuring that administrators and users are authenticated before they are provided access to the TOE or its services.
- FIA_UAU_(EXT).5: See FIA_UAU.1.
- FIA_UID.2: plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any mediated functions. In most cases, the identification cannot be authenticated (e.g., a user attempting to send a data packet through the TOE that does not require authentication. It is impractical to require authentication of all users that attempt to send data through the TOE, therefore, the requirements specified in the TOE require authentication where it is deemed necessary. This does impose some risk that a data packet was sent from an identity other than that specified in the data packet.
- FTA_SSL.3: ensures that inactive user and administrative sessions are dropped.
- FRU_RSA.1: ensures users do not initiate more sessions than allow by administrator.
- FTP_TRP.1: ensures that remote users have a trusted path in order to authenticate.
- FTP_ITC_(EXT).1: provides a trusted channel for services provided by the TOE IT environment (the remote authentication server)

8.2.1.12 O.INTRUSION

The TOE will detect wireless intrusion attacks, alert administrators and, where possible, prevent or contain the intrusion attempts.

This TOE Security Objective is satisfied by ensuring that:

- FAU_ARP.1: addresses the remainder of this objective by taking appropriate action on the detection of an intrusion attempt, including alerting an administrator and, if possible, preventing or containing the attempted intrusion.
- FAU_SAA.3: addresses the first part of this objective by requiring the TOE detect a defined set of wireless intrusion attempts.

Note that O.ADMIN_GUIDANCE, O.CONFIGURATION_IDENTIFICATION, O.DOCUMENTED_DESIGN, O.PARTIAL_FUNCTIONAL_TESTING, and O.VULNERABILITY_ANALYSIS are addressed by the SARs. Please refer to the PP for the rationales.

8.3 Security Assurance Requirements Rationale

The TOE is intended for an environment requiring a moderate to high level of assurance in the security functionality of conventional commodity TOEs, as presented in the statement of security environment (Section 3). EAL4 was selected as the assurance level because the TOE is a commercial product whose users require a moderate to high level of independently assured security. The TOE is targeted at an environment with good physical access security where it is assumed that attackers will have low to enhanced-basic attack potential. Augmentation was chosen to provide the added assurance that is provided by defining flaw remediation procedures. Therefore, the target assurance level of EAL 4 augmented with ALC_FLR.2 is appropriate for such an environment.

8.4 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST and PP. As indicated in the table, all of the dependencies are satisfied with the exception of FAU_ARP.1.

ST Requirement	CC Dependencies	ST Dependencies
FAU_ARP.1	FAU_SAA.1	None (see note below)
FAU_GEN.1	FPT_STM.1	FPT_STM_(EXT).1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.2
FAU_SAA.3	None	None
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1
FCS_BCM_(EXT).1	None	None
FCS_CKM.1(1)	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_CKM.2, FCS_COP.1, and FCS_CKM.4
FCS_CKM.1(2)	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_CKM.2, FCS_COP.1, and FCS_CKM.4
FCS_CKM.2	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_CKM_(EXT).2	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_CKM.4	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1)	FCS_CKM.1
FCS_COP.1(1)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(2)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(3)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(4)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(5)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(6)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1(7)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP_(EXT).1	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FDP_IFC.1(1)	FDP_IFF.1	FDP_IFF.1(1)
FDP_IFF.1(1)	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1(1) and FMT_MSA.3
FDP_IFC.1(2)	FDP_IFF.1	FDP_IFF.1(2)
FDP_IFF.1(2)	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1(2) and FMT_MSA.3
FDP_PUD_(EXT).1	FCS_COP.1(1)	FCS_COP.1(1)
FDP_RIP.1	None	None
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(1)	None	None
FIA_ATD.1(2)	None	None
FIA_SOS.1	None	None
FIA_UAU.1	FIA_UID.1	FIA_UID.2
FIA_UAU_(EXT).5	FIA_UAU.1	FIA_UAU.1
FIA_UID.2	None	None
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1(1)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MOF.1(2)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MOF.1(3)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MOF.1(4)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1(1)	(FDP_ACC.1 or FDP_IFC.1) and FMT_SMR.1 and FMT_SMF.1	FDP_IFC.1(1), FMT_SMR.1, and FMT_SMF.1
FMT_MSA.1(2)	(FDP_ACC.1 or FDP_IFC.1) and	FDP_IFC.1(2), FMT_SMR.1, and

ST Requirement	CC Dependencies	ST Dependencies
	FMT_SMR.1 and FMT_SMF.1	FMT_SMF.1
FMT_MSA.2	(FDP_ACC.1 or FDP_IFC.1) and FMT_MSA.1 and FMT_SMR.1	FDP_IFC.1, FMT_MSA.1, and FMT_SMR.1
FMT_MSA.3(1)	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1(1) and FMT_SMR.1
FMT_MSA.3(2)	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1(2) and FMT_SMR.1
FMT_MTD.1(1)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1(2)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1(3)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1(4)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1(1)	None	None
FMT_SMF.1(2)	None	None
FMT_SMF.1(3)	None	None
FMT_SMF.1(4)	None	None
FMT_SMF.1(5)	None	None
FMT_SMF.1(6)	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_STM_(EXT).1	None	None
FPT_TST_(EXT).1	FCS_COP.1(2) and FCS_COP.1(3)	FCS_COP.1(2) and FCS_COP.1(3)
FPT_TST.1(1)	None	None
FPT_TST.1(2)	None	None
FRU_RSA.1	None	None
FTA_SSL.3	None	None
FTA_TAB.1	None	None
FTP_ITC_(EXT).1	None	None
FTP_TRP.1	None	None
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.4 and ADV_TDS.3
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3
ADV_IMP.1	ADV_TDS.3 and ALC_TAT.1	ADV_TDS.3 and ALC_TAT.1
ADV_TDS.3	ADV_FSP.4	ADV_FSP.4
AGD_OPE.1	ADV_FSP.1	ADV_FSP.4
AGD_PRE.1	None	None
ALC_CMC.4	ALC_CMS.1 and ALC_DVS.1 and ALC_LCD.1	ALC_CMS.4 and ALC_DVS.1 and ALC_LCD.1
ALC_CMS.4	None	None
ALC_DEL.1	None	None
ALC_DVS.1	None	None
ALC_FLR.2	None	None
ALC_LCD.1	None	None
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.4 and ATE_FUN.1
ATE_DPT.2	ADV_ARC.1 and ADV_TDS.3 and ATE_FUN.1	ADV_ARC.1 and ADV_TDS.3 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.2
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.4 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.2 and ATE_FUN.1
AVA_VAN.3	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.4 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1

Table 8 Requirement Dependencies

Note: In CC Version 3.1 R2, FAU_SAA.3 is no longer hierarchical to FAU_SAA.1 (as opposed to Version 2.3). FAU_ARP.1 has a dependency on FAU_SAA.1 because FAU_ARP.1 defines the action(s) that occur when a potential security violation is detected and FAU_SAA.1 defines the rules for monitoring audited events for detecting potential security violation. FAU_SAA.3 also defines a rule for detecting potential security violation; when a system event does not match the normal signature event. Therefore, including FAU_SAA.3 should meet the intent of this dependency for FAU_SAA.1.

8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 9 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	**Cryptographic support**	**User data protection**	**Identification and authentication**	**Security management**	**Protection of the TSF**	**Resource utilization**	**TOE access**	**Trusted path/channels**
FAU_ARP.1	**X**								
FAU_GEN.1	**X**								
FAU_GEN.2	**X**								
FAU_SAA.3	**X**								
FAU_SEL.1	**X**								
FCS_BCM_(EXT).1		**X**							
FCS_CKM.1(1)		**X**							
FCS_CKM.1(2)		**X**							
FCS_CKM.2		**X**							
FCS_CKM_(EXT).2		**X**							
FCS_CKM.4		**X**							
FCS_COP.1(1)		**X**							
FCS_COP.1(2)		**X**							
FCS_COP.1(3)		**X**							
FCS_COP.1(4)		**X**							
FCS_COP.1(5)		**X**							
FCS_COP.1(6)		**X**							
FCS_COP.1(7)		**X**							
FCS_COP_(EXT).1		**X**							
FDP_IFC.1(1)			**X**						
FDP_IFF.1(1)			**X**						
FDP_IFC.1(2)			**X**						
FDP_IFF.1(2)			**X**						
FDP_PUD_(EXT).1			**X**						
FDP_RIP.1			**X**						

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	Resource utilization	TOE access	Trusted path/channels
FIA_AFL.1				**X**					
FIA_ATD.1(1)				**X**					
FIA_ATD.1(2)				**X**					
FIA_SOS.1				**X**					
FIA_UAU.1				**X**					
FIA_UAU_(EXT).5				**X**					
FIA_UID.2				**X**					
FIA_USB.1				**X**					
FMT_MOF.1(1)					**X**				
FMT_MOF.1(2)					**X**				
FMT_MOF.1(3)					**X**				
FMT_MOF.1(4)					**X**				
FMT_MSA.1(1)					**X**				
FMT_MSA.1(1)					**X**				
FMT_MSA.2					**X**				
FMT_MSA.3(1)					**X**				
FMT_MSA.3(2)					**X**				
FMT_MTD.1(1)					**X**				
FMT_MTD.1(2)					**X**				
FMT_MTD.1(3)					**X**				
FMT_MTD.1(4)					**X**				
FMT_MTD.1(5)					**X**				
FMT_SMF.1(1)					**X**				
FMT_SMF.1(2)					**X**				
FMT_SMF.1(3)					**X**				
FMT_SMF.1(4)					**X**				
FMT_SMF.1(5)					**X**				
FMT_SMF.1(6)					**X**				
FMT_SMF.1(7)					**X**				
FMT_SMR.1					**X**				
FPT_STM_(EXT).1						**X**			
FPT_TST_(EXT).1						**X**			
FPT_TST.1(1)						**X**			
FPT_TST.1(2)						**X**			
FRU_RSA.1							**X**		
FTA_SSL.3								**X**	
FTA_TAB.1								**X**	
FTP_ITC_(EXT).1									**X**
FTP_TRP.1									**X**

Table 9 Security Functions vs. Requirements Mapping

8.6 PP Claims Rationale

See Section 7, Protection Profile Claims.

Appendix A

Optional Software Module	Description
Remote AP	Allows an AP to be securely connected from a remote location to a controller across the Internet. Allows the remote AP to be plugged directly into an Internet-connected DSL router; a controller does not need to be installed at the remote location.
Voice Services	Provides standards-based voice over WiFi features and voice control and management.
Ortronics AP	Enables support of the Ortronics Wi-Jack DUO family of wall-installable wireless APs.
Secure Enterprise Mesh	Allows an AP to be configured as a mesh node that bridges multiple Ethernet LANs or extends wireless coverage over wireless hops.
xSec	Enables support for xSec, a Federal Information Processing Standard (FIPS)-certifiable Layer-2 encryption protocol. The encryption algorithm is FIPS-certified AES. This module is used by client who wants extra security at the Layer 2 (MAC level). Encryption is now done at Layer 2 (MAC) and at Layer 3 (IPSec) or Layer 4 (TLS and SSH).
External Services Interface (ESI)	Supports automatic redirect of clients to external third-party devices that provide inline network services such as anti-virus, intrusion detection system (IDS), content filtering, and client remediation.