Aruba, a Hewlett Packard Enterprise company Virtual Intranet Access (VIA) Client Version 3.0 (IVPNCPP14) Security Target

Version 1.5
05/03/2018

Prepared for:

Aruba, a Hewlett Packard Enterprise Company

1344 Crossman Ave. Sunnyvale, CA 94089

Prepared By:

Gossamer Laboratories

www.gossamersec.com

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 the Virtual Intranet Access (VIA) Client Version 3.0 provided by Aruba, a Hewlett Packard Enterprise company The TOE is being evaluated as an IPsec VPN client.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

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.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - 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. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- The Protection Profile for IPsec Virtual Private Network (VPN) Clients (IVPNCPP) uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title –Aruba, a Hewlett Packard Enterprise company Virtual Intranet Access (VIA) Client Version 3.0 (IVPNCPP14) Security Target

ST Version – Version 1.5

ST Date – 05/03/2018

1.2 TOE Reference

TOE Identification – Aruba, a Hewlett Packard Enterprise company Virtual Intranet Access (VIA) Client Version 3.0

TOE Developer – Aruba, a Hewlett Packard Enterprise company

Evaluation Sponsor – Aruba, a Hewlett Packard Enterprise company

1.3 TOE Overview

The Target of Evaluation (TOE) is the Aruba, a Hewlett Packard Enterprise company Virtual Intranet Access (VIA) Client Version 3.0.

This ST focuses on the IPSEC VPN capabilities of the TOE. The TOE provides secure remote network connectivity for Linux, Android, and Windows mobile devices and workstations. The TOE has two primary purposes:

- to provide secure corporate access to employee workstations and smartphones from anywhere
- to provide ease-of-use for the end users and network administrators

The IPsec VPN capabilities are the primary function of the TOE. IPSec is used by the TOE to protect communication between itself and an Aruba Mobility Controller over an unprotected network.

1.4 TOE Description

The TOE is a hybrid Internet Protocol Security (IPsec)/Secure Sockets Layer (SSL) VPN client available for multiple client operating systems. IPsec is the sole means of securing network traffic; SSL functionality involves encapsulation of IPsec inside HTTPS-formatted packets in order to traverse firewalls and proxies where required. SSL functionality is not included in this evaluation and is disabled by default.

VIA can be downloaded directly from an Aruba Mobility Controller, pushed out using enterprise management tools, installed manually, or installed from the Google Play Store. An Aruba Mobility Controller is required to terminate connections from a VIA client – VIA is not a general-purpose VPN client that works with third-party VPN gateways.

1.4.1 TOE Architecture

The VIA Client runs on an end-user device and communicates with a server component located on a Mobility Controller. The server component is used to manage the client and ensure policies are enforced. The controller maintains certain VIA configuration profiles, such as the VIA authentication profile, the VIA connection profile, and the VIA web authentication profile. Each profile plays an important role in authenticating the users and establishing a secure connection. When multiple authentication profiles are available, the VIA client prompts the user to select an authentication profile.

The first time a connection is established, a user opens the VIA client and enters the server name, username, and password. VIA then connects to the server over an HTTPS connection and attempts to authenticate using the user supplied credentials. If the VIA web authentication list has more than one VIA authentication profile, the user can choose a VIA authentication profile from the available ones. After successful authentication, the VIA client downloads the appropriate VIA connection profile and establishes the IPsec connection if the user is connected to an untrusted network.

At a protocol level, VIA operates over UDP port 4500, which is defined for IKE/IPsec traversal of NATs in RFC 3947. VIA uses HTTPS over TCP port 443 in order to contact the authentication server and download configuration profile updates before establishing each IKE/IPsec connection.

1.4.1.1 Physical Boundaries

The TOE is the Aruba Virtual Intranet Access (VIA) client version 3.0 running on the following platforms:

- Samsung Galaxy S7, Samsung Galaxy S8, Samsung Note 8 with Android 7.1 – CC evaluated. Security Target for the evaluation can be found at https://www.niap-ccevs.org/Product/Compliant.cfm?PID=10849
- Microsoft Windows 10. - CC evaluated. Security Target for the evaluation can be found at https://www.niap-ccevs.org/Product/CompliantCC.cfm?CCID=2017.1007. This evaluation includes Windows Operating Systems (OS): Microsoft Windows 10 Home Edition (Anniversary Update) (32-bit and 64- bit versions), Microsoft Windows 10 Pro Edition (Anniversary Update) (32-bit and 64-bit versions), Microsoft Windows 10 Enterprise Edition (Anniversary Update) (32-bit and 64-bit versions), Microsoft Windows Server 2016 Standard Edition, Microsoft Windows Server 2016 Datacenter Edition. TOE Build: Windows 10: build 10.0.14393 and Windows Server 2016: build 10.0.14393.
- Linux (Red Hat Enterprise Linux 6.9 and CentOS Linux 6.9 with kernel version 2.6) – No CC evaluation or Security Target exists (i.e., Linux is unevaluated). See https://wiki.centos.org/Documentation or https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/ for associated documentation.

During evaluation testing, VIA was tested using the following platforms:

- Samsung Galaxy S8 with Android 7.1, Windows 10 Professional, and Centos 6.9

An Aruba Mobility Controller is required to be in the IT environment to communicate with the VIA Client. VIA supported on an Aruba Mobility Controller running one of the following ArubaOS versions:

- ArubaOS 6.4
- ArubaOS 6.5
- ArubaOS 8.2

ArubaOS 6.5 was used for testing. An ArubaOS Advanced Cryptography (ACR) license must also be installed on the Aruba Mobility Controller in order for the Suite B algorithms claimed in this ST to be available in the Aruba Common Cryptographic Module (CCM) version 1.0.0 and to enable client termination using these algorithms.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by Aruba Virtual Intranet Access (VIA) client:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

1.4.1.2.1 Cryptographic support

The IPsec implementation is the primary function of the TOE. IPSec is used by the TOE to protect communication between itself and an Aruba Mobility Controller over an unprotected network.

1.4.1.2.2 User data protection

The TOE ensures that residual information is protected from potential reuse in accessible objects such as network packets.

1.4.1.2.3 Identification and authentication

The TOE provides the ability to use, store, and protect X.509 certificates that are used for IPsec Virtual Private Network (VPN) connections. In some cases, the storage and protection of X.509 certificates and keys is provided by the underlying operating system.

1.4.1.2.4 Security management

The TOE and its IPsec VPN are fully configurable by a combination of functions provided directly by the TOE and those available to the associated VPN gateway.

1.4.1.2.5 Protection of the TSF

The TOE performs self-tests that cover the TOE as well as the functions necessary to securely update the TOE.

1.4.1.2.6 Trusted path/channels

The TOE acts as a VPN client using IPsec to established secure channels to corresponding VPN gateways.

1.4.2 TOE Documentation

Aruba offers a series of documents that describe use and administration of the applicable security features of the VIA products. The following document was examined as part of the evaluation:

Common Criteria Configuration Guidance, VPN Client Protection Profile, Aruba VIA Client v3.0, Version 2.0, March, 2018 (**Admin Guide**)

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Protection Profile for IPsec Virtual Private Network (VPN) Clients, Version 1.4, 21 October 2013 (IVPNCPP14) with the following NIAP Technical Decisions applied: TD0014, TD0037, TD0053, TD0079, TD0097, TD0107, TD0138, TD0140.
- Package Claims:
    - Protection Profile for IPsec Virtual Private Network (VPN) Clients, Version 1.4, 21 October 2013 (IVPNCPP14)

2.1 Conformance Rationale

The ST conforms to the IVPNCPP14. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the IVPNCPP14 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The IVPNCPP14 offers additional information about the identified security objectives, but that has not been reproduced here and the IVPNCPP14 should be consulted if there is interest in that material.

In general, the IVPNCPP14 has defined Security Objectives appropriate for IPsec VPN client and as such are applicable to the Aruba Virtual Intranet Access (VIA) Client TOE.

3.1 Security Objectives for the Operational Environment

OE.NO_TOE_BYPASS Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.

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

OE.TRUSTED_CONFIG Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the IVPNCPP14. The IVPNCPP14 defines the following extended requirements and since they are not redefined in this ST the IVPNCPP14 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FCS_CKM_EXT.2: Cryptographic Key Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_IPSEC_EXT.1: Extended: Internet Protocol Security (IPsec) Communications
- FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
- FDP_IFC_EXT.1: Subset Information Flow Control
- FIA_X509_EXT.1: Extended: X.509 Certificate Validation
- FIA_X509_EXT.2: Extended: X.509 Certificate Use and Management
- FPT_TST_EXT.1: Extended: TSF Self Test
- FPT_TUD_EXT.1: Extended: Trusted Update

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the IVPNCPP14. The refinements and operations already performed in the IVPNCPP14 are not identified (e.g., highlighted) here, rather the requirements have been copied from the IVPNCPP14 and any residual operations have been completed herein. Of particular note, the IVPNCPP14 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the IVPNCPP14 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the IVPNCPP14 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The IVPNCPP14 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Aruba Virtual Intranet Access (VIA) Client TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_CKM.1(1): Cryptographic Key Generation (Asymmetric Keys)
	FCS_CKM.1(2): Cryptographic Key Generation (for asymmetric keys - IKE)
	FCS_CKM_EXT.2: Cryptographic Key Storage
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (Cryptographic Hashing)
	FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)
	FCS_IPSEC_EXT.1: Extended: Internet Protocol Security (IPsec) Communications
	FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	FIA_X509_EXT.1: Extended: X.509 Certificate Validation
	FIA_X509_EXT.2: Extended: X.509 Certificate Use and Management
FMT: Security management	FMT_SMF.1(1): Specification of Management Functions
	FMT_SMF.1(2): Specification of Management Functions
FPT: Protection of the TSF	FPT_TST_EXT.1: Extended: TSF Self Test
	FPT_TUD_EXT.1: Extended: Trusted Update
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel

Table 1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic Key Generation (Asymmetric Keys) (FCS_CKM.1(1))

FCS_CKM.1(1).1

Refinement: The [***TOE***] shall generate asymmetric cryptographic keys used for key establishment in accordance with
- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for finite field-based key establishment schemes;
- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and [***no other curves***] (as defined in FIPS PUB 186-3, 'Digital Signature Standard');

- **[*no other*]**
  and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, 'Recommendation for Key Management' for information about equivalent key strengths.

5.1.1.2 Cryptographic Key Generation (for asymmetric keys - IKE) (FCS_CKM.1(2))

FCS_CKM.1(2).1

Refinement: The [***TOE***] shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a (TD0107 applied):: [***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes; , FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [no other curves]***]
and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.1.3 Cryptographic Key Storage (FCS_CKM_EXT.2)

FCS_CKM_EXT.2.1

The [***TOE***] shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.1.1.4 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

Refinement: The [***TOE***] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.1.5 Cryptographic Operation (Data Encryption/Decryption) (FCS_COP.1(1))

FCS_COP.1(1).1

Refinement: The [***TOE***] shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in GCM and CBC mode with cryptographic key sizes 128-bits and 256-bits that meets the following:
- FIPS PUB 197, 'Advanced Encryption Standard (AES)';
- NIST SP 800-38D, NIST SP 800-38A.

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

FCS_COP.1(2).1

Refinement: The [***TOE***] shall perform cryptographic signature services in accordance with a specified cryptographic algorithm:

- [***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA scheme, FIPS PUB 186-4, 'Digital Signature Standard', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [no other curves]***]
and cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.1.7 Cryptographic Operation (Cryptographic Hashing) (FCS_COP.1(3))

FCS_COP.1(3).1

Refinement: The [***TOE***] shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384***] and message digest sizes [***160, 256, 384***] bits that meet the following: FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.1.8 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(4))

FCS_COP.1(4).1

Refinement: The [***TOE***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384***], -key size [***160, 256, 384***], and message digest size of [***160, 256, 384***] bits that meet the following: FIPS PUB 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS PUB 180-4, 'Secure Hash Standard'.

5.1.1.9 Extended: Internet Protocol Security (IPsec) Communications (FCS_IPSEC_EXT.1)

FCS_IPSEC_EXT.1.1

The [***TOE***] shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2

The [***TOE***] shall implement [***tunnel mode***].

FCS_IPSEC_EXT.1.3

The [***TOE platform***] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4

The [***TOE***] shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC 4106, [***AES-CBC-128 (specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC***].

FCS_IPSEC_EXT.1.5

The [***TOE***] shall implement the protocol: [***IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [no other RFCs for hash functions]***].

FCS_IPSEC_EXT.1.6

The [***TOE***] shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [***no other algorithm***].

FCS_IPSEC_EXT.1.7

The [***TOE***] shall ensure that IKEv1 Phase 1 exchanges use only main mode

FCS_IPSEC_EXT.1.8

The [***TOE***] shall ensure that [***IKEv2 SA lifetimes can be configured by [VPN Gateway] based on [number of packets/number of bytes, length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]***].

FCS_IPSEC_EXT.1.9

The [***TOE***] shall generate the secret value x used in the IKE Diffie-Hellman key exchange ('x' in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**(224, 256, or 384)**] bits.

FCS_IPSEC_EXT.1.10

The [***TOE***] shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[**(112, 128, or 192)**] .

FCS_IPSEC_EXT.1.11

The [***TOE***] shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [***20 (384-bit Random ECP)***].

FCS_IPSEC_EXT.1.12

The [***TOE***] shall ensure that all IKE protocols perform peer authentication using a [***RSA, ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***no other method***].

FCS_IPSEC_EXT.1.13

The TSF shall support peer identifiers of the following types: [***Distinguished Name (DN)***] and [***no other reference identifier type]***]. (TD0037 applied)

FCS_IPSEC_EXT.1.14

The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer. (TD0037 applied)

FCS_IPSEC_EXT.1.15

The [***VPN Gateway (per TD0097)***] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 CHILD_SA***] connection. (Renumbered per TD0037)

5.1.1.10 Extended: Cryptographic operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The [***TOE***] shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***CTR_DRBG (AES)***]. (TD0079 applied)

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [*a software-based noise source,* ***a platform-based RBG***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.1.2 User data protection (FDP)

5.1.2.1 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1

The [***TOE***] shall enforce that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Extended: X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

The [***TOE***] shall validate certificates in accordance with the following rules:
- Perform RFC 5280 certificate validation and certificate path validation.
- Validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560***].
- Validate the certificate path by ensuring the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.
- Validate the extendedKeyUsage field according to the following rules:
o Certificates used for [***no other purpose***] shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3).

FIA_X509_EXT.1.2

The [***TOE***] shall only treat a certificate as a CA certificate if the following is met: the basicConstraints extension is present and the CA flag is set to TRUE.

5.1.3.2 Extended: X.509 Certificate Use and Management (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec exchanges, , and [***no additional uses***].

FIA_X509_EXT.2.2

When a connection to determine the validity of a certificate cannot be established, the [***TOE***] shall [***allow the administrator to choose whether to accept the certificate in these cases***].

FIA_X509_EXT.2.3

The [***TOE***] shall not establish an SA if a certificate or certificate path is deemed invalid.

5.1.4 Security management (FMT)

5.1.4.1 Specification of Management Functions (FMT_SMF.1(1))

FMT_SMF.1(1).1

The TOE shall be capable of performing the following management functions:
- Specify VPN gateways to use for connections,
- load X.509v3 certificates used by the security functions in this PP,
- Specify client credentials to be used for connections,
- [**[see FMT_SMF.1(2)].**].

5.1.4.2 Specification of Management Functions (FMT_SMF.1(2))

FMT_SMF.1(2).1

The [***TOE Platform***, ***VPN Gateway***] shall be capable of performing the following management functions:
- Configuration of IKE protocol version(s) used,
- Configure IKE authentication techniques used,
- Configure the cryptoperiod for the established session keys. The unit of measure for configuring the cryptoperiod shall be no greater than an hour,
- Specify the algorithm suites that may be proposed and accepted during the IPsec exchanges,
- load X.509v3 certificates used by the security functions in this PP,
- Configure certificate revocation check,
- ability to update the TOE, and to verify the updates,
- ability to configure all security management functions identified in other sections of this PP,
- ability to configure the reference identifier for the peer, (per TD0037)
- [***no other actions***].

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Extended: TSF Self Test (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

The [***TOE***] shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2

The [***TOE***] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [**digital signature**].

5.1.5.2 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The [***TOE***] shall provide the ability to query the current version of the TOE firmware/software

FPT_TUD_EXT.1.2

The [***TOE Platform***] shall provide the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

The [***TOE Platform***] shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [***no other functions***] prior to installing those updates.

5.1.6 Trusted path/channels (FTP)

5.1.6.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The [***TOE***] shall use IPsec to provide a trusted communication channel between itself and a VPN Gateway that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The [***TOE***] shall permit the TSF to initiate communication via the trusted channel.

FTP_ITC.1.3

The [***TOE***] shall initiate communication via the trusted channel for all traffic traversing that connection.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the EAL 1 components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 2 EAL 1 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

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

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

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

ADV_FSP.1.1e

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

ADV_FSP.1.2e

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

5.2.2 Guidance documents (AGD)

5.2.2.1 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 fulfil 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.

5.2.2.2 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.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

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

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

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

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

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

AVA_VAN.1.3e

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 Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

The TOE implements the IPsec protocol as specified in RFC 4301; however the TOE relies upon the VPN Gateway to ensure that the cryptographic algorithms and key sizes negotiated during the IKEv2 negotiation ensure that the security strength of the IKE_SA is greater than or equal to that of the CHILD_SA.[1]

The TOE implements RFC 4307, IKEv2 in tunnel mode only. The TOE does not offer transport mode as a configuration option. The TOE implements peer authentication using RSA certificates or ECDSA certificates that conform to RFC 4945. If certificates are used, the TOE ensures that the distinguished name (DN) contained in a certificate matches the expected DN for the entity attempting to establish a connection and ensures that the certificate has not been revoked (using the Online Certificate Status Protocol [OCSP] in accordance with RFC 2560).

The SHA hash algorithm (all claimed key sizes) is used as part of HMAC, but is also used independently as part of digital signature creation and verification. The TOE generates RSA and ECDSA signatures during IKEv2 peer authentication. The TOE verifies RSA & ECDSA signatures during IKEv2 peer authentication and trusted updates.

The TOE implements various HMAC algorithms to be used for authentication with ESP and IKEv2. The specific algorithms used depend upon the ciphersuite being used. The TOE implements AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256 as ESP encryption algorithms and implements HMAC-SHA1 as the authentication algorithm. The TOE implements AES-CBC-128 and AES-CBC-256 as IKEv2 encryption algorithms and implements HMAC-SHA1, HMAC-SHA-256 and HMAC-SHA-384 as the authentication algorithm.

As configured on the VPN Gateway, the TOE supports the following Diffie-Hellman (DH) groups for use in SA negotiation:

- DH 14 (2048-bit MODP),
- 19 (256-bit Random ECP), and
- 20 (384-bit Random ECP).

The TOE generates the secret value x used in the IKEv2 Diffie-Hellman key exchange ('x' in g^x mod p) using the CAVP tested RBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256, or 384 bits. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life of a specific IPsec SA is less than 1 in 2^{112}, 2^{128}, or 2^{192}.

The TOE implements minimal SPD rules that are defined implicitly through the configuration and connection of a VPN session. The TOE does not support direct editing of SPD rules. The PROTECT and BYPASS rules are implicit and are implemented by configuring a split tunnel. The administrator may configure PROTECT and BYPASS rules by enabling or disabling split-tunnel mode in the VIA connection profile on the VPN Gateway. If split-tunnel is disabled, all traffic follows the PROTECT rule. All traffic originating from the client operating system

[1] Note that the algorithm negotiated will be AES because that is the only available algorithm, so strength is based solely upon key size where more bits are stronger.

is passed through the tunnel established by the VIA client to the VPN Gateway. With split-tunneling enabled, the VPN Gateway pushes routes configured with the tunnel address command in the VIA connection profile on the VPN Gateway to the VIA client. Traffic matching the routes is forwarded through the IPsec tunnel. The DISCARD rule is not supported by the TOE and must be provided by firewall rules configured on the VPN Gateway and the Platform.

During the Peer Authentication stage of IPsec, the TOE will verify the authenticity of the VPN gateway's X.509v3 certificate by validating the certificate, validating the certificate path, validating the certificates revocation status using OCSP, validating that the certificate path terminates in a trusted CA certificate, and validating that the CA certificate has the basicConstraints extension present and the CA flag set to true.

The TOE performs cryptographic algorithms using the Aruba Common Cryptographic Module (ACCM) library in accordance with the following NIST standards and has received the following CAVP algorithm certificates.

Functions	Requirement	Windows	Linux (32/64 bit)	Android
Cryptographic key generation				

• RSA IFC Key Gen (2048 bits)
• ECDSA ECC Key Gen (NIST curves P-256, P-384)
• DSA FFC Key Gen
	FCS_CKM.1(1)
	FCS_CKM.1(2)
		2860
		1405
		1380
			1484, 2793
			500, 1356
			1351, 1353
				2860
				1405
				1380

Cryptographic key establishment				

• ECC-based key exchange
• FFC-based key exchange
	FCS_CKM.1(1)
	FCS_CKM.1(2)
		176	159, 161	176

Encryption/Decryption				
AES CBC, GCM (128 and 256 bits)	FCS_COP.1(1)	5345	2746, 5228	5345
Cryptographic signature services				
• RSA Digital Signature Algorithm (rDSA) (2048 bits) • ECDSA Digital Signature Algorithm (NIST curves P-256, P-384)	FCS_COP.1(2)	2860 1405	1484, 2793 500, 1356	2860 1405
Cryptographic hashing				
SHA-1, SHA-256, and SHA-384 (digest sizes 160, 256, and 384 bits)	FCS_COP.1(3)	4296	2317, 4208	4296
Keyed-hash message authentication				
HMAC-SHA-1 SHA-256, and SHA-384 (key and digest sizes 160, 256, and 384bits)	FCS_COP.1(4)	3542	1722, 3460	3542
Random bit generation				
CTR_DRBG with a minimum of 256 bits of non-determinism	FCS_RBG_EXT.1	2065	498, 1994	2065

Table 3 CAVP Algorithms

While the TOE generally fulfills all of the NIST SP 800-56A requirements without extensions, the following table specifically identifies the "should", "should not", and "shall not" conditions from the publication along with an indication of how the TOE conforms to those conditions.

NIST SP800-56A Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation

NIST SP800-56A Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
5.4	should	Yes	Not applicable
5.5.1.1	should	Yes	Not applicable
5.5.2	should	Yes	Not applicable
5.6.2	should	Yes	Not applicable
5.6.2.1	should	Yes	Not applicable
5.6.2.2	should	Yes	Not applicable
5.6.2.3	should	Yes	Not applicable
5.6.3.1	should	Yes	Not applicable
5.6.3.2.1	should	Yes	Not applicable
5.6.4.1	shall not	No	Not applicable
5.6.4.2	shall not	No	Not applicable
5.6.4.2	should	Yes	Not applicable
5.6.4.3	should (first occurrence)	Yes	Not applicable
5.6.4.3	should (second occurrence)	Yes	Not applicable
5.8	shall not (first occurrence)	No	Not applicable
5.8	shall not (second occurrence)	No	Not applicable
6	should (first occurrence)	Yes	Not applicable
6	should (second occurrence)	Yes	Not applicable
7	shall not (first occurrence)	No	Not applicable
7	shall not (second occurrence)	No	Not applicable
9	shall not	No	Not applicable

Table 4 NIST SP800-56A Conformance

Aruba's VIA Client includes an AES-256 CTR_DRBG (irrespective of the underlying Platform) that seeds itself with 384-bits of entropy (composed of a 256-bit entropy_input and a 128-bit nonce) drawn from a Platform function intended to provide cryptographically secure randomness (and conditioned using SHA-1). The following list describes the mechanism by which VIA obtains its seeding. The Android and Windows interfaces are identified in their associated Security Targets.

- Linux - /dev/random
- Android- /dev/random
- Windows - BCryptGenRandom()

.

Based on NIAP's "Clarification to the Entropy Documentation and Assessment Annex[2]", Aruba assumes a minimum entropy of 0.67 bits of entropy per bit of data from platform provided sources. This minimum-entropy estimate along with knowledge that the seed is 384-bits means that the TOE seeds itself with at least 256-bits of entropy.

The following table describes the keys and secrets utilized by the TOE. The details apply to all platforms.

Key Name:	Origin/Purpose:	Storage Location:	Key Destruction:
DH Private Components	Used to derive the secret session key during DH key agreement protocol Group 14 (384 bits)	Temporarily in volatile RAM	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP

[2] https://www.niap-ccevs.org/MMO/GD/Entropy%20Documentation%20and%20Assessment%20Clarification.pdf

DRNG Seed Key	DRBGs for key generation DRBG Seed: SP800-90a DRBG (384 bits) DRBG Key: SP800-90a (256 bits)	Temporarily in volatile RAM	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
RSA Private Key	Used to create RSA digital signatures key size: 2048	TOE Platform Keystore	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
RSA Key Wrapping Private Key	Used for RSA Key Wrapping decryption Operation key size: 2048	Temporarily in volatile RAM	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
ECDSA Private Key	Used to create DSA digital signatures NIST curves : 256, 384	TOE Platform Keystore	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
AES Keys	Used during AES encryption, decryption, and CMAC operations Key length 128 192 256	Temporarily in volatile RAM	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
HMAC Keys	Used during HMAC SHA-1, 256, 384 operations Key size 160 256 384	Temporarily in volatile RAM	An application program which uses the API may destroy the key. The Key Destruction service zeroizes (zero overwrite) this CSP
DH Public Component	Used to derive the secret session key during DH key agreement protocol DH Groups: Group 14 (2048 bits)	Temporarily in volatile RAM	No longer needed by trusted channel
ECDH Public Component	Used to derive the secret session key during ECDH key agreement protocol Group 19 (P-256) , Group20	Temporarily in volatile RAM	N/A

	(P-384)		
RSA Public Keys	Used to verify RSA Signatures key size: 2048	Temporarily in volatile RAM	N/A
RSA Key Wrapping Public Keys	Used for RSA Key Wrapping encryption operation	Temporarily in volatile RAM	N/A
ECDSA Public Keys	Used to verify ECDSA Signatures NIST Curves : 256, 384	Temporarily in volatile RAM	N/A

Table 5 Key Destruction

The Cryptographic support function satisfies the following security functional requirements:

- FCS_CKM.1(1): The TOE supports asymmetric key generation of ECDSA (for curves P-256 and P-384) key pairs for key establishment.
- FCS_CKM.1(2): The TOE supports asymmetric key generation of RSA and ECDSA (for curves P-256 and P-384) key pairs for use with IKE peer authentication.
- FCS_CKM_EXT.2: **Table 5 Key Destruction** lists all the keys manipulated by the TOE. The TOE does not store these keys unencrypted in to persistent storage. While the TOE manipulates keys, on all platforms, the TOE platform's key storage is used. The TOE Platform's keystore is the Android Keystore, the Windows Certificate Store, or a Linux keyring in a file depending on the platform.
- FCS_CKM_EXT.4: The TOE destroys cryptographic keys when they are no longer in use by the system.
- FCS_COP.1(1): See **Table 3 CAVP Algorithms** for the supported cryptographic algorithms.
- FCS_COP.1(2): See **Table 3 CAVP Algorithms** for the supported cryptographic algorithms.
- FCS_COP.1(3): See **Table 3 CAVP Algorithms** for the supported cryptographic algorithms.
- FCS_COP.1(4): The TOE provides the HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 algorithms. Refer to **Table 3 CAVP Algorithms** for the corresponding CAVP certificate demonstrating compliance with these algorithms. These keyed-hash functions can be defined for use in an IPsec connection. The HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 algorithms are used with key sizes and block sizes of 160, 256 and 384 respectively, producing output MAC lengths equal to the block size.
- FCS_IPSEC_EXT.1: The TOE implements IPsec in accordance with FCS_IPSEC_EXT.1 as described above.
- FCS_RBG_EXT.1: The TOE provides a CAVP tested RBG.

6.2 User data protection

The TOE ensures that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing a network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

The User data protection function satisfies the following security functional requirements:

- FDP_RIP.2: The TOE ensures that previous information contents of resources used for new objects are not discernible in any new object, such as network packets, as described above.

6.3 Identification and authentication

The TOE can also use X.509 certificates for authentication. The TOE uses a user specified certificate when attempting to establish that VPN connection. The TOE validates authentication certificates (including the full path) and checks their revocation status using OCSP. The TOE processes a VPN connection to a server by first comparing the Identification (ID) Payload received from the server against the certificate sent by the server, and if the DN of the certificate does not match the ID, then the TOE does not establish the connection. Assuming the server's certificate matches the ID, the TOE then validates that it can construct a certificate path from the server's certificate through any intermediary CAs to the CA certificate specified by the user in the VPN configuration. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the certificates (e.g., checking its validity dates and that the CA flag is present in the basic constraints section for all CA certs). Assuming the certificates are valid, the TOE finally checks the revocation status of all certificates (starting with the server's certificate and working up the chain). Through the Aruba Mobility Controller, the administrator determines if a certificate is accepted or rejected if the connection to OCSP server cannot be reached. Section 6.1 describes how the TOE uses certificates in its IPsec architecture.

The Identification and authentication function satisfies the following security functional requirements:

- FIA_X509_EXT.1: This requirement is satisfied by the TOE.
- FIA_X509_EXT.2: The TOE uses X.509v3 certificates for authentication in IPsec exchanges and rejects any certificates that cannot be validated as described above. Through the Aruba Mobility Controller, the administrator determines if a certificate is accepted or rejected if the connection to OCSP server cannot be reached.

6.4 Security management

The following security management functions are provided directly by the TOE or implemented in the VPN gateway as indicated below:

- The TOE provides functions allowing the user to select VPN gateway and credentials used to connect to those gateways.
- The VPN Gateway (acting as an administrator) to which the TOE connects selects IKEv2 protocols and authentication techniques.
- The VPN Gateway (acting as an administrator) to which the TOE connects selects the algorithms to be used in IPsec exchanges.
- The VPN gateway provides the ability to configure the crypto-period for the established IPsec session keys (including the ability to configure crypto-periods less than an hour in duration).
- The VPN Gateway provides the ability to specify the reference identifier.
- The VPN Gateway provides the ability to configure certificate revocation checking.
- The TOE and TOE Platform provides the ability to load X.509v3 certificates used for VPN connections using IPsec.

- The TOE Platform provides the ability to update the TOE, and to verify the updates.

The Security management function satisfies the following security functional requirements:

- FMT_SMF.1(1): The TOE provides the functions necessary to specify VPN gateways and the corresponding credentials used to establish VPN connections as described above.
- FMT_SMF.1(2): A combination of the TOE Platform and the VPN gateway provide the functions necessary to manage the security functions described in this security target as described above.

6.5 Protection of the TSF

The TOE performs known answer power on self-tests (POST) on its cryptographic algorithms to ensure that they are functioning correctly. The TOE utilizes the Aruba Common Cryptographic Module (ACCM) library which implements known answer tests on its cryptographic algorithms to ensure they are working correctly. These known answer tests involve using the ACCM library functions to encrypt blocks of data and comparing the resulting encrypted block of data to a block that is known to be correct. The result of encrypting a block of data is the same every time if the encryption library operates properly. These tests cover the following algorithms, known answers tests, and pairwise consistency tests:

- AES-GCM,
- AES-CBC,
- SHA-1,
- SHA-256,
- SHA-384,
- HMAC-SHA1,
- HMAC-SHA-256,
- HMAC-SHA-384,
- RSA Pairwise Consistency Test,
- RSA Encrypt/Decrypt Known Answer Test,
- DSA Pairwise Consistency Test,
- ECDSA Pairwise Consistency Test,
- ECDH Pairwise Consistency Test,
- DH Pairwise Consistency Test, and
- FIPS 186-2 RNG Known Answer Test.

The TOE invokes these self-tests of the ACCM library at start to ensure that those cryptographic algorithms are working correctly. If any self-test fails, the TOE will not start.

The TOE ***About*** tab on the VIA Client show the current system image version number.

The TOE Platforms support loading updates by the administrator. For Android versions, the application and signature is provided to and verified by the Google Play Store. The TOE platform checks the signature against the downloaded application when the update is obtained by the TOE platform. For Linux and Windows platforms, the administrator obtains the update in the form of an installer program from Aruba. The installer automatically verifies the digital signature on the update during the installation. An unverified update cannot be installed.

The Protection of the TSF function satisfies the following security functional requirements:

- FPT_TST_EXT.1: The TOE performs a suite of self-tests to verify its integrity.
- FPT_TUD_EXT.1: The TOE Platform provides a means for obtaining and installing digitally signed updates.

6.6 Trusted path/channels

See section 6.1 for a description of how the TOE can establish IPsec VPN connections with configured VPN gateways. The resulting VPNs ensure that both ends of the channel are authenticated and the channel protects data from disclosure and modification.

The Trusted path/channels function satisfies the following security functional requirements:

- FTP_ITC.1: The TOE uses IPsec to provide a protected communication channel between itself and an IPsec VPN gateway. The TOE uses IKEv2 and not IKEv1. The channel provides assurance identification of the end points and protects transmitted data from disclosure and modification.