DataSoft Secure Tactical VPN Client for Android Security Target

Version 0.5
08/07/23

Prepared for:

DataSoft Corporation

10235 S. 51st Street, Suite 115
Phoenix, AZ 85044

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 DataSoft Secure Tactical VPN Client for Android provided by DataSoft Corporation. The TOE is being evaluated as a software application.

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 parenthetical number 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.
  - 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 …").
- 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 – DataSoft Secure Tactical VPN Client for Android Security Target

ST Version – Version 0.5

ST Date – 08/07/23

1.2 TOE Reference

TOE Identification –DataSoft Secure Tactical VPN Client for Android

TOE Developer – DataSoft Corporation

Evaluation Sponsor – DataSoft Corporation

1.3 TOE Overview

The Target of Evaluation (TOE) is the DataSoft Secure Tactical VPN Client for Android (SW version 2.3.7). The TOE enables remote users within an organization to communicate securely as if their devices were directly connected to a private network.

1.4 TOE Description

The TOE provides IPsec VPN client functionality for Android-based End User Devices (EUD) running on Android 11, Android 12, and Android 13 mobile devices (or (Platforms") running Linux Kernel earlier than v5.6. There are a number of evaluated mobile Android devices using this version of Android (e.g., devices by Samsung, Google, Motorola, Panasonic, Microsoft, Zebra, etc.).

The TOE was specifically tested on those three different versions of Android using the following hardware

Phone	Model	CPU	Kernel	Android OS	VID/Date
Samsung	S20 Tactical Edition	Qualcomm snapdragon 865 (SM8250)	4.19	Android 11	11042/ Archived
Google	Pixel 5	Qualcomm snapdragon 765G (SM7250)	4.19	Android 11	11124/ Archived
Google	Pixel 4a-5G	Qualcomm snapdragon 765G (SM7250)	4.19	Android 12	11239/ 02/28/2022
Google	Pixel 5a-5G	Qualcomm snapdragon 765G (SM7250)	4.19	Android 13	11317/ 01/24/2023

The TOE complies with IKEv2 RFCs and can utilize X509v3 certificates for authentication of an IPsec peer. In a basic IPsec VPN connection, all traffic from the VPN client is encrypted and sent across the VPN gateway. Administrators can define profiles through the TOE or load them into a mobile device. Named profiles define the endpoints, authentication data, and cryptographic characteristics for a VPN connection. Profiles define the cryptographic configuration of the set of additional cryptographic options.

The TOE can interoperate with IKEv2 VPN Gateways but also includes extensions to route multicast traffic through the VPN, allowing the TOE to interoperate with DataSoft's small form factor Radio Access Point (RAP), which allows mobile and dismounted operators to perform C2-releated computing functions security across existing tactical communications networks.

1.4.1 TOE Architecture

The TOE product consists of a user space application installed as a standard Android APK.

1.4.1.1 Physical Boundaries

The TOE consists of a software-only VPN client application. The underlying mobile platform on which the TOE executes belongs to the IT environment.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the DataSoft Secure Tactical VPN Client for Android:

- Cryptographic support
- User data protection
- Identification and authentication

- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

1.4.1.2.1 Cryptographic support

The TOE includes its own cryptographic library that implements approved cryptographic algorithms that the TOE uses to protect communication between itself and a VPN gateway over an unprotected network using IPsec. The TOE uses the Platform to protect credential data at rest.

The TOE platform provides asymmetric cryptography (Android's user keychain), which is used by the TOE for IKE peer authentication (using digital signature and hashing services). In addition, the TOE seeds its DRBG from the Platform.

1.4.1.2.2 User data protection

The TOE ensures that residual information from previously sent network packets processed through the platform are protected from being passed into subsequent network packets.

1.4.1.2.3 Identification and authentication

The TOE and TOE platform perform device-level X.509 certificate-based authentication of the VPN Gateway during IKE v2 key exchange. Device-level authentication allows the TOE to establish a secure channel with a trusted VPN Gateway. The secure channel is established only after each endpoint successfully authenticates each other.

1.4.1.2.4 Security management

The TOE provides all the interfaces necessary to manage the security functions identified throughout this Security Target. This includes interfaces to the user as well as to the VPN gateway. The IPsec VPN is fully configurable by a combination of functions provided directly by the TOE and those available to the associated VPN gateway. The TOE platform provides the functions necessary to securely update the TOE.

1.4.1.2.5 Privacy

The TOE does not store or transmit Personally Identifiable Information (PII) over a network.

1.4.1.2.6 Protection of the TSF

The TOE utilizes its own cryptographic functions to perform self-tests that ensure the TOE's integrity and algorithm correctness. The TOE platform provides the functions necessary to securely update the TOE software.

1.4.1.2.7 Trusted path/channels

The TOE establishes an IPsec trusted channel (which protects the transmitted data from unauthorized disclosure and modification) with a corresponding VPN gateway.

1.4.2 TOE Documentation

DataSoft Common Criteria Guide for the VPN Client, Version 1.1, July 26, 2023 (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 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Extended
- PP-Configuration for Application Software and Virtual Private Network (VPN) Clients, Version 1.3, 7 April 2023, which includes the following components:
    - Base-PP: Protection Profile for Application Software, Version 1.4 (PP_APP_V1.4)
    - PP-Module: PP-Module for Virtual Private Network (VPN) Clients, Version 2.4 (MOD_VPNC_V2.4)
- Technical Decisions

Package	Technical Decision	Applied	Notes
MOD_VPNC_V24	TD0725 – Correction to FCS_CKM_EXT.2/4 selections	Yes	
MOD_VPNC_V2.4	TD0711 - FMT_SMF.1 direction when using MDF 3.3	No	Not claiming MDF 3.3
MOD_VPNC_V2.4	TD0697 - Alignment with App PP V1.4 for required NIST curves in FCS_CKM.1/AK	Yes	
MOD_VPNC_V2.4	TD0690 - Missing EAs for FDP_VPN_EXT.1	No	FDP_VPN_EXT.1 not claimed
MOD_VPNC_V2.4	TD0687 - MOD_VPNC FTP_DIT_EXT.1 Alignment for App PP 1.4	Yes	
MOD_VPNC_V2.4	TD0672 - VPN Client PP-Module updated to allow for new PP and PP-Module Versions	Yes	
MOD_VPNC_V2.4	TD0662 - Changes to Testing IPsec NAT Transversal and XAUTH in MOD_VPNC 2.4	Yes	
MOD_VPNC_V2.4	TD0647 - Table 2 Applicability	Yes	
PP_APP_v1.4	TD0756 – Update for platform-provided full disk encryption	Yes	
PP_APP_v1.4	TD0743 – FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
PP_APP_v1.4	TD0736 - Number of elements for iterations of FCS_HTTPS_EXT.1	No	HTTP not claimed
PP_APP_v1.4	TD0719 - ECD for PP APP V1.3 and 1.4	Yes	
PP_APP_v1.4	TD0717 - Format changes for PP_APP_V1.4	Yes	
PP_APP_v1.4	TD0669 - FIA_X509_EXT.1 Test 4 Interpretation	Yes	
PP_APP_v1.4	TD0664 - Testing activity for	Yes	

	FPT_TUD_EXT.2.2		
PP_APP_v1.4	TD0650 - Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	Yes	
PP_APP_v1.4	TD0628 - Addition of Container Image to Package Format	Yes	
PP_APP_v1.4	TD0624 - Addition of DataStore for Storing and Setting Configuration Options	Yes	

2.1 Conformance Rationale

The ST conforms to the Protection Profile for Application Software, Version 1.4, 07 October 2021 (ASPP14) and the PP-Module for Virtual Private Network (VPN) Clients, Version 2.4, 31 March 2022 (VPNC24). 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 ASPP14/VPNC24 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The ASPP14/VPNC24 offers additional information about the identified security objectives, but that has not been reproduced here and the ASPP14/VPNC24 should be consulted if there is interest in that material.

In general, the ASPP14/VPNC24 has defined Security Objectives appropriate for a software VPN client application and as such are applicable to the DataSoft Secure Tactical VPN Client for Android 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 environment.

OE.PLATFORM The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.

OE.PROPER_ADMIN The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

OE.PROPER_USER The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.

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

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the ASPP14/VPNC24. The ASPP14/VPNC24 defines the following extended requirements and since they are not redefined in this ST the ASPP14/VPNC24 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- ASPP14:FCS_CKM_EXT.1: Cryptographic Key Generation Services
- VPNC24:FCS_CKM_EXT.2: Cryptographic Key Storage – per TD0725
- VPNC24:FCS_CKM_EXT.4: Cryptographic Key Destruction – per TD0725
- VPNC24:FCS_IPSEC_EXT.1: IPsec - per TD0662
- ASPP14:FCS_RBG_EXT.1: Random Bit Generation Services
- ASPP14:FCS_RBG_EXT.2: Random Bit Generation from Application
- ASPP14:FCS_STO_EXT.1: Storage of Credentials
- ASPP14:FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
- ASPP14:FDP_DEC_EXT.1: Access to Platform Resources
- ASPP14:FDP_NET_EXT.1: Network Communications
- ASPP14:FIA_X509_EXT.1: X.509 Certificate Validation - per TD0669
- ASPP14:FIA_X509_EXT.2: X.509 Certificate Authentication
- VPNC24:FIA_X509_EXT.2: X.509 Certificate Authentication
- ASPP14:FMT_CFG_EXT.1: Secure by Default Configuration
- ASPP14:FMT_MEC_EXT.1: Supported Configuration Mechanism - per TD0624
- ASPP14:FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
- ASPP14:FPT_AEX_EXT.1: Anti-Exploitation Capabilities
- ASPP14:FPT_API_EXT.1: Use of Supported Services and APIs
- ASPP14:FPT_IDV_EXT.1: Software Identification and Versions
- ASPP14:FPT_LIB_EXT.1: Use of Third Party Libraries
- VPNC24:FPT_TST_EXT.1/VPN: TSF Self-Test
- ASPP14:FPT_TUD_EXT.1: Integrity for Installation and Update
- ASPP14:FPT_TUD_EXT.2: Integrity for Installation and Update - per TD0628
- ASPP14:FTP_DIT_EXT.1: Protection of Data in Transit - per TD0743
- VPNC24:FTP_DIT_EXT.1: Protection of Data in Transit

Extended SARs:

- ALC_TSU_EXT.1: Timely Security Updates

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 ASPP14/VPNC24. The refinements and operations already performed in the ASPP14/VPNC24 are not identified (e.g., highlighted) here, rather the requirements have been copied from the ASPP14/VPNC24 and any residual operations have been completed herein. Of particular note, the ASPP14/VPNC24 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 ASPP14/VPNC24. The ASPP14/VPNC24 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by DataSoft Secure Tactical VPN Client for Android TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	VPNC24:FCS_CKM.1: Cryptographic Key Generation Services
	ASPP14:FCS_CKM.1/AK: Cryptographic Asymmetric Key Generation - per TD0717
	VPNC24:FCS_CKM.1/AK: Cryptographic Asymmetric Key Generation
	ASPP14:FCS_CKM.1/SK: Cryptographic Symmetric Key Generation
	VPNC24:FCS_CKM.1/VPN: VPN Cryptographic Key Generation (IKE)
	ASPP14:FCS_CKM.2: Cryptographic Key Establishment
	VPNC24:FCS_CKM.2: Cryptographic Key Establishment
	ASPP14:FCS_CKM_EXT.1: Cryptographic Key Generation Services - per TD0717
	VPNC24:FCS_CKM_EXT.2: Cryptographic Key Storage – per TD0725
	VPNC24:FCS_CKM_EXT.4: Cryptographic Key Destruction – per TD0725
	ASPP14:FCS_COP.1/Hash: Cryptographic Operation - Hashing - per TD0717
	ASPP14:FCS_COP.1/KeyedHash: Cryptographic Operation - Keyed-Hash Message Authentication - per TD0717
	ASPP14:FCS_COP.1/Sig: Cryptographic Operation - Signing - per TD0717
	ASPP14:FCS_COP.1/SKC: Cryptographic Operation - Encryption/Decryption - per TD0717
	VPNC24:FCS_COP.1/SKC: Cryptographic Operation
	VPNC24:FCS_IPSEC_EXT.1: IPsec - per TD0662
	ASPP14:FCS_RBG_EXT.1: Random Bit Generation Services
	ASPP14:FCS_RBG_EXT.2: Random Bit Generation from Application
	ASPP14:FCS_STO_EXT.1: Storage of Credentials
FDP: User data protection	ASPP14:FDP_DAR_EXT.1: Encryption Of Sensitive Application Data

	ASPP14:FDP_DEC_EXT.1: Access to Platform Resources
	ASPP14:FDP_NET_EXT.1: Network Communications
	VPNC24:FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	ASPP14:FIA_X509_EXT.1: X.509 Certificate Validation - per TD0669
	ASPP14:FIA_X509_EXT.2: X.509 Certificate Authentication
	VPNC24:FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security management	ASPP14:FMT_CFG_EXT.1: Secure by Default Configuration
	ASPP14:FMT_MEC_EXT.1: Supported Configuration Mechanism - per TD0624
	ASPP14:FMT_SMF.1: Specification of Management Functions
	VPNC24:FMT_SMF.1/VPN: Specification of Management Functions (VPN)
FPR: Privacy	ASPP14:FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
FPT: Protection of the TSF	ASPP14:FPT_AEX_EXT.1: Anti-Exploitation Capabilities
	ASPP14:FPT_API_EXT.1: Use of Supported Services and APIs
	ASPP14:FPT_IDV_EXT.1: Software Identification and Versions
	ASPP14:FPT_LIB_EXT.1: Use of Third Party Libraries
	VPNC24:FPT_TST_EXT.1/VPN: TSF Self-Test
	ASPP14:FPT_TUD_EXT.1: Integrity for Installation and Update
	ASPP14:FPT_TUD_EXT.2: Integrity for Installation and Update - per TD0628
FTP: Trusted path/channels	ASPP14:FTP_DIT_EXT.1: Protection of Data in Transit - per TD0743
	VPNC24:FTP_DIT_EXT.1: Protection of Data in Transit

Table 1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic Key Generation Services (VPNC24:FCS_CKM.1)

VPNC24:FCS_CKM.1.1

The application shall [***implement asymmetric key generation***].

5.1.1.2 Cryptographic Asymmetric Key Generation - per TD0717 (ASPP14:FCS_CKM.1/AK)

ASPP14:FCS_CKM.1.1/AK

The application shall [***implement functionality***] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [***[ECC schemes] using 'NIST curves' P-384 and [P-256] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4***].

5.1.1.3 Cryptographic Asymmetric Key Generation (VPNC24:FCS_CKM.1/AK)

VPNC24:FCS_CKM.1.1/AK

The application shall [***implement functionality***] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm
- ECC schemes using 'NIST curves' P-384 and [***P-256***] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4, and
[***- no other key generation methods***]
(TD0697 applied)

5.1.1.4 Cryptographic Symmetric Key Generation (ASPP14:FCS_CKM.1/SK)

ASPP14:FCS_CKM.1.1/SK

The application shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes [***256 bit***].

5.1.1.5 VPN Cryptographic Key Generation (IKE) (VPNC24:FCS_CKM.1/VPN)

VPNC24:FCS_CKM.1.1/VPN

The TSF shall [***invoke platform-provided functionality***] to generate asymmetric cryptographic keys used for IKE peer authentication in accordance with:
[***- 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.

5.1.1.6 Cryptographic Key Establishment (ASPP14:FCS_CKM.2)

ASPP14:FCS_CKM.2.1

The application shall [***implement functionality***] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'***].

5.1.1.7 Cryptographic Key Establishment (VPNC24:FCS_CKM.2)

VPNC24:FCS_CKM.2.1

The application shall [***implement functionality***] to perform cryptographic key establishment in accordance with a specified key establishment method:
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'; and
[***- No other schemes***].

5.1.1.8 Cryptographic Key Generation Services - per TD0717 (ASPP14:FCS_CKM_EXT.1)

ASPP14:FCS_CKM_EXT.1.1

The application shall [***implement asymmetric key generation***].

5.1.1.9 Cryptographic Key Storage – per TD0725 (VPNC24:FCS_CKM_EXT.2)

VPNC24:FCS_CKM_EXT.2.1

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

5.1.1.10 Cryptographic Key Destruction – per TD0725 (VPNC24:FCS_CKM_EXT.4)

VPNC24:FCS_CKM_EXT.4.1

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

5.1.1.11 Cryptographic Operation - Hashing - per TD0717 (ASPP14:FCS_COP.1/Hash)

ASPP14:FCS_COP.1.1/Hash

The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-256, SHA-384, SHA-512***] and message digest sizes [***256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.1.12 Cryptographic Operation - Keyed-Hash Message Authentication - per TD0717 (ASPP14:FCS_COP.1/KeyedHash)

ASPP14:FCS_COP.1.1/KeyedHash

The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] and [***no other algorithms***] with key sizes [**256, 384, 512**] and message digest sizes [***256, 384, 512***] and [***no other size***] bits that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

5.1.1.13 Cryptographic Operation - Signing - per TD0717 (ASPP14:FCS_COP.1/Sig)

ASPP14:FCS_COP.1.1/Sig

The application shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [***ECDSA schemes using 'NIST curves' P-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6***].

5.1.1.14 Cryptographic Operation - Encryption/Decryption - per TD0717 (ASPP14:FCS_COP.1/SKC)

ASPP14:FCS_COP.1.1/SKC

The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm
[***AES-CBC (as defined in NIST SP 800-38A) mode, AES-GCM (as defined in NIST SP 800-38D) mode***]
and cryptographic key sizes [***128-bit, 256-bit***].

5.1.1.15 Cryptographic Operation (VPNC24:FCS_COP.1/SKC)

VPNC24:FCS_COP.1.1/SKC

The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm
- AES-CBC (as defined in NIST SP 800-38A) mode
- AES-GCM (as defined in NIST SP 800-38D) mode
and
[***- no other modes***]
and cryptographic key sizes 128-bit, 256-bit.

5.1.1.16 IPsec - per TD0662 (VPNC24:FCS_IPSEC_EXT.1)

VPNC24:FCS_IPSEC_EXT.1.1

The TSF shall implement the IPsec architecture as specified in RFC 4301.

VPNC24:FCS_IPSEC_EXT.1.2

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

VPNC24:FCS_IPSEC_EXT.1.3

The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

VPNC24:FCS_IPSEC_EXT.1.4

The TSF 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, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC***].

VPNC24:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol:
[***IKEv2 as defined in RFCs 7296 (with mandatory support for NAT traversal as specified in section 2.23), RFC 8784, RFC 8247, and [RFC 4868 for hash functions]***].

VPNC24:FCS_IPSEC_EXT.1.6

The TSF 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 [***AES-GCM-128 as specified in RFC 5282, AES-GCM- 256 as specified in RFC 5282***].

VPNC24:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [***IKEv2 SA lifetimes can be configured by [a VPN Gateway] based on [length of time]***].

If length of time is used, it must include at least one option that is 24 hours or less for Phase 1 SAs and 8 hours or less for Phase 2 SAs.

VPNC24:FCS_IPSEC_EXT.1.8

The TSF shall ensure that all IKE protocols implement DH groups 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and [***[no other DH Groups] according to RFC 5114***].

VPNC24:FCS_IPSEC_EXT.1.9

The TSF shall generate the secret value x used in the IKE DH 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 [**256, 384**] bits.

VPNC24:FCS_IPSEC_EXT.1.10

The TSF 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^ [**128, 192**].

VPNC24:FCS_IPSEC_EXT.1.11

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

VPNC24:FCS_IPSEC_EXT.1.12

The TSF shall not establish an SA if the [***IP address, Fully Qualified Domain Name (FQDN)***] and [***no other reference identifier type***] contained in a certificate does not match the expected value(s) for the entity attempting to establish a connection.

VPNC24:FCS_IPSEC_EXT.1.13

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

VPNC24:FCS_IPSEC_EXT.1.14

The [***VPN Gateway***] 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.

5.1.1.17 Random Bit Generation Services (ASPP14:FCS_RBG_EXT.1)

ASPP14:FCS_RBG_EXT.1.1

The application shall [***implement DRBG functionality***] for its cryptographic operations.

5.1.1.18 Random Bit Generation from Application (ASPP14:FCS_RBG_EXT.2)

ASPP14:FCS_RBG_EXT.2.1

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

ASPP14:FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [***no other source***] 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.

5.1.1.19 Storage of Credentials (ASPP14:FCS_STO_EXT.1)

ASPP14:FCS_STO_EXT.1.1

The application shall [***invoke the functionality provided by the platform to securely store [IKEv2 Auth private keys and certificates]***] to non-volatile memory.

5.1.2 User data protection (FDP)

5.1.2.1 Encryption Of Sensitive Application Data - per TD0756 (ASPP14:FDP_DAR_EXT.1)

ASPP14:FDP_DAR_EXT.1.1

The application shall [***not store any sensitive data***] in non-volatile memory.

5.1.2.2 Access to Platform Resources (ASPP14:FDP_DEC_EXT.1)

ASPP14:FDP_DEC_EXT.1.1

The application shall restrict its access to [***network connectivity***].

ASPP14:FDP_DEC_EXT.1.2

The application shall restrict its access to [***no sensitive information repositories***].

5.1.2.3 Network Communications (ASPP14:FDP_NET_EXT.1)

ASPP14:FDP_NET_EXT.1.1

The application shall restrict network communication to [***user-initiated communication for [IPsec VPN connections to a VPN GW]***].

5.1.2.4 Full Residual Information Protection (VPNC24:FDP_RIP.2)

VPNC24:FDP_RIP.2.1

The [***TOE***] shall ensure 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 X.509 Certificate Validation - per TD0669 (ASPP14:FIA_X509_EXT.1)

ASPP14:FIA_X509_EXT.1.1

The application shall [***implement functionality***] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [***CRL as specified in RFC 8603***]
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:

  o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

  o Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.

o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
o S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
o Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kpcmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

ASPP14:FIA_X509_EXT.1.2

The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

5.1.3.2 X.509 Certificate Authentication (ASPP14:FIA_X509_EXT.2)

ASPP14:FIA_X509_EXT.2.1

The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***IPsec***].

ASPP14:FIA_X509_EXT.2.2

When the application cannot establish a connection to determine the validity of a certificate, the application shall [***not accept the certificate***].

5.1.3.3 X.509 Certificate Authentication (VPNC24:FIA_X509_EXT.2)

VPNC24:FIA_X509_EXT.2.1

The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and no other protocols.

VPNC24:FIA_X509_EXT.2.2

When the application cannot establish a connection to determine the validity of a certificate, the TSF shall [***not accept the certificate***].

5.1.4 Security management (FMT)

5.1.4.1 Secure by Default Configuration (ASPP14:FMT_CFG_EXT.1)

ASPP14:FMT_CFG_EXT.1.1

The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

ASPP14:FMT_CFG_EXT.1.2

The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

5.1.4.2 Supported Configuration Mechanism - per TD0624 (ASPP14:FMT_MEC_EXT.1)

ASPP14:FMT_MEC_EXT.1.1

The application shall [***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options***].

5.1.4.3 Specification of Management Functions (ASPP14:FMT_SMF.1)

ASPP14:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [***no management functions***].

5.1.4.4 Specification of Management Functions (VPN) (VPNC24:FMT_SMF.1/VPN)

VPNC24:FMT_SMF.1.1/VPN

The TSF shall be capable of performing the following management functions:
[- ***Specify VPN gateways to use for connections,***
- Specify client credentials to be used for connections,
- Configure the reference identifier of the peer]

5.1.5 Privacy (FPR)

5.1.5.1 User Consent for Transmission of Personally Identifiable (ASPP14:FPR_ANO_EXT.1)

ASPP14:FPR_ANO_EXT.1.1

The application shall [***not transmit PII over a network***].

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Anti-Exploitation Capabilities (ASPP14:FPT_AEX_EXT.1)

ASPP14:FPT_AEX_EXT.1.1

The application shall not request to map memory at an explicit address except for [**no exceptions**].

ASPP14:FPT_AEX_EXT.1.2

The application shall [***not allocate any memory region with both write and execute permissions***].

ASPP14:FPT_AEX_EXT.1.3

The application shall be compatible with security features provided by the platform vendor.

ASPP14:FPT_AEX_EXT.1.4

The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

ASPP14:FPT_AEX_EXT.1.5

The application shall be built with stack-based buffer overflow protection enabled.

5.1.6.2 Use of Supported Services and APIs (ASPP14:FPT_API_EXT.1)

ASPP14:FPT_API_EXT.1.1

The application shall use only documented platform APIs.

5.1.6.3 Software Identification and Versions (ASPP14:FPT_IDV_EXT.1)

ASPP14:FPT_IDV_EXT.1.1

The application shall be versioned with [***[an APK version number]***]

5.1.6.4 Use of Third Party Libraries (ASPP14:FPT_LIB_EXT.1)

ASPP14:FPT_LIB_EXT.1.1

The application shall be packaged with only [**OpenSSL library**].

5.1.6.5 TSF Self-Test (VPNC24:FPT_TST_EXT.1/VPN)

VPNC24:FPT_TST_EXT.1.1/VPN

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

VPNC24:FPT_TST_EXT.1.2/VPN

The [***TOE platform***] 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 verification**].

5.1.6.6 Integrity for Installation and Update (ASPP14:FPT_TUD_EXT.1)

ASPP14:FPT_TUD_EXT.1.1
The application shall [***leverage the platform***] to check for updates and patches to the application software.

ASPP14:FPT_TUD_EXT.1.2
The application shall [***leverage the platform***] to query the current version of the application software.

ASPP14:FPT_TUD_EXT.1.3
The application shall not download, modify, replace or update its own binary code.

ASPP14:FPT_TUD_EXT.1.4
Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

ASPP14:FPT_TUD_EXT.1.5
The application is distributed [***as an additional software package to the platform OS***].

5.1.6.7 Integrity for Installation and Update - per TD0628 (ASPP14:FPT_TUD_EXT.2)

ASPP14:FPT_TUD_EXT.2.1
The application shall be distributed using [***the format of the platform-supported package manager***].

ASPP14:FPT_TUD_EXT.2.2
The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

ASPP14:FPT_TUD_EXT.2.3
The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Protection of Data in Transit - per TD0743 (ASPP14:FTP_DIT_EXT.1)

ASPP14:FTP_DIT_EXT.1.1
The application shall [***encrypt all transmitted [data] with [IPsec as defined in the PP-Module for VPN Client for [all transmitted data]]***] between itself and another trusted IT product. (TD0743 applied)

5.1.7.2 Protection of Data in Transit (VPNC24:FTP_DIT_EXT.1)

VPNC24:FTP_DIT_EXT.1.1
The application shall encrypt all transmitted sensitive data using IPsec as specified in FCS_IPSEC_EXT.1 and [***no other protocols***] between itself and another trusted IT product. (TD0687 applied)

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the 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
	ALC_TSU_EXT.1: Timely Security Updates
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 2 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 categorization 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 securityrelevant 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.

5.2.2.2 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 application shall be labelled with a 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 description in the TSS of how timely security updates are made to the TOE. Application developers must support updates to their products for purposes of fixing security vulnerabilities.

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.

5.2.3.3 Timely Security Updates (ALC_TSU_EXT.1)

ALC_TSU_EXT.1.1d

The developer shall provide a description in the TSS of how timely security updates are made to the TOE. Note: Application developers must support updates to their products for purposes of fixing security vulnerabilities.

ALC_TSU_EXT.1.2d

The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

ALC_TSU_EXT.1.1c

The description shall include the process for creating and deploying security updates for the TOE software.

ALC_TSU_EXT.1.2c

The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3c

The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Note: The reporting mechanism could include web sites, email addresses, as well as a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

ALC_TSU_EXT.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
- Privacy
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

VPNC24:FCS_CKM.1|ASPP14:FCS_CKM.1/AK|VPNC24:FCS_CKM.1/AK:

The TOE supports DH groups 19 and 20 and thus supports generation of ECC asymmetric keys against NIST curves P-256 and P-384 respectively.

ASPP14:FCS_CKM.1/SK:

The TOE uses its own OpenSSL library's DRBG to generate random values as part of IKEv2/CHILD_SA secret key generation.

VPNC24:FCS_CKM.1/VPN:

The TOE invokes Android's keychain functionality to secure manage IKE authentication parameters. Android provides the user with a secure (either Android's Keystore, a Trust-Zone backed key storage tied to the CPU, or Android's StrongBox keystore, a hardware secure element with its own dedicated storage) keychain, in which they can import certificates and private keys. Android provides methods to allow generation and import of a certificate and private key. Android provides the both the KeyPairGenerator and the Android KeyStore methods to allow secure generation of keys. In addition to generation, the TOE's UI presents an interface to Androids System UI to import a certificate (chain) and private key in p12/PFX format, or alternatively, the user can separate load the p12 file through Android's System UI (an MDM Agent or Device Policy Controller can also import p12 certificates as directed by an MDM server). When creating a new VPN profile, the TOE prompts the user to select the certificate/private key they wish the TOE to use during IKE authentication.

ASPP14:FCS_CKM.2|VPNC24:FCS_CKM.2:

The TOE uses only ECC key exchange/establishment and the TOE uses it exclusively as part of IKEv2 and ESP negotiation.

VPNC24:FCS_CKM_EXT.2:

The TOE's only persistent secrets are the IKEv2 authentication keypairs/certificates, which the TOE stores in Android's hardware backed Keystore or in Android's strongbox backed Keystore. The TOE does not store any of its other keys persistently, and instead the TOE stores non-persistent or ephemeral keys pertaining to IKEv2 and ESP SAs only in memory.

VPNC24:FCS_CKM_EXT.4:

The TOE's keys all pertain to IKEv2/IPsec and consist of the persistent IKEv2 authentication certificate/keypairs (for which the TOE relies upon the Android platform for management, including zeroization) and IKEv2 and ESP SA session keys (which the TOE stores only in working memory and clears after use).

Key	Storage	Cleared by
IKEv2 Authentication cert/key	Android KeyChain/KeyStore	User clearing keychain through Android UI
IKEv2/ESP Session keys	Working RAM	Automatically cleared when IPsec tunnel closed.

ASPP14:FCS_COP.1:

The TOE possesses the following cryptographic algorithm certificates as a part of its Secure Tactical VPN Client OpenSSL Cryptographic Library, version 3.1.0 (https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=36328):

Requirements	Functions	CAVP Cert
	Cryptographic key generation	
ASPP14:FCS_CKM_EXT.1 ASPP14:FCS_CKM.1/AK VPNC24:FCS_CKM.1 VPNC24:FCS_CKM/AK	ECC schemes using 'NIST curves' P-256, P-384	A3718
	Cryptographic key establishment/distribution	
ASPP14:FCS_CKM.2 VPNC24:FCS_CKM.2	Elliptic curve-based key establishment schemes: P-256, P-384	A3718
	IPsec/ESP Encryption/Decryption	
ASPP14:FCS_COP.1/SKC VPNC24:FCS_COP.1/SKC	AES CBC/GCM (128/256 bits)	A3718
	Cryptographic hashing	
ASPP14:FCS_COP.1/Hash	SHA-256/384/512 (digest size 256/384/512 bits)	A3718
	Keyed-hash message authentication	
ASPP14:FCS_COP.1/KeyedHash	HMAC-SHA-256/384/512 (key and output MAC size 256/384/512)	A3718
	Cryptographic signature services	
ASPP14:FCS_COP.1/Sig	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve P-256, P-384	A3718
	Random bit generation	
FCS_RBG_EXT.1	CTR_DRBG (AES-256 bit)	A3718

ASPP14:FCS_COP.1/Hash|ASPP14:FCS_COP.1/KeyedHash:

The TOE uses the SHA-256 384 and 512 algorithms, and the TOE uses them during IKEv2 authentication message signing and verification and as part of the HMAC-SHA-256/384/512 integrity for IKEv2 and ESP SAs.

ASPP14:FCS_COP.1/Sig:

The TOE supports use of ECDSA signatures during IKEv2 peer authentication using either NIST curve P-256 and P-384.

ASPP14:FCS_COP.1/SKC|VPNC24:FCS_COP.1/SKC:

The TOE only supports AES-CBC and GCM as part of IPsec. The TOE relies upon the Android platform for credential (i.e., IKE peer authentication certificates and private keys) encryption/protection.

VPNC24:FCS_IPSEC_EXT.1.1:

The TOE is a standalone Android APK and is not integrated into the OS nor is it a standalone executable bundled into the OS package. A user must install the TOE (either through the PlayStore or by obtaining the APK file and side-loading it on the Android device). The TOE provides the entirety of both IKEv2 and IPsec/ESP functionality and does not rely upon Android's Linux kernel for any cryptographic processing other than during IKE peer authentication, where the TOE relies upon Android's Keystore to securely store, manage, and utilize user credentials (certificates and private keys). The TOE also relies upon Android's documented, evaluated APIs to enforce packet routing decisions by Android's network drivers). The TOE only provides a "full-tunnel" VPN implementation, which means that the TOE instructs Android (through Android's VPN APIs) to direct all traffic through the PROTECT/encrypt network packet processing.

VPNC24:FCS_IPSEC_EXT.1.2:

The TOE provides only tunnel mode IPsec.

VPNC24:FCS_IPSEC_EXT.1.3:

The TOE always enforces a "full-tunnel VPN" and thus subjects all traffic to IPsec/ESP encryption.

VPNC24:FCS_IPSEC_EXT.1.4:

The TOE provides ESP ciphers of AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256. Additionally, the TOE provides HMAC-SHA-384 for IKEv2 and ESP SA integrity.

VPNC24:FCS_IPSEC_EXT.1.5:

The TOE implements only IKEv2 with mandatory support for NAT traversal.

VPNC24:FCS_IPSEC_EXT.1.6:

The TOE provides IKEv2 ciphers of AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256.

VPNC24:FCS_IPSEC_EXT.1.7:

The VPN Gateway can specify the IKEv2 SA lifetime when configuring a VPN connection. By default, the TOE is configured for lifetimes of 23 hours or less for IKEv2 SAs and 7 hours or less for CHILD/ESP SAs, however these values are not configurable through the TOE. If the VPN Gateway is configured to use a smaller value, those values will be used instead.

VPNC24:FCS_IPSEC_EXT.1.8:

The TOE supports DH groups 19 and 20 (ECP-256 and ECP-384, respectively), and the user can configure the TOE's VPN profiles to use either or both groups for that profiles' VPN connection.

VPNC24:FCS_IPSEC_EXT.1.9:

The TOE supports key exchange groups DH19 and DH20 and generates a secret "x" of size 256 and 384 bits, respectively using the FIPS validated RBG specified in FCS_RBG_EXT.1. 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^{128} or 2^{192}.

VPNC24:FCS_IPSEC_EXT.1.10:

The TOE generates IKEv2 nonces using its DRBG and ensures a nonce length of 128 or 192 bits.

VPNC24:FCS_IPSEC_EXT.1.11|VPNC24:FCS_IPSEC_EXT.1.12|VPNC24:FCS_IPSEC_EXT.1.13:

The TOE performs IKE peer authentication using ECDSA only and provides the user the ability to specify the "Server" (i.e., the VPN gateway) identifier. The TOE uses this value to compare against the Distinguished Name (DN) found in the peer's (VPN Gateway's) presented IKE auth certificate.

VPNC24:FCS_IPSEC_EXT.1.14:

The TOE implements RFC 4106 conformant AES-GCM-128 and AES-GCM-256, and RFC 3602 conformant AES-CBC-128 and AES-CBC-256 as encryption algorithms. The TOE implements HMAC-SHA-256, SHA-384, and SHA-512 as authentication algorithms as well as Diffie-Hellman Groups 19 and 20. The encrypted payload for IKEv2 uses AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and AES-GCM-128 and AES-GCM-256 as specified in RFC 5282. 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 Phase 1/IKE_SA are greater than or equal to that of the Phase 2/CHILD_SA.

ASPP14:FCS_RBG_EXT.1|ASPP14:FCS_RBG_EXT.2:

The TOE implements DRBG functionality (the AES-256 CTR_DRBG within its OpenSSL library) and seeds it using the ASPP14 proscribed method of calling the Android platform's /dev/random interface.

ASPP14:FCS_STO_EXT.1:

As described before, the TOE's persistent credentials include only the IKE authentication credentials (certificates and their corresponding private keys) for which the TOE relies upon the platform's secure storage using the Android Keystore and KeyChain APIs.

6.2 User data protection

ASPP14:FDP_DAR_EXT.1:

The TOE does not store any sensitive data. The application/TOE only provides the capability for the user to create and utilize VPN profiles. The VPN profiles themselves contain no data, only configuration information. Thus, design of the TOE prevents it from storing any data or sensitive data.

ASPP14:FDP_DEC_EXT.1:

The TOE makes use of network connectivity and accesses no sensitive information repositories.

ASPP14:FDP_NET_EXT.1:

The TOE allows the user to initiate IPsec VPN connections.

VPNC24:FDP_RIP.2:

The TOE has been designed to ensure that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing 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 is overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

6.3 Identification and authentication

ASPP14:FIA_X509_EXT.1|ASPP14:FIA_X509_EXT.2|VPNC24:FIA_X509_EXT.2:

The TOE uses X.509 certificates for IKEv2 authentication. The TOE requires that for each VPN connection, the user specify the client certificate for the TOE to use (the user must have previously loaded such a certificate into the keystore) and specify the CA certificate to which the Gateway/server's certificate must chain. The TOE thus uses the specified certificate when attempting to establish that VPN connection. The TOE validates authentication certificates (including the full path) and checks their revocation status using CRL (compliant with RFC 8603). The TOE processes a VPN connection to a Gateway/server by first comparing the Identification (ID) Payload received from the server against the certificate sent by the server, and if the IP address or FQDN 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). The TOE will reject any certificate for which it cannot determine the validity and reject the connection attempt.

6.4 Security management

ASPP14:FMT_CFG_EXT.1:

The TOE requires credentials (i.e., X.509 IKEv2 authentication certificates) for each configured VPN profile; however, the TOE does not install with any such credentials, and the TOE does not manage these credentials, but instead relies upon the Platform (Android Keystore) to handle all user credentials.

ASPP14:FMT_MEC_EXT.1:

The TOE stores user configured or imported VPN profiles in the Android permitted fashion (namely as files within the applications /data/data/package/ directory.

ASPP14:FMT_SMF.1:

The TOE provides no ASPP14 management functions.

VPNC24:FMT_SMF.1/VPN:

The TOE provides users the ability to configure, on a per VPN profile basis, the VPN gateway/server, the client credentials/certificate, and the expected reference identified of the VPN gateway/server.

6.5 Privacy

ASPP14:FPR_ANO_EXT.1:

The TOE does not transmit any PII over a network.

6.6 Protection of the TSF

ASPP14:FPT_AEX_EXT.1:

The TOE's native libraries (built using Android's NDK) enable ASLR and stack protection by fPIC, -DOPENSSL_PIC, and the –fstack-protector-all flags. Furthermore, the application does not allocate any memory region with execute permissions (and thus prevents allocation of any memory region with both write and execute permissions).

ASPP14:FPT_API_EXT.1:

The TOE uses the following Platform APIs:

android.annotation.TargetApi

android.app.AlarmManager

android.app.Application

android.app.Dialog

android.app.Notification

android.app.NotificationChannel

android.app.NotificationManager

android.app.PendingIntent

android.app.Service

android.content.ActivityNotFoundException

android.content.BroadcastReceiver

android.content.ComponentName

android.content.ContentProvider

android.content.ContentResolver

android.content.ContentValues

android.content.Context

android.content.DialogInterface

android.content.Intent

android.content.IntentFilter

android.content.pm.ApplicationInfo

android.content.pm.PackageInfo

android.content.pm.PackageManager

android.content.pm.PackageManager.NameNotFoundException

android.content.res.TypedArray

android.content.ServiceConnection

android.content.SharedPreferences

android.database.Cursor

android.database.MatrixCursor

android.database.SQLException

android.database.sqlite.SQLiteDatabase

android.database.sqlite.SQLiteOpenHelper

android.database.sqlite.SQLiteQueryBuilder

android.graphics.drawable.Drawable

android.graphics.drawable.Icon

android.Manifest

android.net.ConnectivityManager

android.net.http.SslCertificate

android.net.Network

android.net.NetworkRequest

android.net.Uri

android.net.VpnService

android.os.Binder

android.os.Build

android.os.Bundle

android.os.FileObserver

android.os.Handler

android.os.IBinder

android.os.Looper

android.os.Message

android.os.Parcel

android.os.Parcelable

android.os.ParcelFileDescriptor

android.os.PowerManager

android.os.SystemClock

android.provider.OpenableColumns

android.provider.Settings

android.security.KeyChain

android.security.KeyChainAliasCallback

android.security.KeyChainException

android.service.quicksettings.Tile

android.service.quicksettings.TileService

android.system.OsConstants

android.text.Editable

android.text.format.Formatter

android.text.method.LinkMovementMethod

android.text.SpannableString

android.text.Spanned

android.text.TextUtils

android.text.TextWatcher

android.util.AttributeSet

android.util.Base64

android.util.Log

android.util.Pair

android.util.TypedValue

android.util.Xml

android.view.ActionMode

android.view.GestureDetector

android.view.LayoutInflater

android.view.Menu

android.view.MenuInflater

android.view.MenuItem

android.view.MotionEvent

android.view.View

android.view.ViewConfiguration

android.view.ViewGroup

android.view.View.OnClickListener

android.view.View.OnTouchListener

android.widget.AbsListView.MultiChoiceModeListener

android.widget.AdapterView

android.widget.AdapterView.OnItemClickListener

android.widget.AdapterView.OnItemSelectedListener

android.widget.ArrayAdapter

android.widget.BaseAdapter

android.widget.Button

android.widget.Checkable

android.widget.CheckBox

android.widget.CompoundButton

android.widget.CompoundButton.OnCheckedChangeListener

android.widget.EditText

android.widget.Filter

android.widget.Filterable

android.widget.FrameLayout

android.widget.ImageView

android.widget.LinearLayout

android.widget.ListView

android.widget.MultiAutoCompleteTextView

android.widget.ProgressBar

android.widget.RelativeLayout

android.widget.SearchView

android.widget.SearchView.OnQueryTextListener

android.widget.Spinner

android.widget.TextView

android.widget.Toast

androidx.activity.result.ActivityResultLauncher

androidx.activity.result.contract.ActivityResultContracts

androidx.annotation.Keep

androidx.annotation.NonNull

androidx.annotation.Nullable

androidx.annotation.RequiresApi

androidx.appcompat.app.ActionBar

androidx.appcompat.app.AlertDialog

androidx.appcompat.app.AppCompatActivity

androidx.appcompat.app.AppCompatDialogFragment

androidx.appcompat.widget.SearchView

androidx.appcompat.widget.SwitchCompat

androidx.core.app.NotificationCompat

androidx.core.content.ContextCompat

androidx.core.content.pm.ShortcutInfoCompat

androidx.core.content.pm.ShortcutManagerCompat

androidx.core.graphics.drawable.IconCompat

androidx.core.os.HandlerCompat

androidx.core.text.HtmlCompat

androidx.core.view.ViewCompat

androidx.core.view.ViewPropertyAnimatorListenerAdapter

androidx.fragment.app.Fragment

androidx.fragment.app.FragmentActivity

androidx.fragment.app.FragmentManager

androidx.fragment.app.FragmentTransaction

androidx.fragment.app.ListFragment

androidx.loader.app.LoaderManager

androidx.loader.app.LoaderManager.LoaderCallbacks

androidx.loader.content.AsyncTaskLoader

androidx.loader.content.Loader

androidx.localbroadcastmanager.content.LocalBroadcastManager androidx.preference.ListPreference

androidx.preference.Preference

androidx.preference.PreferenceFragmentCompat

androidx.preference.PreferenceManager

androidx.viewpager2.adapter.FragmentStateAdapter

androidx.viewpager2.widget.ViewPager2

com.google.android.material.tabs.TabLayout

com.google.android.material.tabs.TabLayoutMediator

com.google.android.material.textfield.TextInputLayout

java.io.BufferedOutputStream

java.io.BufferedReader

java.io.ByteArrayInputStream

java.io.ByteArrayOutputStream

java.io.File

java.io.FileInputStream

java.io.FileNotFoundException

java.io.FileOutputStream

java.io.FileReader

java.io.InputStream

java.io.InputStreamReader

java.io.IOException

java.io.OutputStream

java.io.StringReader

java.lang.ref.WeakReference

java.net.HttpURLConnection

java.net.Inet4Address

java.net.Inet6Address

java.net.InetAddress

java.net.SocketTimeoutException

java.net.UnknownHostException

java.net.URL

java.nio.ByteBuffer

java.nio.channels.ClosedByInterruptException

java.security.cert.Certificate

java.security.cert.CertificateEncodingException

java.security.cert.CertificateException

java.security.cert.CertificateFactory

java.security.cert.CertificateParsingException

java.security.cert.X509Certificate

java.security.Key

java.security.KeyStore

java.security.KeyStoreException

java.security.KeyStoreSpi

java.security.MessageDigest

java.security.NoSuchAlgorithmException

java.security.PrivateKey

java.security.Provider

java.security.SecureRandom

java.security.Security

java.security.UnrecoverableKeyException

java.text.Collator

java.util.ArrayList

java.util.Arrays

java.util.Collection

java.util.Collections

java.util.Comparator

java.util.concurrent.Callable

java.util.concurrent.CancellationException

java.util.concurrent.ConcurrentHashMap

java.util.concurrent.ExecutionException

java.util.concurrent.Executor

java.util.concurrent.Executors

java.util.concurrent.ExecutorService

java.util.concurrent.Future

java.util.concurrent.locks.ReentrantReadWriteLock

java.util.concurrent.TimeoutException

java.util.concurrent.TimeUnit

java.util.Date

java.util.Enumeration

java.util.EnumSet

java.util.HashSet

java.util.Hashtable

java.util.Iterator

java.util.LinkedHashMap

java.util.LinkedList

java.util.List

java.util.Locale

java.util.Map.Entry

java.util.Observable

java.util.Observer

java.util.PriorityQueue

java.util.regex.Matcher

java.util.regex.Pattern

java.util.SortedSet

java.util.TreeSet

java.util.UUID

javax.net.ssl.SSLHandshakeException

org.json.JSONArray

org.json.JSONException

org.json.JSONObject

org.xmlpull.v1.XmlPullParser

org.xmlpull.v1.XmlPullParserException

ASPP14:FPT_IDV_EXT.1:

The TOE uses a major, minor, and build number.

ASPP14:FPT_LIB_EXT.1:

The TOE statically links the 3rd party OpenSSL library.

VPNC24:FPT_TST_EXT.1/VPN:

The TOE performs a series of self-tests upon loading/execution. These include cryptographic algorithm self-tests for the algorithms within its OpenSSL library. For each cryptographic algorithm test, the TOE uses a known key, plaintext (or message) and a known result. The TOE uses the algorithm on the known inputs and compares the result to the known/expected output. If the output differs, the TOE fails the test and exits. These tests cover the following Cryptographic Algorithm Tests:

- AES-CBC, AES-GCM Known Answer Tests
- HMAC-SHA-256 Known Answer Test
- HMAC-SHA-384 Known Answer Test
- HMAC-SHA-512 Known Answer Test
- SHA-256 Known Answer Test
- SHA-384 Known Answer Test
- SHA-512 Known Answer Test
- AES-CTR DRBG Known Answer Test
- ECDSA Known Answer Test

The TOE relies upon the Platform for storage, integrity, and verification of the TOE's executable code.

ASPP14:FPT_TUD_EXT.1:

DataSoft makes VPN Client updates through the APK package format and distributes updated APKs both directly to customers as well as through the Google PlayStore.

ASPP14:FPT_TUD_EXT.2:

DataSoft signs their VPN Client APK with their unique developer private key to ensure authenticity of updates. That DataSoft unique developer private key corresponds to DataSoft's developer public key registered with Google's PlayStore.

ASPP14:ALC_TSU_EXT.1:

The vendor provides timely security updates for the TOE in case vulnerabilities have been discovered. Reported vulnerabilities and defects are investigated and rated based on the threat and result of the impact analysis and then scheduled for an upcoming bug fix release based on the severity. The vendor aims for security updates as soon as possible with a maximum of 30 days. Third party library updates (OpenSSL) are also included as a part of the TOE’s update. The vendor actively monitors both internal and third-party components and accepts vulnerability reports through the DataSoft email support address (support@DataSoft.com).

6.7 Trusted path/channels

ASPP14:FTP_DIT_EXT.1|VPNC24:FTP_DIT_EXT.1:

The TOE uses IPsec to encrypt VPN traffic exchanged with the Gateway/server.