Avast Virtual Mobile Platform Client (ASPP12) Security Target

Version 0.5
May 30, 2017

Prepared for:

Avast Software, Inc.

2625 Broadway Street
Redwood City, CA 94063-1532

Prepared By:

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 Virtual Mobile Platform Client provided by Avast Software, Inc. 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 – Avast Virtual Mobile Platform Client (ASPP12) Security Target

ST Version – Version 0.5

ST Date – May 30, 2017

1.2 TOE Reference

TOE Identification – Avast Software, Inc. Virtual Mobile Platform Client, Version 3.4

TOE Developer – Avast Software, Inc.

Evaluation Sponsor – Avast Software, Inc.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Avast Virtual Mobile Platform (VMP) Client Version 3.4.

The TOE is the Virtual Mobile Platform Client application for Android and iOS platforms. The TOE is a thin client providing access to an Avast Virtual Mobile Infrastructure (VMI) server from a mobile device. The TOE runs on evaluated Samsung Galaxy S7, S7 Edge, S6, S6 Edge, Note 4, Note 5, Note Edge and Tab S2 devices running Android 6.0.1. The TOE also runs on evaluated Apple iOS 9.3.2 on iPhone and iPad devices using the A7 or A8 processor.

1.4 TOE Description

A VMP client streams standard mobile apps to any device-- hosting the data on a secure server. Mobile applications run on the secure server and no data is transferred from the server and stored on the physical mobile device. The VMP client presents only the interface offered by the VMP server and ensures that communication with the server utilizes secured protocols.

The TOE when executed, connects to the specified Avast Virtual Mobile Infrastructure (VMI) server, authenticating the server's certificate received while negotiating the HTTPS or TLS session. The TOE is responsible only for protecting data-in-transit between the physical mobile device and the VMP server.

1.4.1 TOE Architecture

The TOE is an application installed onto a physical mobile device from the Google Playstore or Apple App Store.

1.4.1.1 Physical Boundaries

The physical boundary of the TOE is the physical perimeter of the evaluated device on which the TOE resides.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by VMP Client:

- 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 VMP client utilizes platform APIs to provide secure network communication using the HTTPS and TLS protocols.

1.4.1.2.2 User data protection

The VMP client does not store sensitive data in local files. The VMP client can access most physical resources on the mobile device, but none of the logical data repositories.

1.4.1.2.3 Identification and authentication

The VMP client utilizes platform provided functionality to verify certificates authenticating network endpoints. The iOS platform support OCSP while the Android platform supports both OSCP and CRL.

1.4.1.2.4 Security management

The VMP client does not include any predefined or default credentials, and utilize the platform recommended storage process for configuration options.

1.4.1.2.5 Privacy

The VMP client does not collect any PII and does not transmit any PII over a network.

1.4.1.2.6 Protection of the TSF

The VMP client relies on the physical boundary of the evaluated platform as well as the Android and iOS operating system for the protection of the TOE's application components. The VMP client also makes use of specific 3rd party libraries to support WebRTC. All compiled VMP client code is designed to utilize compiler provided anti-exploitation capabilities. The VMP client application is available through the Google Playstore and the Apple store.

1.4.1.2.7 Trusted path/channels

The VMP client utilizes platform API to establish HTTPS and TLS connections to a VMP server.

1.4.2 TOE Documentation

Avast offers documents that describe the Administration, operation and maintenance for the TOE. This documentation is currently under development.

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 Extended
- Package Claims:
    - Protection Profile for Application Software, Version 1.2, 22 April 2016 (ASPP12)

2.1 Conformance Rationale

The ST conforms to the ASPP12. 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 ASPP12 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The ASPP12 offers additional information about the identified security objectives, but that has not been reproduced here and the ASPP12 should be consulted if there is interest in that material.

In general, the ASPP12 has defined Security Objectives appropriate for a software application and as such are applicable to the Virtual Mobile Platform Client TOE.

3.1 Security Objectives for the Operational 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.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the ASPP12. The ASPP12 defines the following extended requirements and since they are not redefined in this ST the ASPP12 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_RBG_EXT.1: Random Bit Generation Services
- FCS_STO_EXT.1: Storage of Credentials
- FCS_TLSC_EXT.1: TLS Client Protocol
- FCS_TLSC_EXT.4: TLS Client Protocol
- FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
- FDP_DEC_EXT.1: Access to Platform Resources
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X.509 Certificate Authentication
- FMT_CFG_EXT.1: Secure by Default Configuration
- FMT_MEC_EXT.1: Supported Configuration Mechanism
- FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
- FPT_AEX_EXT.1: Anti-Exploitation Capabilities
- FPT_API_EXT.1: Use of Supported Services and APIs
- FPT_LIB_EXT.1: Use of Third Party Libraries
- FPT_TUD_EXT.1: Integrity for Installation and Update
- 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 ASPP12. The refinements and operations already performed in the ASPP12 are not identified (e.g., highlighted) here, rather the requirements have been copied from the ASPP12 and any residual operations have been completed herein. Of particular note, the ASPP12 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 ASPP12 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the ASPP12 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The ASPP12 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Virtual Mobile Platform Client TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation Services
	FCS_STO_EXT.1: Storage of Credentials
	FCS_TLSC_EXT.1: TLS Client Protocol
	FCS_TLSC_EXT.4: TLS Client Protocol
FDP: User data protection	FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
	FDP_DEC_EXT.1: Access to Platform Resources
FIA: Identification and authentication	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security management	FMT_CFG_EXT.1: Secure by Default Configuration
	FMT_MEC_EXT.1: Supported Configuration Mechanism
	FMT_SMF.1: Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
FPT: Protection of the TSF	FPT_AEX_EXT.1: Anti-Exploitation Capabilities
	FPT_API_EXT.1: Use of Supported Services and APIs
	FPT_LIB_EXT.1: Use of Third Party Libraries
	FPT_TUD_EXT.1: Integrity for Installation and Update
FTP: Trusted path/channels	FTP_DIT_EXT.1: Protection of Data in Transit

Table 5-1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1
The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2
The application shall implement HTTPS using TLS (FCS_TLSC_EXT.1).

FCS_HTTPS_EXT.1.3
The application shall notify the user and [***not establish the connection***] if the peer certificate is deemed invalid.

5.1.1.2 Random Bit Generation Services (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1
The application shall [***invoke platform-provided DRBG functionality***] for its cryptographic operations.

5.1.1.3 Storage of Credentials (FCS_STO_EXT.1)

FCS_STO_EXT.1.1
The application shall [***invoke the functionality provided by the platform to securely store [certificate of VMP server]***] to non-volatile memory.

5.1.1.4 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1
The application shall [***invoke platform-provided TLS 1.2***] supporting the following cipher suites:

Mandatory Cipher Suites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246

Optional Cipher Suites: [

- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

].

FCS_TLSC_EXT.1.2
The application shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3
The application shall establish a trusted channel only if the peer certificate is valid.

5.1.1.5 TLS Client Protocol (FCS_TLSC_EXT.4)

FCS_TLSC_EXT.4.1
The application shall present the supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1, secp521r1***] and no other curves.

5.1.2 User data protection (FDP)

5.1.2.1 Encryption Of Sensitive Application Data (FDP_DAR_EXT.1)

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 (FDP_DEC_EXT.1)

FDP_DEC_EXT.1.1
The application shall restrict its access to [***location services, camera, phone, storage, microphone, bluetooth, network access, audio settings, vibration, notifications, and cellular data***].

FDP_DEC_EXT.1.2
The application shall restrict its access to [***no sensitive information repositories***].

5.1.2.3 Network Communications (FDP_NET_EXT.1)

FDP_NET_EXT.1.1
The application shall restrict network communication to [***user-initiated communication for [connecting to a VMP server]***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1
The application shall [***invoked platform-provided 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 and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The application shall validate the extendedKeyUsage field according to the following rules:
    - 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.
    - 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 extendedKeyUsage field.
    - 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 extendedKeyUsage field.
    - 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 extendedKeyUsage field.
    - 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 extendedKeyUsage field.
    - 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 extendedKeyUsage field.

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 (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

NOTE: The behavior differs between the iOS and Android platforms, thus both selections are being made here.

5.1.4 Security management (FMT)

5.1.4.1 Secure by Default Configuration (FMT_CFG_EXT.1)

FMT_CFG_EXT.1.1

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

FMT_CFG_EXT.1.2

The application shall be configured by default with file permissions which protect it and its data from unauthorized access.

5.1.4.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

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 (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [***[***

- ***Enable CC mode***
- ***Specify the nickname and network address of a VMP server***

]].

5.1.5 Privacy (FPR)

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

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 (FPT_AEX_EXT.1)

FPT_AEX_EXT.1.1

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

FPT_AEX_EXT.1.2

The application shall [***allocate memory regions with write and execute permissions for only [v8: used by Crosswalk Project (libxwalkcore.so) to perform just-in-time compilation of Javascript code]***].

FPT_AEX_EXT.1.3

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

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.

FPT_AEX_EXT.1.5

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

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

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

5.1.6.3 Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1

The application shall be packaged with only [

- **crosswalk-chromium**
- **libwebrtc**
- **openh264**

].

Application Note: The crosswalk-chromium is a library used only on Android versions of the TOE. The libwebrtc and openh264 are used only on iOS.

5.1.6.4 Integrity for Installation and Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3

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.

FPT_TUD_EXT.1.4

The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.1.6

The application installation package and its updates 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 (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1

The application shall [***encrypt all transmitted data with [HTTPS, TLS]***] between itself and another trusted IT product.

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 5-2 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 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.

5.2.3.3 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. 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. 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

The TOE utilizes the AES-256 CTR_DRBG provided by the Samsung Galaxy S7, S7 Edge, S6, S6 Edge, Note 4, Note 5, Note Edge and Tab S2 on Android 6.0.1 platform. On the iOS platform, the TOE uses the AES-256 CTR_DRBG provided by iOS 9.3.2.

The TOE utilizes platform provided TLS functionality for both Android and iOS platforms, relying upon the platform for the TLS and HTTPS protocol functionality. When configured in CC mode, the TOE allows the use of TLS version 1.2 only (regardless of whether it is using HTTPS or TLS for real-time communications).

The TOE uses the following ciphersuites (that are provided by the platform) when acting as a TLS client on any iOS platform.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384The TOE use the same BoringSSL APIs during its use of TLS, regardless of the Android operating platform. The TOE uses the following ciphersuites (that are provided by the platform) when acting as a TLS client on any Android platform.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The Avast VMP client makes use of a platform entropy source only to create a key which is used to encrypt VMP client sensitive configuration data. The VMP client does not include its own DRBG. Also, the VMP client utilizes platform security protocols, and thus does not use random values in support of TLS. On Android platforms, the TOE obtains random data from /dev/random. On iOS platforms, the VMP client utilizes the Apple iOS cryptographic library to obtain random data that is used as a key to protect sensitive configuration data.

Because the TOE utilizes platform protocols for communications, the reference identifies and certificate pinning in effect are those supported by the platform. The Android and iOS platforms do not use certificate pinning. Both platforms use reference identifiers of Common Name (CN) or Subject Alternate Name (SAN) (i.e., DNS or IP Address).

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

- FCS_HTTPS_EXT.1: The TOE supports HTTPS using platform provided APIs.
- FCS_RBG_EXT.1: The product obtains a random Device ID once for each VMP server to which the VMP client connects. This value as well as all other random data used by the VMP client, is obtained from an approved DRBG provided by the platform.
- FCS_STO_EXT.1: The certificate used to authenticate the VMP server is protected using the Android platform's KeyStore or the iOS keychain (depending upon the platform).
- FCS_TLSC_EXT.1: The TOE supports only TLS version 1.2 on both Android and iOS platforms, using the ciphersuites listed above. The TOE allows a connection only when a certificate is deemed to be valid. The TOE supports all of the ciphersuites required by FCS_TLSC_EXT.1 on an Android platform and supports the mandatory ciphersuite on an iOS platform.
- FCS_TLSC_EXT.4: By default, the TOE allows ECDHE ciphersuites on the Android platform only. The TOE relies upon the Android platform to support only the NIST curves secp256r1, secp384r1 and secp521r1. The iOS platform does not support any ECDHE ciphersuites and thus support for NIST curves is not available.

6.2 User data protection

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_DAR_EXT.1: The TOE does not store sensitive data in local files on either iOS or Android platforms.
- FDP_DEC_EXT.1: The TOE can access the physical resources on the mobile device. For an Android device, the TOE can access Location services, Camera, Phone, Storage, Microphone, Bluetooth, Network Access, Audio Settings, and Vibration. For an iOS device, the TOE can access location services, camera, photos, microphone, notifications, and cellular data. However, the TOE cannot access any of the logical data repositories.
- FDP_NET_EXT.1: The TOE allows network communication to be initiated by a user in order to connect to a VMP server.

The VMP client does not transmit any PII from a mobile device to the VMP server. Network traffic is limited to traffic associated with the establishment of the user initiated connection to a VMP server.

6.3 Identification and authentication

The TOE uses X.509v3 certificates to authenticate network endpoints for TLS and HTTPS communication. The X.509v3 certificates are verified by the TOE through the use of APIs provided by the Android and iOS platforms. The TOE uses the Android and iOS platforms to verify fields within certificates related to usage, validity periods, revocation status and all other flags of a certificate.

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

- FIA_X509_EXT.1: The TOE utilizes platform provided functionality to verify certificates authenticating network endpoints. The iOS platform support OCSP while the Android platform supports both OSCP and CRL. The TOE relies upon the platforms to verify the validity of certificates, their certificate status and their certificate path.

- FIA_X509_EXT.2: Because the TOE relies upon the platform for network communication, it also relies upon the platform for validation of X.509v3 certificates as well as for checking the revocation status of the certificate. The iOS platform support OCSP while always accepting certificates as valid when revocation status cannot be determined. The Android platform support both OSCP and CRL, and does not accept certificates as valid when revocation status cannot be determined.

6.4 Security management

The following are management operation available to VMP client users:

- Enable CC mode
- Specify the nickname and network address of a VMP server

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

- FMT_CFG_EXT.1: The VMP client does not include any predefined or default credentials.
- FMT_MEC_EXT.1: The evaluated Android platform on which the TOE executes automatically uses /data/data/package/shared_prefs/ to store configuration options and settings. For an iOS platform, all settings are stored in the iOS the UserDefaults system.
- FMT_SMF.1: The management functions offered by the TOE are listed above.

6.5 Privacy

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

- FPR_ANO_EXT.1: The VMP client does not collect any PII and does not intentionally transmit any PII over a network. Users may choose to transmit any data over an established connection to the VMP server, but it is not specifically identifiable as PII.

6.6 Protection of the TSF

The TOE is physically protected by the boundary of the evaluated device. The TOE is executed on an evaluated Android 6.0.1 or iOS 9.3.2 device. The TOE is constructed as a Java script application that utilizes APIs provided by either an iOS or Android platform for cryptography and secure network communications.

Memory mapping and permissions on memory regions are not functions applicable to a Java script application. However, some 3rd party libraries are written in a language other than Java and thus are subject to the requirement for Andi-Exploitation Capabilities. However, none of the 3rd party libraries used by the TOE request memory mapping at explicit addresses, and none allocate memory for both write and execute permission.

Android's application management requires application updates to be signed with an Android key, thus allowing the secure updates of its applications. The Android OS Linux kernel is capable of ASLR (address space layout randomization), ensuring that no application uses the same address layout on two different devices.

3rd Party Libraries
crosswalk-chromium
libwebrtc
openh264

Table 6-1 3rd Party Libraries

The crosswalk-chromium is a library used only on Android versions of the TOE. The libwebrtc and openh264 are used only on iOS.

Avast allows users to submit bug reports and vulnerabilities through its support site at http://remotium.freshdesk.com. Each reported issue is assigned a tracking number, and Avast personnel monitor the reported issues and feed that into our internal bug tracking database. As Avast processes the reported issues, an issue is marked as closed or resolved, which causes the user to be sent an email notification advising of the change in status. Avast also leverages existing app store mechanisms (Google Play, Apple App Store) for distributing app updates to users.

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

- FPT_AEX_EXT.1: The TOE libraries are also compiled with the '-fstack-protector-all –fno-exceptions' flags in order to enable ASLR and stack-based buffer over flow protections. On iOS the ASLR feature (-pie) is not set by a compiler flag, because it is on by default on the C-language compiler and this setting is required by the Apple App store. The TOE[1] does allocate memory regions with both write and execute permissions for the following functions only, which are used in support of just-in-time (JIT) compilation:

  - v8: used by Crosswalk Project (libxwalkcore.so) to perform JIT compilation of Javascript code.

  The TOE produces such pieces of executable code in runtime and are available in-memory only for the running instance of the application. No piece of user-initiated JIT executable code is ever stored on disk. Also, after the Javascript engine has finished producing this JIT code it is turned into read-only executable memory, limiting the exposure of write-and-execute memory areas. V8 is a state-of-the-art Javascript compiler that counts with modern techniques of anti-exploitation specifically devised to thwart attacks on JIT engines, such as: Constant blinding, ASLR, and memory allocation limitations.

- FPT_API_EXT.1: The TOE uses the platform provided APIs for all cryptographic operations. Refer to the [Admin] documentation for a full list of APIs used by the TOE.

- FPT_LIB_EXT.1: The TOE only uses the third party libraries listed in the Table 6-1 3rd Party Libraries above and explained in the note following the table.

- FPT_TUD_EXT.1: The TOE (VMP client) application is available through the Google Playstore and the Apple store. Thus, the platform will be providing all required capabilities for trusted updates. Bug reporting is available to users as described above.

- ALC_TSU_EXT.1: Avast accepts bug reports (including reports for security vulnerabilities) through a Technical Support Contact form on the remotium.freshdesk.com.web site. Avast reviews all bug reports when making product changes to resolve issues associated with the TOE. Avast makes updates and code patches to resolve issues as quickly as possible, and makes updates available to customers. TOE updates are distributed through the Apple App Store and Google Play. For maximum compatibility, Avast recommends customers use the iOS and Android update mechanisms to keep Avast Workspace up-to-date.

6.7 Trusted path/channels

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_DIT_EXT.1: The TOE utilizes platform API to establish HTTPS and TLS connections to a VMP server.

[1] The Android Version of the TOE.

7. Security Related Platform APIs Invoked by TOE

This section identifies the Platform APIs that are invoked by the TOE which utilize security functions provided by the platform.

7.1 Android C APIs (OpenSSL)

BIO_ctrl_get_read_request

BIO_ctrl_get_write_guarantee

BIO_free_all

BIO_get_callback_arg

BIO_new_bio_pair

BIO_new_mem_buf

BIO_pending

BIO_read

BIO_set_callback

BIO_set_callback_arg

BIO_shutdown_wr

BIO_wpending

BIO_write

BIO_zero_copy_get_read_buf_done

BIO_zero_copy_get_write_buf_done

SSL_certs_clear

SSL_CIPHER_get_bits

SSL_CIPHER_get_id

SSL_clear_mode

SSL_clear_options

SSL_CTX_add_client_custom_ext

SSL_CTX_new

SSL_CTX_sess_set_new_cb

SSL_CTX_set_cert_cb

SSL_CTX_set_cert_verify_callback

SSL_CTX_set_keylog_callback

SSL_CTX_set_next_proto_select_cb

SSL_CTX_set_quiet_shutdown

SSL_CTX_set_verify

SSL_do_handshake

SSL_enable_ocsp_stapling

SSL_enable_signed_cert_timestamps

SSL_enable_tls_channel_id

SSL_export_keying_material

SSL_free

SSL_get_ciphers

SSL_get_client_CA_list

SSL_get_current_cipher

SSL_get_curve_name

SSL_get_error

SSL_get_ex_data

SSL_get_ex_data_X509_STORE_CTX_idx

SSL_get_ex_new_index

SSL_get_extms_support

SSL_get_peer_cert_chain

SSL_get_peer_certificate

SSL_get_secure_renegotiation_support

SSL_get_server_key_exchange_hash

SSL_get_session

SSL_get_state

SSL_get_wbio

SSL_get0_alpn_selected

SSL_get0_certificate_types

SSL_get0_ocsp_response

SSL_get0_signed_cert_timestamp_list

SSL_load_client_CA_file

SSL_new

SSL_read

SSL_SESSION_free

SSL_session_reused

SSL_SESSION_up_ref

SSL_set_accept_state

SSL_set_alpn_protos

SSL_set_bio

SSL_set_cipher_list

SSL_set_client_CA_list

SSL_set_connect_state

SSL_set_ex_data

SSL_set_max_version

SSL_set_min_version

SSL_set_mode

SSL_set_options

SSL_set_private_key_digest_prefs

SSL_set_private_key_method

SSL_set_renegotiate_mode

SSL_set_session

SSL_set_tlsext_host_name

SSL_set1_chain

SSL_set1_tls_channel_id

SSL_shutdown

SSL_use_certificate

SSL_use_PrivateKey

SSL_version

SSL_write

SSLv23_client_method

TLS_method

7.2 Android Java APIs

android.net.http.X509TrustManagerExtensions

android.security.KeyChain

java.security.KeyStore

java.security.KeyStoreException

java.security.MessageDigest

java.security.NoSuchAlgorithmException

java.security.PublicKey

java.security.cert.Certificate

java.security.cert.CertificateException

java.security.cert.CertificateExpiredException

java.security.cert.CertificateFactory

java.security.cert.CertificateNotYetValidException java.security.cert.X509Certificate

javax.net.ssl.TrustManager

javax.net.ssl.TrustManagerFactory

javax.net.ssl.X509TrustManager

javax.security.auth.x500.X500Principal

7.3 iOS Obj-C APIs

NSURLProtocol

NSURLRequest

NSURLConnection

[NSURLConnection willSendRequestForAuthenticationChallenge]

[NSURLConnection willSendRequest]

[NSURLConnection didFailWithError]