

# VMware Workspace ONE Unified Endpoint Management Version 2209

---

## Security Target

ST Version: 1.0

February 6, 2023

**VMware, Inc.**

3401 Hillview Ave.

Palo Alto, CA 94304

Prepared By:

**Booz | Allen | Hamilton**

delivering results that endure

Cyber Assurance Testing Laboratory

1100 West Street

Laurel, MD 20707

## Table of Contents

|       |   |    |
|-------|---|----|
| 1     | Security Target Introduction .....                              | 7  |
| 1.1   | ST Reference.....   | 7  |
| 1.1.1 | ST Identification .....   | 7  |
| 1.1.2 | Document Organization .....                                     | 7  |
| 1.1.3 | Terminology.....  | 7  |
| 1.1.4 | Acronyms.....   | 8  |
| 1.1.5 | Reference .....   | 9  |
| 1.2   | TOE Reference.....  | 10 |
| 1.3   | TOE Overview .....  | 10 |
| 1.4   | TOE Type.....   | 12 |
| 2     | TOE Description .....   | 13 |
| 2.1   | Evaluated Components of the TOE .....                           | 13 |
| 2.2   | Components and Applications in the Operational Environment..... | 13 |
| 2.3   | Excluded from the TOE .....                                     | 14 |
| 2.3.1 | Not Installed.....  | 14 |
| 2.3.2 | Installed but Requires a Separate License.....                  | 14 |
| 2.3.3 | Installed But Not Part of the TSF.....                          | 14 |
| 2.4   | Physical Boundary .....   | 15 |
| 2.4.1 | Hardware.....   | 15 |
| 2.4.2 | Software .....  | 15 |
| 2.5   | Logical Boundary.....   | 15 |
| 2.5.1 | Security Audit .....  | 15 |
| 2.5.2 | Communication.....  | 16 |
| 2.5.3 | Cryptographic Support.....                                      | 16 |
| 2.5.4 | Identification and Authentication.....                          | 17 |
| 2.5.5 | Security Management .....                                       | 17 |
| 2.5.6 | Protection of the TSF .....                                     | 17 |
| 2.5.7 | TOE Access .....  | 18 |
| 2.5.8 | Trusted Path/Channels .....                                     | 18 |
| 3     | Conformance Claims .....  | 19 |

- 3.1 CC Version.....19
- 3.2 CC Part 2 Conformance Claims.....19
- 3.3 CC Part 3 Conformance Claims.....19
- 3.4 PP Claims.....19
- 3.5 Package Claims.....19
- 3.6 Package Name Conformant or Package Name Augmented.....20
- 3.7 Conformance Claim Rationale.....20
- 3.8 Technical Decisions .....21
- 4 Security Problem Definition .....23
  - 4.1 Threats.....23
  - 4.2 Organizational Security Policies .....23
  - 4.3 Assumptions.....23
  - 4.4 Security Objectives .....24
    - 4.4.1 TOE Security Objectives .....24
    - 4.4.2 Security Objectives for the Operational Environment .....25
  - 4.5 Security Problem Definition Rationale .....25
- 5 Extended Components Definition.....26
  - 5.1 Extended Security Functional Requirements .....26
  - 5.2 Extended Security Assurance Requirements .....26
- 6 Security Functional Requirements .....27
  - 6.1 Conventions .....27
  - 6.2 Security Functional Requirements Summary.....27
  - 6.3 Security Functional Requirements .....29
    - 6.3.1 Class FAU: Security Audit .....29
    - 6.3.2 Class FCO: Communication .....35
    - 6.3.3 Class FCS: Cryptographic Support .....35
    - 6.3.4 Class FIA: Identification and Authentication .....37
    - 6.3.5 Class FMT: Security Management .....40
    - 6.3.6 Class FPT: Protection of the TSF .....47
    - 6.3.7 Class FTA: TOE Access .....48
    - 6.3.8 Class FTP: Trusted Path/Channels.....48

- 6.4 Statement of Security Functional Requirements Consistency .....50
- 7 Security Assurance Requirements .....51
  - 7.1 Class ASE: Security Target evaluation.....51
    - 7.1.1 ST introduction (ASE\_INT.1).....51
    - 7.1.2 Conformance claims (ASE\_CCL.1).....52
    - 7.1.3 Security objectives for the operational environment (ASE\_OBJ.1) .....53
    - 7.1.4 Extended components definition (ASE\_ECD.1).....53
    - 7.1.5 Stated security requirements (ASE\_REQ.1) .....54
    - 7.1.6 TOE summary specification (ASE\_TSS.1).....55
  - 7.2 Class ADV: Development.....56
    - 7.2.1 Basic Functional Specification (ADV\_FSP.1).....56
  - 7.3 Class AGD: Guidance Documentation .....57
    - 7.3.1 Operational User Guidance (AGD\_OPE.1) .....57
    - 7.3.2 Preparative Procedures (AGD\_PRE.1) .....58
  - 7.4 Class ALC: Life Cycle Support .....58
    - 7.4.1 Labeling of the TOE (ALC\_CMC.1).....58
    - 7.4.2 TOE CM Coverage (ALC\_CMS.1) .....59
  - 7.5 Class ATE: Tests.....59
    - 7.5.1 Independent Testing - Conformance (ATE\_IND.1) .....59
  - 7.6 Class AVA: Vulnerability Assessment .....60
    - 7.6.1 Vulnerability Survey (AVA\_VAN.1) .....60
- 8 TOE Summary Specification .....61
  - 8.1 Security Audit .....62
    - 8.1.1 [MDMPP] FAU\_ALT\_EXT.1 .....62
    - 8.1.2 [AGENTMOD] FAU\_ALT\_EXT.2/ANDROID .....63
    - 8.1.3 [AGENTMOD] FAU\_ALT\_EXT.2/IOS .....64
    - 8.1.4 [MDMPP] FAU\_GEN.1(1).....65
    - 8.1.5 [MDMPP] FAU\_GEN.1(2).....68
    - 8.1.6 [AGENTMOD] FAU\_GEN.1(2) .....68
    - 8.1.7 [MDMPP] FAU\_NET\_EXT.1 .....70
    - 8.1.8 [MDMPP] FAU\_SAR.1 .....70

8.1.9 [AGENTMOD] FAU\_SEL.1(2) .....70

8.1.10 [MDMPP] FAU\_STG\_EXT.1 .....71

8.2 Communication.....71

8.2.1 [MDMPP] FCO\_CPC\_EXT.1 .....71

8.3 Cryptographic Support.....72

8.3.1 [MDMPP] FCS\_CKM.1 .....72

8.3.2 [MDMPP] FCS\_CKM.2 .....73

8.3.3 [MDMPP] FCS\_CKM\_EXT.4.....73

8.3.4 [MDMPP] FCS\_COP.1(1).....73

8.3.5 [MDMPP] FCS\_COP.1(2).....74

8.3.6 [MDMPP] FCS\_COP.1(3).....74

8.3.7 [MDMPP] FCS\_COP.1(4).....74

8.3.8 [MDMPP] FCS\_RBG\_EXT.1 .....75

8.3.9 [MDMPP] FCS\_STG\_EXT.1 .....75

8.3.10 [AGENTMOD] FCS\_STG\_EXT.1(2).....76

8.4 Identification and Authentication.....76

8.4.1 [MDMPP] FIA\_ENR\_EXT.1/ANDROID .....76

8.4.2 [MDMPP] FIA\_ENR\_EXT.1/IOS .....77

8.4.3 [AGENTMOD] FIA\_ENR\_EXT.2.....78

8.4.4 [MDMPP] FIA\_UAU.1 .....78

8.4.5 [MDMPP] FIA\_X509\_EXT.1(1) and [MDMPP] FIA\_X509\_EXT.2 .....78

8.4.6 [MDMPP] FIA\_X509\_EXT.5 .....81

8.5 Security Management .....81

8.5.1 [MDMPP] FMT\_MOF.1(1).....81

8.5.2 [MDMPP] FMT\_MOF.1(2).....82

8.5.3 [MDMPP] FMT\_MOF.1(3).....82

8.5.4 [MDMPP] FMT\_POL\_EXT.1 .....83

8.5.5 [AGENTMOD] FMT\_POL\_EXT.2.....83

8.5.6 [MDMPP] FMT\_SMF.1(1)/ANDROID and [MDMPP] FMT\_SMF.1(1)/IOS .....84

8.5.7 [MDMPP] FMT\_SMF.1(2)/ANDROID and [MDMPP] FMT\_SMF.1(2)/IOS .....89

8.5.8 [MDMPP] FMT\_SMF.1(3).....90

|        |   |     |
|--------|---|-----|
| 8.5.9  | [AGENTMOD] FMT_SMF_EXT.4 .....                  | 91  |
| 8.5.10 | [MDMPP] FMT_SMR.1(1) .....                      | 91  |
| 8.5.11 | [MDMPP] FMT_SMR.1(2) .....                      | 92  |
| 8.5.12 | [AGENTMOD] FMT_UNR_EXT.1 .....                  | 92  |
| 8.6    | Protection of the TSF .....                     | 93  |
| 8.6.1  | [MDMPP] FPT_API_EXT.1 .....                     | 93  |
| 8.6.2  | [MDMPP] FPT_ITT.1(2).....                       | 93  |
| 8.6.3  | [MDMPP] FPT_LIB_EXT.1 .....                     | 94  |
| 8.6.4  | [MDMPP] FPT_TST_EXT.1 .....                     | 94  |
| 8.6.5  | [MDMPP] FPT_TUD_EXT.1.....                      | 95  |
| 8.7    | TOE Access .....                                | 95  |
| 8.7.1  | [MDMPP] FTA_TAB.1 .....                         | 95  |
| 8.8    | Trusted Path/Channels .....                     | 95  |
| 8.8.1  | [MDMPP] FTP_ITC_EXT.1 .....                     | 95  |
| 8.8.2  | [MDMPP] FTP_ITC.1(1).....                       | 96  |
| 8.8.3  | [MDMPP] FTP_TRP.1(1).....                       | 96  |
| 8.8.4  | [MDMPP] FTP_TRP.1(2).....                       | 97  |
| 9      | Appendix A: List of Third-Party Libraries ..... | 98  |
| 9.1    | Windows Server 2019 Libraries .....             | 98  |
| 9.2    | Android 11 Libraries.....                       | 100 |
| 9.3    | iOS 14 and iPadOS 14 Libraries .....            | 101 |

## **Table of Figures**

|                              |    |
|------------------------------|----|
| Figure 1: TOE Boundary ..... | 11 |
|------------------------------|----|

## **Table of Tables**

|   |   |
|---|---|
| Table 1: Customer-Specific Terminology..... | 7 |
| Table 2: CC-Specific Terminology.....       | 8 |
| Table 3: Acronym Definitions .....          | 8 |

Table 4: Evaluated Components of the TOE .....13

Table 5: Components of the Operational Environment .....13

Table 6: Cryptographic Algorithm Table for the Hub Agents .....16

Table 7: TOE Threats.....23

Table 8: TOE Organizational Security Policies .....23

Table 9: TOE Assumptions.....24

Table 10: TOE Objectives .....24

Table 11: Operational Environment Objectives.....25

Table 12: Security Functional Requirements for the TOE.....27

Table 13: Server Auditable Events .....30

Table 14: Agent Auditable Events .....33

Table 15: SFR and TOE Component Mapping.....61

Table 16: Auditable Events by Enforcing Component .....65

Table 17: Agent Auditable Events by Enforcing Component .....68

Table 18: Keys and CSPs for the UEM Server Platform .....75

Table 19: Keys and CSPs for the Device .....76

Table 20: UEM Server Management Functions .....84

# 1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

## 1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation.

### 1.1.1 ST Identification

**ST Title:** VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target  
**ST Version:** 1.0  
**ST Publication Date:** February 6, 2023  
**ST Author:** Booz Allen Hamilton

### 1.1.2 Document Organization

*Chapter 1* of this document provides identifying information for the ST and TOE as well as a brief description of the TOE and its associated TOE type.

*Chapter 2* describes the TOE in terms of its physical boundary, logical boundary, exclusions, and dependent Operational Environment components.

*Chapter 3* describes the conformance claims made by this ST.

*Chapter 4* describes the threats, assumptions, objectives, and organizational security policies that apply to the TOE.

*Chapter 5* defines extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

*Chapter 6* describes the SFRs that are to be implemented by the TSF.

*Chapter 7* describes the SARs that will be used to evaluate the TOE.

*Chapter 8* provides the TOE Summary Specification, which describes how the SFRs that are defined for the TOE are implemented by the TSF.

### 1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1 and 2. These tables are to be used by the reader as a quick reference guide for terminology definitions.

**Table 1: Customer-Specific Terminology**

| <b>Term</b>          | <b>Definition</b>   |
|----------------------|---|
| <b>End User</b>      | An individual who possesses a mobile device that is managed by VMware and who has limited authority to perform management functions using the Self-Service Portal |
| <b>Internal Apps</b> | Apps which are stored on the TOE's MAS Server.  |



|                             |  |
|-----------------------------|--|
| <b>Managed Apps</b>         | Apps which are stored on the Google Play Store and/or Apple App Store that are installed on the mobile device due to TOE policies.                     |
| <b>Role</b>                 | The level of access given to Administrator accounts. The TOE comes with pre-defined roles but new roles with custom sets of privileges can be created. |
| <b>System Administrator</b> | The class of TOE Administrators that have complete access to a VMware environment, including the underlying Windows Server 2019 platform.              |

Table 2: CC-Specific Terminology

| <b>Term</b>                     | <b>Definition</b>  |
|---------------------------------|--|
| <b>Administrator</b>            | The claimed Protection Profile defines an Administrator as the person who is responsible for management activities, including setting the policy that is applied by the enterprise on the mobile device. This TOE defines separate user roles. |
| <b>Authorized Administrator</b> | Synonymous with Administrator.   |
| <b>MD User</b>                  | User with a mobile device (MD).  |
| <b>Trusted Channel</b>          | An encrypted connection between the TOE and a system in the Operational Environment.   |
| <b>Trusted Path</b>             | An encrypted connection between the TOE and the application an Authorized Administrator uses to manage it (web browser, terminal client, etc.).  |
| <b>User</b>                     | In a CC context, any individual who has the ability to manage TOE functions or data.   |

#### 1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Table 3: Acronym Definitions

| <b>Acronym</b> | <b>Definition</b>  |
|----------------|--|
| <b>APNS</b>    | Apple Push Notification Service  |
| <b>CA</b>      | Certificate Authority  |
| <b>CC</b>      | Common Criteria  |
| <b>CPU</b>     | Central Processing Unit  |
| <b>CSP</b>     | Critical Security Parameter  |
| <b>DEP</b>     | [Apple] Device Enrollment Program  |
| <b>FCM</b>     | [Android] Firebase Cloud Messaging [Service]                                 |
| <b>FQDN</b>    | Fully Qualified Domain Name  |
| <b>GUI</b>     | Graphical User Interface   |
| <b>HTTP</b>    | Hypertext Transfer Protocol  |
| <b>HTTPS</b>   | Hypertext Transfer Protocol Secure over a bidirectional TLS encrypted tunnel |
| <b>IMEI</b>    | International Mobile Equipment Identity                                      |
| <b>IP</b>      | Internet Protocol  |
| <b>IIS</b>     | Internet Information Services  |
| <b>IT</b>      | Information Technology   |
| <b>LDAP</b>    | Lightweight Directory Access Protocol  |
| <b>MAS</b>     | Mobile Application Store   |
| <b>MD</b>      | Mobile Device  |
| <b>MDM</b>     | Mobile Device Management   |
| <b>NFC</b>     | Near-Field Communication   |
| <b>NIAP</b>    | National Information Assurance Partnership                                   |

|             |                                 |
|-------------|---------------------------------|
| <b>OS</b>   | Operating System                |
| <b>OSP</b>  | Organizational Security Policy  |
| <b>PP</b>   | Protection Profile              |
| <b>SAR</b>  | Security Assurance Requirement  |
| <b>SFP</b>  | Security Function Policy        |
| <b>SFR</b>  | Security Functional Requirement |
| <b>SSL</b>  | Secure Sockets Layer            |
| <b>ST</b>   | Security Target                 |
| <b>TCP</b>  | Transmission Control Protocol   |
| <b>TLS</b>  | Transport Layer Security        |
| <b>TOE</b>  | Target of Evaluation            |
| <b>TSF</b>  | TOE Security Function           |
| <b>UEM</b>  | Unified Endpoint Management     |
| <b>UI</b>   | User Interface                  |
| <b>USB</b>  | Universal Serial Bus            |
| <b>VPN</b>  | Virtual Private Network         |
| <b>WLAN</b> | Wireless Local Area Network     |

### 1.1.5 Reference

- [1] Protection Profile for Mobile Device Management, version 4.0 [MDMPP]
- [2] Protection Profile Module for Mobile Device Management Agents, version 1.0 [AGENTMOD]
- [3] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001
- [4] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002
- [5] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003
- [6] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-004
- [7] NIST Special Publication 800-56A Revision 3: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, April 2018
- [8] NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007
- [9] NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
- [10] NIST Special Publication 800-57 Part 1 Revision 4, Recommendation for Key Management, January 2016
- [11] FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules, May 25, 2001
- [12] FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS), August 2015
- [13] FIPS PUB 186-4 Digital Signature Standard (DSS), July 2013
- [14] FIPS PUB 197 Advanced Encryption Standard, November 26, 2001

- [15] FIPS PUB 198-1 Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC), July 2008
- [16] VMware Workspace ONE Unified Endpoint Management v2209 Supplemental Administrative Guidance version 1.0
- [17] Apple iOS 14: iPhones Security Target, Version 1.5 (VID 11146)
- [18] Apple iPadOS 14: iPads Security Target, Version 1.5 (VID 11147)
- [19] Samsung Electronics Co., Ltd. Samsung Galaxy Devices on Android 11 – Spring Security Target, Version 1.0 (VID 11160)
- [20] Microsoft Windows Common Criteria Evaluation Microsoft Windows 10 version 1809 (October 2018 Update) Microsoft Windows Server version 1809 (October 2018 Update) Security Target, Version 0.05

## **1.2 TOE Reference**

The TOE is the VMware Workspace ONE Unified Endpoint Management version 2209 comprising of the Unified Endpoint Management Server and one or more VMware Intelligent Hubs installed on Apple and Android devices. The minimum configuration for this evaluation is one Unified Endpoint Management Server, and one VMware Intelligent Hub installed on an Apple device and/or one VMware Intelligent Hub installed on an Android device. Including additional VMware Intelligent Hubs installed on multiple Apple devices and additional VMware Intelligent Hubs installed on multiple Android devices as part of an operational configuration will not affect the validity of the functional claims made within this document and the Common Criteria certification.

## **1.3 TOE Overview**

The TOE is a Mobile Device Management product and is comprised of an MDM Server component (UEM Server) and one or more agent components called VMware Intelligent Hubs (iOS Hub agent and Android Hub agent). In the evaluated configuration of the TOE, the UEM Server is deployed in an on-premises configuration. The UEM Server component provides a centralized enterprise level management capability for a collection of mobile devices running the iOS and Android Hub agents. The UEM Server is also a Mobile Application Store (MAS) Server that allows managed devices to download apps from a trusted repository that resides within the organization managing the device. The management functionality includes management of Administrators and users, mobile device enrollment, mobile device status, mobile device compliance and policy management, and application management. Administrators access the UEM Server through the Admin Console interface in order to manage users, policies, and devices. Users access the UEM Server through the Self-Service Portal, which allows them to perform administrative functions relating to their own devices.

The UEM Server runs on a Microsoft Windows Server 2019 operating system and authentication to the UEM Server is provided by the Active Directory/LDAP service. The UEM Server also provides local authentication for initial setup and to mitigate a denial of service in the event the Active Directory/LDAP service is unavailable. The UEM Server stores audit data remotely in a SQL database but can send audit records via a TLS encrypted trusted channel to a remote Syslog Server for remote storage.

In the evaluated configuration, the Hub agent runs on mobile devices with either Apple iOS 14, Apple iPadOS 14, and/or Android 11 operating systems. The communication channel between the Hub agent and



## 1.4 TOE Type

The TOE is a Mobile Device Management product consisting of UEM Server software and one or more Hub agents which runs on mobile devices. The [MDMPP] states:

“The MDM Server is software (an application, service, etc.) on a general-purpose platform, a network device, or cloud architecture executing in a trusted network environment. The MDM Server provides administration of the mobile device policies and reporting on mobile device behavior. The MDM Server is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status, and sending commands to the Agents. The MDM Server may be standalone or distributed, where a distributed TOE is one that requires multiple distinct components to operate as a logical whole in order to fulfill the requirements of this PP.”

The MDM Server TOE type is justified because the TOE software provides centralized enterprise level management capabilities for MDM Agents (iOS and Android Hub agents) running on mobile devices, including enrollment, policy management and device status and the MDM Server (UEM) runs on Microsoft Windows Server 2019, which is a general-purpose platform.

The [MDMPP] also states:

“The MDM Agent establishes a secure connection back to the MDM Server controlled by an enterprise administrator and configures the mobile device per the administrator's policies. The MDM Agent is addressed in the Module for MDM Agents. If the MDM Agent is installed on a mobile device as an application developed by the MDM developer, the extends this PP and is included in the TOE. In this case, the TOE security functionality specified in this PP must be addressed by the MDM Agent in addition to the MDM Server. Otherwise, the MDM Agent is provided by the mobile device vendor and is out of scope of this PP; however, MDMs are required to indicate the mobile platforms supported by the MDM Server and must be tested against the native MDM agent of those platforms.”

This statement is re-iterated in the [AGENTMOD]. The MDM Agent TOE type is justified because the TOE Agent software (iOS and Android Hub agents) is installed on a mobile device as an application developed by VMware and establishes a secure connection back to the MDM Server (UEM Server) protected by HTTPS.

## 2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

### 2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

**Table 4: Evaluated Components of the TOE**

| Component  | Definition  |
|--|---|
| <b>Workspace ONE Unified Endpoint Management 2209 (UEM Server)</b> | This satisfies the MDM Server Component of the TOE as it provides an enterprise-level management capability for a collection of mobile devices, including the administration of mobile device policies, reporting on device behavior, and sending commands to the iOS and Android Hub agent(s). This MDM Server Component also provides a Mobile Application Store (MAS) Server that allows managed devices to download apps from a trusted repository. |
| <b>Android Intelligent Hub 22.09 (Android Hub agent)</b>           | This satisfies the MDM Agent Component of the TOE as it is a VMware-developed application installed on mobile devices running the Samsung Android 11 operating system and uses the Android platform to establish a secure connection back to the UEM Server for the Android Hub agent to provide status and policy information about the device.  |
| <b>iOS Intelligent Hub 22.11 (iOS Hub agent)</b>                   | This satisfies the MDM Agent Component of the TOE as it is a VMware-developed application installed on mobile devices running either the Apple iOS 14 and/or Apple iPadOS 14 operating systems. The iOS Hub agent uses the iOS/iPadOS platforms to establish a secure connection back to the UEM Server for the iOS Hub agent and iOS/iPadOS platform to provide status and policy information about the device.  |

As shown in Figure 1, the TOE boundary on the end user mobile devices includes only the iOS and Android Hub agents itself; the actual devices have been evaluated against the Mobile Device Fundamentals Protection Profile under the Validation ID number identified in Table 5 below.

### 2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the TOE's operational environment that must be present for the TOE to be operating in its evaluated configuration:

**Table 5: Components of the Operational Environment**

| Component                                       | Definition   |
|---|--|
| <b>Active Directory / LDAP Server</b>           | Identity store that defines users for device enrollment and administrator accounts for access to the Admin Console. In the evaluated configuration, Windows Server 2019 (Version 1809) Active Directory/LDAP Server is used. |
| <b>Apple iOS 14 Mobile Device (VID11146)</b>    | The MDM Agent Component of the TOE (Hub agent) is an application that is installed on Apple mobile devices running iOS 14 operating systems so that the TOE can provide management functionality to the device.              |
| <b>Apple iPadOS 14 Mobile Device (VID11147)</b> | The MDM Agent Component of the TOE (Hub agent) is an application that is installed on Apple mobile devices running iPadOS 14 operating systems so that the TOE can provide management functionality to the device.           |

|   |  |
|---|--|
| <b>Apple Push Notification Service (APNS) / Apple DEP</b> | APNS is an iOS/iPadOS platform push notification service that enables the UEM Server to notify iOS Hub agents and the iOS/iPadOS platform to connect directly to the UEM Server to retrieve data (e.g. policies). Apple DEP is an online service that automates the enrollment of iOS devices into the TOE in the evaluated configuration. |
| <b>Certification Authority (CA) Server</b>                | The MDM Server Component and Android Hub agent of the TOE connect to the CA Server during device enrollment so that the TOE can provide each device with a unique certificate generated by the CA Server. In the evaluated configuration, Windows Server 2019 (Version 1809) Active Directory Certificate Services is used.                |
| <b>Firebase Cloud Messaging Service (FCM)</b>             | FCM is an Android platform push notification service that enables the UEM Server to notify Android Hub agents to connect directly to the UEM Server to retrieve data (e.g. policies).  |
| <b>Samsung Android 11 Mobile Device (VID 11160)</b>       | The MDM Agent Component of the TOE (Hub agent) is an application that is installed on mobile devices running Android 11 operating systems so that the TOE can provide management functionality to the device.  |
| <b>SQL database</b>                                       | The TOE's RDBMS database used to store configuration settings and device data. In the evaluated configuration, Microsoft SQL Server 2019 is used.  |
| <b>Syslog Server</b>                                      | The MDM Server Component of the TOE connects to the Syslog Server to persistently store audit data for the UEM Server's own operation as well as the audit data collected from the Hub agent that it manages.  |
| <b>Windows Server 2019 (Version 1809)</b>                 | This is the OS that the UEM Server is installed on.  |
| <b>Workstation</b>  | Any general-purpose computer that is used by an administrator to manage the TOE via the Admin Console and a user to manage their device via the Self-Service Portal. For the TOE to be accessed remotely, the workstation is required to have a browser to access the TOE's GUI based interfaces.  |

## 2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with the TOE but are not included in the evaluated configuration. They provide no added security related functionality for the evaluated product. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

### 2.3.1 Not Installed

There are no optional components that are omitted from the installation process.

### 2.3.2 Installed but Requires a Separate License

There are no excluded components that are installed and require a separate license.

### 2.3.3 Installed But Not Part of the TSF

This section contains functionality or components that are part of the purchased product but are not part of the TSF relevant functionality that is being evaluated as the TOE:

There are no discrete individual components that are excluded from the TSF. Note however that the logical boundary of the TOE only includes the functionality that satisfies the Security Functional Requirements in the claimed Protection Profiles. If the product provides functionality that is not used to satisfy any of these requirements, it is considered to be security-non-interfering functionality.

In particular, note that the VMware product also includes a Secure Email Gateway and Mobile Access Gateway. These components have not been evaluated because their functionality is outside the scope of the claimed Protection Profiles. However, their presence in the Operational Environment does not interfere with the security enforcement of the TSF and therefore can be deployed in an environment with the TOE.

## **2.4 Physical Boundary**

### **2.4.1 Hardware**

The TOE is a software product. All hardware that is present is part of the TOE's Operational Environment.

### **2.4.2 Software**

The VMware Workspace ONE UEM runs on Windows Server 2019 (Version 1809) and includes the Admin Console, Self-Service Portal, UEM Server, and MAS Server functions. The software version of the VMware Workspace ONE UEM is 2209 (displayed as 22.9).

The VMware Hub agent runs on Apple iOS 14, Apple iPadOS 14, and Android 11 operating systems in its evaluated configuration. The software version of the iOS Hub agent is 22.11. The software version of the Android Hub agent is 22.09 and includes OpenSSL version 2.0.16.

Refer to the VMware Workspace ONE Unified Endpoint Management v2209 Supplemental Administrative Guidance version 1.0 for information on delivery of the TOE software.

## **2.5 Logical Boundary**

The TOE is comprised of several security features. Each of these security features consists of several security functionalities, as identified below. This ST includes both UEM Server and the Hub agent functionality.

1. Security Audit
2. Communication
3. Cryptographic Support
4. Identification and Authentication
5. Security Management
6. Protection of the TSF
7. TOE Access
8. Trusted Path/Channels

### **2.5.1 Security Audit**

The UEM Server component of the TOE creates audit records for auditable events related to administrative actions, configuration of the UEM Server itself, and server-initiated management activities that affect one or more managed mobile devices. The UEM Server's MAS Server functionality also generates audit records when it experiences a failure to push or update an application on a managed mobile device. The audit records are stored in an SQL database and are transferred to a remote Syslog Server over a TLS encrypted trusted channel. Audit records can be viewed on the Admin Console.



The UEM Server can issue ‘compliance policies’ to managed mobile devices. Compliance policies are used to compare the configuration, status, or characteristics of a mobile device against a certain baseline and can be used to generate an alert to an Administrator if an anomaly is detected. The Administrator can also request on-demand connectivity status updates through the use of push notifications.

iOS and Android Hub agents’ audit records are created as long as the underlying mobile device is powered on. The iOS and Android Hub agents generate audit records for the activities it performs as a result of its interactions with the UEM Server or as a result of stored policy information. The iOS and Android Hub agents facilitate alerts by providing data to the UEM Server on a periodic basis. The UEM Server can then analyze this data (or the absence of data in the case of periodic reachability events) in order to determine if anomalous behavior is occurring.

**2.5.2 Communication**

The iOS and Android Hub agents mobile devices are registered with the UEM Server so they can be enrolled into management by the UEM Server. This requires an Administrator to enable communications between these TOE components by including the mobile device’s identifier in an allow list of devices that are allowed to enroll on the UEM Server. The enrollment process occurs over an HTTPS/TLS trusted channel that is handled by each TOE components’ underlying platform. An Administrator can disable the communications between an iOS or Android Hub agent and the UEM Server by performing a wipe of the Hub agent’s mobile device.

**2.5.3 Cryptographic Support**

The UEM Server invokes the Windows Server 2019 platform for cryptographic services to establish TLS and HTTPS/TLS trusted channels and paths to ensure secure communications of data in transit. This includes the use of RSA and Elliptic Curve Cryptography (ECC) key establishment techniques. The MAS Server is integrated with the UEM Server, so it invokes the same cryptography services. The UEM Server also invokes the Windows Server 2019 platform to digitally sign policies sent to the Hub agents.

The iOS and Android Hub agents invoke their underlying mobile device platforms (Apple iOS 14, Apple iPadOS 14, and Android 11 respectively) for cryptographic services to also establish trusted communications. The iOS Hub agent invokes its underlying platform to verify the digital signatures of all policies received from the UEM Server. The Android Hub agent software contains an OpenSSL library for implementing the digital signature verification of all policies received from the UEM Server.

All cryptographic mechanisms use the TOE components’ platform provided DRBG functionality to support their cryptographic operations. Cryptographic functionality includes encryption/decryption services, credential/key storage, key establishment, key destruction, hashing services, signature services, and hashed message authentication.

The following table contains the CAVP algorithm certificates corresponding to the Android Hub agent’s digital signature verification cryptographic functionality which is implemented by its OpenSSL module.

**Table 6: Cryptographic Algorithm Table for the Hub Agents**

|                                     | <b>Algorithm</b>            | <b>CAVP Cert. #<br/>(Android 11)</b> |
|-------------------------------------|-----------------------------|--------------------------------------|
| FCS_COP.1(2) – Hashing Algorithms   | SHA-512                     | A3270                                |
| FCS_COP.1(3) – Signature Algorithms | ECDSA with P-521 NIST curve | A3270                                |

#### **2.5.4 Identification and Authentication**

The iOS and Android Hub agents register with the UEM Server so that their mobile device can be enrolled into management by the UEM Server. The mobile device user that is performing the enrollment must have a user account on the UEM Server to access the Self-Service Portal and authenticate to the TOE. During the enrollment process, the iOS and Android Hub agents record the UEM Server's DNS name and full URL with hostname. The iOS and Android Hub agents also receive a unique certificate during enrollment that is used to establish an HTTPS trusted channel with the UEM Server.

Administrators (through the Admin Console) and users (through the Self-Service Portal) cannot access the UEM Server without being authenticated. Administrators and users can view the configured pre-authentication warning banner and query the UEM Server's software version number prior to authentication.

The UEM Server interfaces with the underlying Windows Server 2019 platform to provide certificate validation services. Certificates are used for HTTPS/TLS authentication, code signing for software updates, code signing for integrity verification, and signing of MDM policies. The iOS and Android Hub agents rely on the underlying platform to perform all certificate validation services, except for policy signing on Android devices which is validated by the Android Hub agent's implementation of OpenSSL.

#### **2.5.5 Security Management**

The TSF provides separate administrative interfaces for Administrators and for mobile device users. Administrators use the Admin Console to manage users, policies, and devices, while MD users use the Self-Service Portal to perform actions related to their own devices. The mobile device user installs the TOE's iOS or Android Hub agent on the mobile device which will communicate with the UEM Server to enroll in management. Once enrolled, the TOE will prevent user-directed unenrollment from management.

The UEM Server can be used to transmit specific commands to a managed device such as forcibly locking the device, initiating a wipe operation, or sending a push notification. The UEM Server can also define policies (known as profiles) that specify the configuration settings for a device. These configuration settings can include functionality such as configuration of the password policy and what settings are applied to Wi-Fi connections. The UEM Server transmits iOS policies either to the iOS Hub agent or iOS/iPadOS platform directly, depending on the functionality being configured. The UEM Server transmits Android policies to the Android Hub agent. The UEM Server invokes its underlying platform to sign all policy data using ECDSA with SHA-512. The underlying iOS/iPadOS mobile platform and Android Hub agent will validate the signed policies when they are received.

The UEM Server also includes the MAS Server functionality, which provides the ability to grant or deny access to specific applications stored on the MAS Server to devices or groups of devices. The MAS Server is accessed through the same Admin Console interface as the UEM Server, so the administrative roles defined for both components are the same.

#### **2.5.6 Protection of the TSF**

The communications between the UEM Server and iOS and Android Hub agents are protected using HTTPS/TLS which is provided by the underlying platforms of the TOE components.

The UEM Server invokes its platform to verify the digital signatures of executables and .dlls using Microsoft's Authenticode making use of X.509v3 certificates. In addition, the UEM Server's platform uses FIPS validated cryptographic modules which perform their own integrity checks at startup.

The TOE components invoke their underlying platforms to update their software and the platforms will verify the digital signatures of the updates prior to installing them. The TOE components' software contains third party libraries. The TOE components use only documented APIs from their underlying platforms.

### **2.5.7 TOE Access**

The UEM Server displays a pre-authentication banner for the Admin Console and the Self-Service Portal. This can be customized by Administrators to fit the needs of the organization deploying the TOE.

### **2.5.8 Trusted Path/Channels**

The trusted communication channels between the UEM Server and the devices running the iOS and Android Hub agents, the Syslog Server, and the AD/LDAP Server make use of TLS or HTTPS/TLS, depending on the interface. The trusted communication channels are provided by the TOE components' underlying platforms.

The UEM Server platform uses HTTPS/TLS to provide a trusted path between itself and remote Administrators through the Admin Console and mobile device users through the Self-Service Portal as well as during the enrollment of a mobile device.

## 3 Conformance Claims

### 3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5, April 2017.

### 3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) are Part 2 (extended) to include all applicable NIAP and International interpretations through February 6, 2023.

### 3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are Part 3 (conformant) to include all applicable NIAP and International interpretations through February 6, 2023.

### 3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- Protection Profile for Mobile Device Management, version 4.0 [MDMPP]
- Protection Profile Module for Mobile Device Management Agents, version 1.0 [AGENTMOD]

### 3.5 Package Claims

The TOE claims exact compliance to the *Protection Profile for Mobile Device Management, version 4.0* and *Protection Profile Module for Mobile Device Management Agents, version 1.0*, which are conformant with CC Part 3.1.

The TOE claims the following Selection-Based SFRs that are defined in the appendices of the claimed PP:

[MDMPP]:

- FAU\_GEN.1(2)
- FMT\_MOF.1(3)
- FMT\_SMF.1(3)
- FMT\_SMR.1(2)
- FPT\_ITT.1(2)

The TOE claims the following Optional SFRs that are defined in the appendices of the claimed PP:

[MDMPP]:

- FAU\_SAR.1
- FTA\_TAB.1

The TOE claims the following Objective SFRs that are defined in the appendices of the claimed PP:

[MDMPP]:

- FCO\_CPC\_EXT.1

This does not violate the notion of exact conformance because the PP specifically indicates these as allowable selections, options, and objectives, and provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

### **3.6 Package Name Conformant or Package Name Augmented**

This ST and TOE claim exact conformance to the [MDMPP] and [AGENTMOD].

### **3.7 Conformance Claim Rationale**

The [MDMPP] states the following:

“The MDM Server is software (an application, service, etc.) on a general-purpose platform, a network device, or cloud architecture executing in a trusted network environment. The MDM Server provides administration of the mobile device policies and reporting on mobile device behavior. The MDM Server is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status, and sending commands to the Agents. The MDM Server may be standalone or distributed, where a distributed TOE is one that requires multiple distinct components to operate as a logical whole in order to fulfill the requirements of this PP”

The [AGENTMOD] states the following:

“The MDM system consists of two primary components: the MDM Server software and the MDM Agent. This PP-Module specifically addresses the MDM Agent. The MDM Agent establishes a secure connection back to the MDM Server, from which it receives policies to enforce on the mobile device. Optionally, the MDM Agent interacts with the Mobile Application Store (MAS) Server to download and install enterprise-hosted applications.

A compliant MDM Agent is installed on a mobile device as an application (supplied by the developer of the MDM Server software) or is part of the mobile device's OS. This PP-Module builds on either the MDF PP or the MDM PP. A TOE that claims conformance to this PP-Module must also claim conformance to one of those PPs as its Base-PP. A compliant TOE is obligated to implement the functionality required in the Base-PP along with the additional functionality defined in this PP Module in order to mitigate the threats that are defined by this PP-Module.

This PP-Module shall build on the MDF PP if the TOE is a native part of a mobile operating system. The TOE for this PP Module combined with the MDF PP is the mobile device itself plus the MDM Agent. If the MDM Agent is part of the mobile device's OS, the MDM Agent may present multiple interfaces for configuring the mobile device, such as a local interface and a remote interface. Agents conforming to this PP-Module must at least offer an interface with a trusted channel that serves as one piece of an MDM system. Conformant MDM Agents may also offer other interfaces, and the configuration aspects of these additional interfaces are in scope of this PP-Module.

This PP-Module shall build on the MDM Server PP if the TOE is a third-party application that is provided with an MDM Server and installed on a mobile device by the user after acquiring the mobile device. The distributed TOE for this PP-Module combined with the MDM Server PP is the entire MDM environment, which includes both the MDM Server and the MDM Agent. Even though the mobile device itself is not part of the TOE, it is expected to be evaluated against the MDF PP so that its baseline security capabilities can be assumed to be present.”

The MDM Server component (UEM Server) of the TOE is designed to provide centralized management capabilities of the MDM Agent components (iOS and Android Hub agents) of the TOE which reside on mobile devices. The iOS and Android Hub agent communicates with the UEM Server over a secure channel. Therefore, the conformance claim is appropriate.

### 3.8 Technical Decisions

The following is the list of NIAP Technical Decisions that are applicable to the ST/TOE and a summary of their impact:

| TD #   | Title  | References  | Changes |    |       | Analysis to this evaluation |  |
|--------|--|---|---------|----|-------|-----------------------------|--|
|        |  |   | SFR     | AA | Notes | NA                          | Reason   |
| TD0438 | <a href="#">TST and TUD on the MDM Agent</a>                                     | [MDMPP]<br>FPT_TST_EXT.1 and<br>FPT_TUD_EXT.1.1     | X       |    | X     |                             | Clarifying distributed TOE components<br>Footnote 5,6,7  |
| TD0461 | <a href="#">Security Audit for Distributed TOEs</a>                              | [MDMPP]<br>Section 6.2.2, Bullet 2                  |         |    | X     |                             | Clarify audit transfer on distributed TOE  |
| TD0462 | <a href="#">MDM Distributed TOE: Registration Channel Updates</a>                | [MDMPP]<br>Section 3.1 and<br>FCO_CPC_EXT.1         | X       | X  | X     |                             | Update registration channel selection for distributed TOE<br>Footnote 3                                |
| TD0479 | <a href="#">FMT_SMF.1(1) Reliance on MDF Evals</a>                               | [MDMPP]<br>FMT_SMF.1(1)                             |         | X  | X     |                             | TOE claims more functions than evaluation  |
| TD0491 | <a href="#">Update to FMT_SMF_EXT.4 Test 2</a>                                   | [AGENTMOD]<br>FMT_SMF_EXT.4                         |         | X  |       |                             | Updates wording of Test 2  |
| TD0497 | <a href="#">SFR Rationale, Consistency of SPD, and Implicitly Satisfied SFRs</a> | [AGENTMOD]<br>Section 5, Section 6, and Appendix H  |         |    | X     |                             | Clarifies SFR rationale, consistency of the security problem definition, and implicitly satisfied SFRs |
| TD0552 | <a href="#">SFR Rationale and Implicitly Satisfied SFRs</a>                      | [MDMPP]<br>Section 6 and Appendix I                 |         |    | X     |                             | Clarifies SFR rationale and implicitly satisfied SFRs  |
| TD0594 | <a href="#">Distributed TOE tests in FCO_CPC_EXT.1.3</a>                         | [MDMPP]<br>FCO_CPC_EXT.1.3                          |         | X  |       |                             | Clarifies tests for a claimed SFR  |
| TD0600 | <a href="#">Conformance claim sections updated to allow for MOD_VPNC_V2.3</a>    | [AGENTMOD]<br>Section 2 and<br>[MDMPP]<br>Section 2 |         |    | X     | X                           | PP-Module for VPN Clients, Version 2.3 was not included as a conformance claim for this evaluation     |
| TD0616 | <a href="#">MDM PP Use Case Mappings</a>   | [MDMPP]<br>Appendix G                               |         |    | X     |                             | Fixes references   |
| TD0629 | <a href="#">Audit Events for Startup and Shutdown</a>                            | [MDMPP]<br>FAU_GEN.1.1(1)                           | X       |    | X     |                             | Updates start up and shut down audit events to a selection<br>Footnote 1                               |
| TD0641 | <a href="#">Alternative revocation checking for MDM</a>                          | [MDMPP]<br>FIA_X509_EXT.1(1)                        | X       | X  | X     |                             | Updates to revocation checking and validation of ECC certificates<br>Footnote 4                        |

|        |  |   |   |  |   |   |  |
|--------|--|---|---|--|---|---|--|
| TD0650 | <a href="#">Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4</a>  | [AGENTMOD]<br>Section 2<br>[MDMPP]<br>Section 2 |   |  | X | X | PP-Module for VPN Clients, Version 2.4 was not included as a conformance claim for this evaluation                       |
| TD0660 | <a href="#">Mislabelled SFRs in MDM Agent Auditable Events Table</a>                   | [AGENTMOD]<br>FAU_GEN.1(2)                      | X |  |   |   | Addresses mislabelled SFRs in audit table Footnote 2   |
| TD0673 | <a href="#">MDM-Agent PP-Module updated to allow for new PP and PP-Module Versions</a> | [AGENTMOD]<br>Section 1.1, Section 2            |   |  |   | X | Address the addition of PP modules that can now be using in conjunction with the PP_MDF. Does not affect this evaluation |

Note that Technical Decisions were not considered to be applicable if any of the following conditions were true:

- The Technical Decision does not apply to the [MDMPP] or [AGENTMOD].
- The Technical Decision applies to an SFR that was not claimed by the TOE.
- The Technical Decision was superseded by a more recent Technical Decision.
- The Technical Decision is issued as guidance for future versions of the [MDMPP] or [AGENTMOD].

## 4 Security Problem Definition

### 4.1 Threats

Note: Unless otherwise stated Threats, Organizational Security Policies (OSPs), Assumptions and Security Objectives apply to both the Agent and Server.

This section identifies the threats against the TOE. These threats have been taken from the [MDMPP] and [AGENTMOD].

Table 7: TOE Threats

| Threat                     | Threat Definition   |
|----------------------------|---|
| <b>T.MALICIOUS_APPS</b>    | [MDMPP] Malicious or flawed application threats exist because apps loaded onto a mobile device may include malicious or exploitable code. An administrator of the MDM or mobile device user may inadvertently import malicious code, or an attacker may insert malicious code into the TOE, resulting in the compromise of TOE or TOE data. |
| <b>T.BACKUP</b>            | [AGENTMOD] An attacker may try to target backups of data or credentials and exfiltrate data. Since the backup is stored on either a personal computer or end user’s backup repository, it’s not likely the enterprise would detect compromise.  |
| <b>T.NETWORK_ATTACK</b>    | [MDMPP] An attacker may masquerade as an MDM Server and attempt to compromise the integrity of the mobile device by sending malicious management commands.  |
| <b>T.NETWORK_EAVESDROP</b> | [MDMPP] An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the MDM Server and other endpoints.   |
| <b>T.PHYSICAL_ACCESS</b>   | [MDMPP] The mobile device may be lost or stolen, and an unauthorized individual may attempt to access user data. Although these attacks are primarily directed against the mobile device platform, the TOE configures features, which address these threats.  |

### 4.2 Organizational Security Policies

This section identifies the organizational security policies which are expected to be implemented by an organization that deploys the TOE. These policies have been taken from the [MDMPP] and [AGENTMOD].

Table 8: TOE Organizational Security Policies

| Policy                  | Policy Definition  |
|-------------------------|--|
| <b>P.ACCOUNTABILITY</b> | Personnel operating the TOE shall be accountable for their actions within the TOE.   |
| <b>P.ADMIN</b>          | The configuration of the mobile device security functions must adhere to the Enterprise security policy.   |
| <b>P.DEVICE_ENROLL</b>  | A mobile device must be enrolled for a specific user by the administrator of the MDM prior to being used in the Enterprise network by the user.                            |
| <b>P.NOTIFY</b>         | The mobile user must immediately notify the administrator if a mobile device is lost or stolen so that the administrator may apply remediation actions via the MDM system. |

### 4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the TOE’s Operational Environment. These assumptions have been taken from the [MDMPP] and [AGENTMOD].



Table 9: TOE Assumptions

| Assumption  | Assumption Definition   |
|---|---|
| <b>A.COMPONENTS_RUNNING</b><br>(applies to distributed TOEs only) | [MDMPP] For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.  |
| <b>A.CONNECTIVITY</b>   | The TOE relies on network connectivity to carry out its management activities. The TOE will robustly handle instances when connectivity is unavailable or unreliable.   |
| <b>A.MDM_SERVER_PLATFORM</b>                                      | [MDMPP] The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities. The MDM Server relies on this platform to provide a range of security-related services including reliable timestamps, user and group account management, logon and logout services via a local or network directory service, remote access control, and audit log management services to include offloading of audit logs to other servers. The platform is expected to be configured specifically to provide MDM services, employing features such as a host-based firewall, which limits its network role to providing MDM functionality. |
| <b>A.MOBILE_DEVICE_PLATFORM</b>                                   | [AGENTMOD] The MDM Agent relies upon mobile platform and hardware evaluated against the MDF PP and assured to provide policy enforcement as well as cryptographic services and data protection. The mobile platform provides trusted updates and software integrity verification of the MDM Agent.  |
| <b>A.PROPER_ADMIN</b>   | One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the TOE Administrators, and do so using and abiding by guidance documentation.   |
| <b>A.PROPER_USER</b>  | Mobile device users are not willfully negligent or hostile, and use the device within compliance of a reasonable Enterprise security policy.  |

## 4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

### 4.4.1 TOE Security Objectives

This section identifies the security objectives of the TOE. These objectives have been taken directly from the [MDMPP] and [AGENTMOD].

Table 10: TOE Objectives

| Objective                        | Objective Definition   |
|----------------------------------|--|
| <b>O.ACCOUNTABILITY</b>          | The TOE must provide logging facilities which record management actions undertaken by its administrators.  |
| <b>O.APPLY_POLICY</b>            | The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the mobile OS and the MDM Server. This will include the initial enrollment of the device into management through its entire lifecycle, including policy updates and its possible unenrollment from management services. |
| <b>O.DATA_PROTECTION_TRANSIT</b> | Data exchanged between the MDM Server and the MDM Agent must be protected from being monitored, accessed, or altered.  |

|                     |  |
|---------------------|--|
| <b>O.INTEGRITY</b>  | [MDMPP] The TOE will provide the capability to perform self tests to ensure the integrity of critical functionality, software, firmware, and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates.  |
| <b>O.MANAGEMENT</b> | [MDMPP] The TOE provides access controls around its management functionality.  |
| <b>O.QUALITY</b>    | [MDMPP] To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs. |
| <b>O.STORAGE</b>    | [AGENTMOD] To address the issue of loss of confidentiality of user data in the event of loss of a Mobile Device (T.PHYSICAL), conformant TOEs will use platform provide key storage. The TOE is expected to protect its persistent secrets and private keys.   |

#### 4.4.2 Security Objectives for the Operational Environment

The TOE’s operational environment must satisfy the following objectives:

**Table 11: Operational Environment Objectives**

| <b>Objective</b>   | <b>Objective Definition</b>  |
|--|--|
| <b>OE.COMPONENTS_RUNNING</b>   | [MDMPP] For distributed TOEs the Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly. |
| <b>OE.PROPER_ADMIN [MDMPP]</b><br><b>OE.DATA_PROPER_ADMIN [AGENTMOD]</b> | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.   |
| <b>OE.PROPER_USER [MDMPP]</b><br><b>OE.DATA_PROPER_USER [AGENTMOD]</b>   | Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.   |
| <b>OE.IT_ENTERPRISE</b>  | The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access.  |
| <b>OE.MOBILE_DEVICE_PLATFORM</b>   | [AGENTMOD] The MDM Agent relies upon the trustworthy mobile platform and hardware to provide policy enforcement as well as cryptographic services and data protection. The mobile platform provides trusted updates and software integrity verification of the MDM Agent.  |
| <b>OE.TIMESTAMP</b>  | [MDMPP] Reliable timestamp is provided by the operational environment for the TOE.   |
| <b>OE.WIRELESS_NETWORK</b>   | A wireless network will be available to the mobile devices.  |

#### 4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profiles to which the TOE claims conformance.

## **5 Extended Components Definition**

### **5.1 Extended Security Functional Requirements**

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PPs to which the ST and TOE claim conformance. These extended components are formally defined in the PPs in which their usage is required.

### **5.2 Extended Security Assurance Requirements**

There are no extended Security Assurance Requirements in this ST.

## 6 Security Functional Requirements

### 6.1 Conventions

The CC permits four functional component operations—assignment, refinement, selection, and iteration—to be performed on functional requirements. This ST will highlight the operations in the following manner:

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized text*.
- **Refinement:** allows the addition of details. Indicated with **bold text**.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration operation:** are identified with a number inside parentheses (e.g., "(1)") or a forward-slash with the device type (e.g., “/ANDROID”) when functionality differs between device type.

When multiple operations are combined, such as an assignment that is provided as an option within a selection or refinement, a combination of the text formatting is used.

Text that is formatted in a claimed PP, such as if the PP’s instantiation of the SFR has a refinement (bolded font), or a completed assignment (inside brackets), the formatting is not preserved when reproduced in this ST. Only the assignments and selections made by the ST author are within [brackets]. This is so that the reader can easily identify the operations that are performed by the ST author.

### 6.2 Security Functional Requirements Summary

The following tables list the SFRs claimed by the TOE per platform. SFRs that originate from the Mobile Device Management Protection Profile are denoted by a [MDMPP]; SFRs that originated from the Mobile Device Management Agents PP-Module are denoted by [AGENTMOD].

Table 12: Security Functional Requirements for the TOE

| Class Name                   | Component Identification         | Component Name                                       |
|------------------------------|----------------------------------|--|
| <b>Security Audit</b>        | [MDMPP] FAU_ALT_EXT.1            | Server Alerts  |
|                              | [AGENTMOD] FAU_ALT_EXT.2/ANDROID | Agent Alerts   |
|                              | [AGENTMOD] FAU_ALT_EXT.2/IOS     | Agent Alerts   |
|                              | [MDMPP] FAU_GEN.1(1)             | Audit Data Generation                                |
|                              | [MDMPP] FAU_GEN.1(2)             | Audit Generation (MAS Server)                        |
|                              | [AGENTMOD] FAU_GEN.1(2)          | Audit Data Generation                                |
|                              | [MDMPP] FAU_NET_EXT.1            | Network Reachability Review                          |
|                              | [MDMPP] FAU_SAR.1                | Audit Review   |
|                              | [AGENTMOD] FAU_SEL.1(2)          | Security Audit Event Selection                       |
| [MDMPP] FAU_STG_EXT.1        | External Trail Storage           |  |
| <b>Communication</b>         | [MDMPP] FCO_CPC_EXT.1            | Component Registration Channel Definition            |
| <b>Cryptographic Support</b> | [MDMPP] FCS_CKM.1                | Cryptographic Key Generation                         |
|                              | [MDMPP] FCS_CKM.2                | Cryptographic Key Establishment                      |
|                              | [MDMPP] FCS_CKM_EXT.4            | Cryptographic Key Destruction                        |
|                              | [MDMPP] FCS_COP.1(1)             | Cryptographic Operation (Confidentiality Algorithms) |

| Class Name                               | Component Identification      | Component Name   |
|--|-------------------------------|--|
|  | [MDMPP] FCS_COP.1(2)          | Cryptographic Operation (Hashing Algorithms)                                     |
|  | [MDMPP] FCS_COP.1(3)          | Cryptographic Operation (Signature Algorithms)                                   |
|  | [MDMPP] FCS_COP.1(4)          | Cryptographic Operation (Keyed-Hash Message Authentication)                      |
|  | [MDMPP] FCS_RBG_EXT.1         | Extended: Random Bit Generation  |
|  | [MDMPP] FCS_STG_EXT.1         | Cryptographic Key Storage  |
|  | [AGENTMOD] FCS_STG_EXT.1(2)   | Cryptographic Key Storage  |
| <b>Identification and Authentication</b> | [MDMPP] FIA_ENR_EXT.1/ANDROID | Enrollment of Mobile Device into Management (Android)                            |
|  | [MDMPP] FIA_ENR_EXT.1/IOS     | Enrollment of Mobile Device into Management (iOS)                                |
|  | [AGENTMOD] FIA_ENR_EXT.2      | Agent Enrollment of Mobile Device into Management                                |
|  | [MDMPP] FIA_UAU.1             | Timing of Authentication   |
|  | [MDMPP] FIA_X509_EXT.1(1)     | Validation of Certificates   |
|  | [MDMPP] FIA_X509_EXT.2        | X.509 Certification Authentication   |
|  | [MDMPP] FIA_X509_EXT.5        | X.509 Unique Certificate   |
| <b>Security Management</b>               | [MDMPP] FMT_MOF.1(1)          | Management of Functions Behavior   |
|  | [MDMPP] FMT_MOF.1(2)          | Management of Functions Behavior (Enrollment)                                    |
|  | [MDMPP] FMT_MOF.1(3)          | Management of Functions in (MAS Server Downloads)                                |
|  | [MDMPP] FMT_POL_EXT.1         | Trusted Policy Update  |
|  | [AGENTMOD] FMT_POL_EXT.2      | Agent Trusted Policy Update  |
|  | [MDMPP] FMT_SMF.1(1)/ANDROID  | Specification of Management Functions (Server Configuration of Agent) (Android)  |
|  | [MDMPP] FMT_SMF.1(1)/IOS      | Specification of Management Functions (Server Configuration of Agent) (iOS)      |
|  | [MDMPP] FMT_SMF.1(2)/ANDROID  | Specification of Management Functions (Server Configuration of Server) (Android) |
|  | [MDMPP] FMT_SMF.1(2)/IOS      | Specification of Management Functions (Server Configuration of Server) (iOS)     |
|  | [MDMPP] FMT_SMF.1(3)          | Specification of Management Functions (MAS Server)                               |
|  | [AGENTMOD] FMT_SMF_EXT.4      | Specification of Management Functions  |
|  | [MDMPP] FMT_SMR.1(1)          | Security Management Roles  |
|  | [MDMPP] FMT_SMR.1(2)          | Security Management Roles (MAS Server)   |
|  | [AGENTMOD] FMT_UNR_EXT.1      | User Unenrollment Prevention   |
| <b>Protection of the TSF</b>             | [MDMPP] FPT_API_EXT.1         | Use of Supported Services and API's  |
|  | [MDMPP] FPT_ITT.1(2)          | Internal TOE TSF Data Transfer (To MDM Agent)                                    |

| Class Name            | Component Identification | Component Name                                     |
|-----------------------|--------------------------|--|
|                       | [MDMPP] FPT_LIB_EXT.1    | Use of Third Party Libraries                       |
|                       | [MDMPP] FPT_TST_EXT.1    | Functionality Testing                              |
|                       | [MDMPP] FPT_TUD_EXT.1    | Trusted Update                                     |
| TOE Access            | [MDMPP] FTA_TAB.1        | Default TOE Access Banners                         |
| Trusted Path/Channels | [MDMPP] FTP_ITC_EXT.1    | Trusted Channel                                    |
|                       | [MDMPP] FTP_ITC.1(1)     | Inter-TSF Trusted Channel (Authorized IT Entities) |
|                       | [MDMPP] FTP_TRP.1(1)     | Trusted Path (for Remote Administration)           |
|                       | [MDMPP] FTP_TRP.1(2)     | Trusted Path (for Enrollment)                      |

## 6.3 Security Functional Requirements

### 6.3.1 Class FAU: Security Audit

---

#### 6.3.1.1 [MDMPP] FAU\_ALT\_EXT.1 *Server Alerts*

---

##### FAU\_ALT\_EXT.1.1

The TSF shall alert the administrators in the event of any of the following:

- a. Change in enrollment status
- b. Failure to apply policies to a mobile device
- c. *[[presence of apps on a deny list, presence of apps not on an allow list, absence of required apps, last time a device communicated with the MDM Server, unapproved model (iOS only), unapproved device manufacturer (Android only), unapproved operating system version, audit processing failure]]*

---

#### 6.3.1.2 [AGENTMOD] FAU\_ALT\_EXT.2/ANDROID *Agent Alerts*

---

##### FAU\_ALT\_EXT.2.1/ANDROID

The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following audit events:

- successful application of policies to a mobile device,
- [generating] periodic reachability events,
- [
  - change in enrollment state,
  - failure to install an application from the MAS Server,
  - failure to update an application from the MAS Server,
  - [detection of apps on a deny list, detection of apps not on an allow list, required apps missing, unapproved device manufacturer, unapproved operating system version].

##### FAU\_ALT\_EXT.2.2/ANDROID

The MDM Agent shall queue alerts if the trusted channel is not available.

6.3.1.3 [AGENTMOD] FAU\_ALT\_EXT.2/IOS Agent Alerts

FAU\_ALT\_EXT.2.1/IOS

The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following audit events:

- successful application of policies to a mobile device,
- [generating] periodic reachability events,
- [
  - change in enrollment state,
  - failure to install an application from the MAS Server,
  - failure to update an application from the MAS Server,
  - [detection of apps on a deny list, detection of apps not on an allow list, required apps missing, unapproved model, unapproved operating system version]].

FAU\_ALT\_EXT.2.2/IOS

The MDM Agent shall queue alerts if the trusted channel is not available.

6.3.1.4 [MDMPP] FAU\_GEN.1(1) Audit Data Generation

FAU\_GEN.1.1(1)<sup>1</sup>

The TSF shall [invoke platform-provided functionality, implement functionality] to generate an audit record of the following auditable events:

- a. All administrative actions
- b. [Commands issued to the MDM Agent]
- c. Specifically defined auditable events listed in **Table 13**
- d. [start up and shut down of the MDM system, [MDM Agent alerts (generated by FAU ALT\_EXT.2.1 in the MDM Agent PP-Module), MDM Agent audit records (generated by FAU\_GEN.1.1(2) in the MDM Agent PP-Module)]].

FAU\_GEN.1.2(1)

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

- date and time of the event
- type of event
- subject identity
- (if relevant) the outcome (success or failure) of the event
- additional information in **Table 13**
- [where the event occurred].

Table 13: Server Auditable Events

| Requirement | Auditable Events | Additional Audit Record Contents |
|-------------|------------------|----------------------------------|
|-------------|------------------|----------------------------------|

<sup>1</sup> TD0629

|  |   |  |
|--|---|--|
| <b>FAU_ALT_EXT.1<br/>(man)</b>         | Type of alert.  | Identity of Mobile Device that sent alert.             |
| <b>FAU_GEN.1(1)<br/>(man)</b>          | None.   | N/A  |
| <b>FAU_GEN.1(2)<br/>(sel)</b>          | None.   | N/A  |
| <b>FAU_NET_EXT.1<br/>(man)</b>         | None.   | N/A  |
| <b>FAU_SAR.1<br/>(opt)</b>             | None.   | N/A  |
| <b>FAU_STG_EXT.1<br/>(man)</b>         | None.   | N/A  |
| <b>FCO_CPC_EXT.1<br/>(obj)</b>         | Enabling/Disabling communications between a pair of components.       | Identities of the endpoints pairs enabled or disabled. |
| <b>FCS_CKM_EXT.4<br/>(man)</b>         | None.   | N/A  |
| <b>FCS_CKM.1<br/>(man)</b>             | [ <u>Failure of key generation activity for authentication keys</u> ] | No additional information.                             |
| <b>FCS_CKM.2<br/>(man)</b>             | None.   | N/A  |
| <b>FCS_COP.1(1)<br/>(man)</b>          | None.   | N/A  |
| <b>FCS_COP.1(2)<br/>(man)</b>          | None.   | N/A  |
| <b>FCS_COP.1(3)<br/>(man)</b>          | None.   | N/A  |
| <b>FCS_COP.1(4)<br/>(man)</b>          | None.   | N/A  |
| <b>FCS_RBG_EXT.1<br/>(man)</b>         | Failure of the randomization process.                                 | No additional information.                             |
| <b>FCS_STG_EXT.1<br/>(man)</b>         | None.   | N/A  |
| <b>FIA_ENR_EXT.1/ANDROID<br/>(man)</b> | Failure of MD user authentication.                                    | Presented username.                                    |
| <b>FIA_ENR_EXT.1/IOS<br/>(man)</b>     | Failure of MD user authentication.                                    | Presented username.                                    |
| <b>FIA_UAU.1<br/>(man)</b>             | None.   | N/A  |
| <b>FIA_X509_EXT.1(1)<br/>(man)</b>     | Failure to validate X.509 certificate.                                | Reason for failure.                                    |
| <b>FIA_X509_EXT.2<br/>(man)</b>        | Failure to establish connection to determine revocation status.       | No additional information.                             |
| <b>FIA_X509_EXT.5<br/>(man)</b>        | None.   | N/A  |



|   |  |  |
|---|--|--|
| <b>FMT_MOF.1(1)</b><br><b>(man)</b>         | Issuance of command to perform function.<br><br>Change of policy settings.         | Command sent and identity of MDM Agent recipient(s).<br><br>Policy changed and value or full policy. |
| <b>FMT_MOF.1(2)</b><br><b>(man)</b>         | Enrollment by a user.  | Identity of user.  |
| <b>FMT_MOF.1(3)</b><br><b>(sel)</b>         | None.  | N/A  |
| <b>FMT_POL_EXT.1</b><br><b>(man)</b>        | None.  | N/A  |
| <b>FMT_SMF.1(1)/ANDROID</b><br><b>(man)</b> | None.  | N/A  |
| <b>FMT_SMF.1(1)/IOS</b><br><b>(man)</b>     | None.  | N/A  |
| <b>FMT_SMF.1(2)/ANDROID</b><br><b>(man)</b> | Success or failure of function.  | No additional information.   |
| <b>FMT_SMF.1(2)/IOS</b><br><b>(man)</b>     | Success or failure of function.  | No additional information.   |
| <b>FMT_SMF.1(3)</b><br><b>(sel)</b>         | None.  | N/A  |
| <b>FMT_SMR.1(1)</b><br><b>(man)</b>         | None.  | N/A  |
| <b>FMT_SMR.1(2)</b><br><b>(sel)</b>         | None.  | N/A  |
| <b>FPT_API_EXT.1</b><br><b>(man)</b>        | None.  | N/A  |
| <b>FPT_ITT.1(2)</b><br><b>(sel)</b>         | Initiation and termination of the trusted channel.                                 | Trusted channel protocol. Identity of initiator and recipient.                                       |
| <b>FPT_LIB_EXT.1</b><br><b>(man)</b>        | None.  | N/A  |
| <b>FPT_TST_EXT.1</b><br><b>(man)</b>        | Initiation of self-test.<br>Failure of self-test.<br>Detected integrity violation. | Algorithm that caused failure.<br>The TSF code file that caused the integrity violation.             |
| <b>FPT_TUD_EXT.1</b><br><b>(man)</b>        | Success or failure of signature verification.                                      | No additional information.   |
| <b>FTA_TAB.1</b><br><b>(opt)</b>            | Change in banner setting.  | No additional information.   |
| <b>FTP_ITC.1(1)</b><br><b>(man)</b>         | Initiation and termination of the trusted channel.                                 | Trusted channel protocol. Non-TOE endpoint of connection.  |
| <b>FTP_ITC_EXT.1</b><br><b>(man)</b>        | None.  | N/A  |
| <b>FTP_TRP.1(1)</b><br><b>(man)</b>         | Initiation and termination of the trusted channel.                                 | Trusted channel protocol.<br>Identity of administrator.  |
| <b>FTP_TRP.1(2)</b><br><b>(man)</b>         | Initiation and termination of the trusted channel.                                 | Trusted channel protocol.  |

---

**6.3.1.5 [MDMPP] FAU\_GEN.1(2) Audit Generation (MAS Server)**


---

**FAU\_GEN.1.1(2)**

The MAS Server shall be able to generate an audit record of the following auditable events:

- a. Failure to push a new application on a managed mobile device
- b. Failure to update an existing application on a managed mobile device.

**FAU\_GEN.1.2(2)**

The [MAS Server] shall record within each TSF audit record at least the following information:

- date and time of the event
- type of event
- mobile device identity
- [no other information].

---

**6.3.1.6 [AGENTMOD] FAU\_GEN.1(2) Audit Data Generation**


---

**FAU\_GEN.1.1(2)<sup>2</sup>**

The MDM Agent shall [invoke platform-provided functionality, implement functionality] to generate an MDM Agent audit record of the following auditable events:

- a. Startup and shutdown of the MDM Agent;
- b. All auditable events for not specified level of audit; and
- c. MDM policy updated, any modification commanded by the MDM Server, specifically defined auditable events listed in **Table 14**, and [no other events].

**FAU\_GEN.1.2(2)**

The [TSF, TOE platform] shall record within each MDM Agent audit record at least the following information:

- a. Date and time of the event, type of event, subject identity, (if relevant) the outcome (success or failure) of the event, and additional information in **Table 14**; and
- b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-Module/ST, [no other information].

**Table 14: Agent Auditable Events**

| Requirement                  | Auditable Events                  | Additional Audit Record Contents |
|------------------------------|-----------------------------------|----------------------------------|
| <b>FAU_ALT_EXT.2/ANDROID</b> | Success/failure of sending alert. | No additional information.       |
| <b>FAU_ALT_EXT.2/IOS</b>     | Success/failure of sending alert. | No additional information.       |
| <b>FAU_GEN.1(2)</b>          | None.                             | N/A                              |

---

<sup>2</sup> TD0660

|                         |   |                                     |
|-------------------------|---|-------------------------------------|
| <b>FAU_SEL.1(2)</b>     | All modifications to the audit configuration that occur while the audit collection functions are operating. | No additional information.          |
| <b>FCS_STG_EXT.1(2)</b> | None.   | N/A                                 |
| <b>FIA_ENR_EXT.2</b>    | Enrollment in management.   | Reference identifier of MDM Server. |
| <b>FMT_POL_EXT.2</b>    | Failure of policy validation.   | Reason for failure of validation.   |
| <b>FMT_SMF_EXT.4</b>    | Outcome (Success/failure) of function.  | No additional information.          |
| <b>FMT_UNR_EXT.1.1</b>  | [None]  | No additional information.          |

---

**6.3.1.7 [MDMPP] FAU\_NET\_EXT.1 Network Reachability Review**

---

**FAU\_NET\_EXT.1.1**

The TSF shall provide authorized administrators with the capability to read the network connectivity status of an enrolled agent.

---

**6.3.1.8 [MDMPP] FAU\_SAR.1 Audit Review**

---

**FAU\_SAR.1.1**

The TSF shall [invoke platform-provided functionality, implement functionality] to provide Authorized Administrators with the capability to read all audit data from the audit records.

**FAU\_SAR.1.2**

The TSF shall [invoke platform-provided functionality, implement functionality] to provide the audit records in a manner suitable for the Authorized Administrators to interpret the information.

---

**6.3.1.9 [AGENTMOD] FAU\_SEL.1(2) Security Audit Event Selection**

---

**FAU\_SEL.1.1(2)**

The TSF shall [invoke platform-provided functionality, implement functionality] to select the set of events to be audited from the set of all auditable events based on the following attributes:

- a. event type
- b. success of auditable security events, failure of auditable security events, [*no other attributes*].

---

**6.3.1.10 [MDMPP] FAU\_STG\_EXT.1 External Trail Storage**

---

**FAU\_STG\_EXT.1.1**

The TSF shall be able to use a trusted channel per FTP\_ITC.1(1) to transmit audit data to an external IT entity and [no other method].

## 6.3.2 Class FCO: Communication

---

### 6.3.2.1 [MDMPP] FCO\_CPC\_EXT.1 *Component Registration Channel Definition*

---

#### FCO\_CPC\_EXT.1.1

The TSF shall [implement functionality] to require an Administrator to enable communications between any pair of TOE components before such communication can take place.

#### FCO\_CPC\_EXT.1.2<sup>3</sup>

The TSF shall [invoke platform-provided functionality] to implement a registration process in which components establish and use a communications channel that uses [

- A channel that meets the secure registration channel requirements in [FTP\_TRP.1(2)]

] for at least TSF data.

#### FCO\_CPC\_EXT.1.3

The TSF shall [implement functionality] to enable an administrator to disable communications between any pair of TOE components.

## 6.3.3 Class FCS: Cryptographic Support

---

### 6.3.3.1 [MDMPP] FCS\_CKM.1 *Cryptographic Key Generation*

---

#### FCS\_CKM.1.1

The TSF shall [invoke platform-provided functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meets FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;
- 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].

---

### 6.3.3.2 [MDMPP] FCS\_CKM.2 *Cryptographic Key Establishment*

---

#### FCS\_CKM.2.1

The TSF shall [invoke platform-provided functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1\_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1",

---

<sup>3</sup> TD0462

- 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”].

---

**6.3.3.3 [MDMPP] FCS\_CKM\_EXT.4*****Cryptographic Key Destruction***

---

**FCS\_CKM\_EXT.4.1**

The TSF shall destroy plaintext keying material and critical security parameters by: [

- invoking platform-provided functionality with the following rules:
  - For volatile memory, the destruction shall be executed by [
    - a single direct overwrite consisting of [zeroes]]
  - For non-volatile memory that consists of the invocation of an interface provided by the underlying platform that [
    - logically addresses the storage location of the key and performs a [single] direct overwrite consisting of [ones]].

**FCS\_CKM\_EXT.4.2**

The TSF shall destroy all plaintext keying material and critical security parameters (CSPs) when no longer needed.

---

**6.3.3.4 [MDMPP] FCS\_COP.1(1)*****Cryptographic Operation (Confidentiality Algorithms)***

---

**FCS\_COP.1.1(1)**

The TSF shall [invoke platform-provided functionality] to perform encryption/decryption in accordance with a specified cryptographic algorithm [

- AES-GCM (as defined in NIST SP 800-38D)

and cryptographic key sizes [128-bit, 256-bit].

---

**6.3.3.5 [MDMPP] FCS\_COP.1(2)*****Cryptographic Operation (Hashing Algorithms)***

---

**FCS\_COP.1.1(2)**

The TSF shall [invoke platform-provided functionality, implement functionality] to perform cryptographic hashing 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.

---

**6.3.3.6 [MDMPP] FCS\_COP.1(3)*****Cryptographic Operation (Signature Algorithms)***

---

**FCS\_COP.1.1(3)**

The TSF shall [invoke platform-provided functionality, implement functionality] to perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4.
- ECDSA schemes using “NIST curves” P-384 and [P-256, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5].

---

**6.3.3.7 [MDMPP] FCS\_COP.1(4)**
***Cryptographic Operation (Keyed-Hash Message Authentication)***


---

**FCS\_COP.1.1(4)**

The TSF shall [invoke platform-provided functionality] to perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[SHA-256, SHA-384], key sizes [160 bits], and message digest sizes [256, 384] bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, “Secure Hash Standard.”

---

**6.3.3.8 [MDMPP] FCS\_RBG\_EXT.1**
***Extended: Random Bit Generation***


---

**FCS\_RBG\_EXT.1.1**

The TSF shall [invoke platform-provided functionality] to perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [CTR\_DRBG (AES)].

**FCS\_RBG\_EXT.1.2**

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

---

**6.3.3.9 [MDMPP] FCS\_STG\_EXT.1**
***Cryptographic Key Storage***


---

**FCS\_STG\_EXT.1.1**

The TSF shall utilize [platform-provided key storage] for all persistent secrets and private keys.

---

**6.3.3.10 [AGENTMOD] FCS\_STG\_EXT.1(2) Cryptographic Key Storage**


---

**FCS\_STG\_EXT.1.1(2)**

The MDM Agent shall use the platform-provided key storage for all persistent secret and private keys.

**6.3.4 Class FIA: Identification and Authentication**


---

**6.3.4.1 [MDMPP] FIA\_ENR\_EXT.1/ANDROID**
***Enrollment of Mobile Device into Management (Android)***


---

**FIA\_ENR\_EXT.1.1/ANDROID**

The TSF shall authenticate the remote users over a trusted channel during the enrollment of a mobile device.

**FIA\_ENR\_EXT.1.2/ANDROID**

The TSF shall limit the user's enrollment of devices to devices specified by [IMEI] and [specific device models, a number of devices, [serial number, manufacturer, operating system]].

---

**6.3.4.2 [MDMPP] FIA\_ENR\_EXT.1/IOS Enrollment of Mobile Device into Management (iOS)**

---

**FIA\_ENR\_EXT.1.1/IOS**

The TSF shall authenticate the remote users over a trusted channel during the enrollment of a mobile device.

**FIA\_ENR\_EXT.1.2/IOS**

The TSF shall limit the user's enrollment of devices to devices specified by [DEP identifier] and [no other features].

---

**6.3.4.3 [AGENTMOD] FIA\_ENR\_EXT.2 Agent Enrollment of Mobile Device into Management**

---

**FIA\_ENR\_EXT.2.1**

The MDM Agent shall record the reference identifier of the MDM Server during the enrollment process.

---

**6.3.4.4 [MDMPP] FIA\_UAU.1 Timing of Authentication**

---

**FIA\_UAU.1.1**

The TSF shall [implement functionality] to allow [view login banner, view MDM software version, request password reset, select language] on behalf of the user to be performed before the user is authenticated with the Server.

**FIA\_UAU.1.2**

The TSF shall [implement functionality] that requires each user to be successfully authenticated with the Server before allowing any other TSF-mediated actions on behalf of that user.

---

**6.3.4.5 [MDMPP] FIA\_X509\_EXT.1(1) Validation of Certificates**

---

**FIA\_X509\_EXT.1.1(1)<sup>4</sup>**

The TSF shall [invoke platform-provided functionality, 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 TSF 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.

---

<sup>4</sup> TD0641

- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The TSF shall validate the revocation status of the certificate using [OCSP as specified in RFC 6960, an OCSP TLS Status Request Extension (i.e., OCSP stapling) as specified in RFC 6066].
- The TSF 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 ECU 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 ECU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the ECU field.

**FIA\_X509\_EXT.1.2(1)**

The TSF shall [invoke platform-provided functionality, implement functionality] to treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

**6.3.4.6 [MDMPP] FIA\_X509\_EXT.2****X.509 Certificate Authentication****FIA\_X509\_EXT.2.1**

The TSF shall [

- invoke platform-provided functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [HTTPS, TLS], and [
  - code signing for system software updates,
  - code signing for integrity verification,
  - policy signing],
- implement functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [
  - no protocols]and [
  - policy signing].]

**FIA\_X509\_EXT.2.2**

When the [TSF, TOE platform] cannot establish a connection to determine the validity of a certificate, the TSF shall [invoke platform-provided functionality, implement functionality] to [accept the certificate, not accept the certificate].



---

**6.3.4.7 [MDMPP] FIA\_X509\_EXT.5 X.509 Unique Certificate**

---

**FIA\_X509\_EXT.5.1**

The TSF shall [invoke platform-provided functionality] to require a unique certificate for each client device.

**6.3.5 Class FMT: Security Management**

---

**6.3.5.1 [MDMPP] FMT\_MOF.1(1) Management of Functions Behavior**

---

**FMT\_MOF.1.1(1)**

The TSF shall restrict the ability to perform the functions

- listed in FMT\_SMF.1(1)
- enable, disable, and modify policies listed in FMT\_SMF.1(1)
- listed in FMT\_SMF.1(2)
- [enable, disable and modify policies listed in FMT\_SMF.1(3)]

to authorized administrators.

---

**6.3.5.2 [MDMPP] FMT\_MOF.1(2) Management of Functions Behavior (Enrollment)**

---

**FMT\_MOF.1.1(2)**

The MDM Server shall restrict the ability to initiate the enrollment process to authorized administrators and MD users.

---

**6.3.5.3 [MDMPP] FMT\_MOF.1(3) Management of Functions in (MAS Server Downloads)**

---

**FMT\_MOF.1.1(3)**

The MAS Server shall restrict the ability to download applications, allowing only enrolled mobile devices that are compliant with MDM policies and assigned to a user in the application access group to perform this function.

---

**6.3.5.4 [MDMPP] FMT\_POL\_EXT.1 Trusted Policy Update**

---

**FMT\_POL\_EXT.1.1**

The TSF shall provide digitally signed policies and policy updates to the MDM Agent.

---

**6.3.5.5 [AGENTMOD] FMT\_POL\_EXT.2 Agent Trusted Policy Update**

---

**FMT\_POL\_EXT.2.1**

The MDM Agent shall only accept policies and policy updates that are digitally signed by a certificate that has been authorized for policy updates by the MDM Server.

**FMT\_POL\_EXT.2.2**

The MDM Agent shall not install policies if the policy-signing certificate is deemed invalid.

---

6.3.5.6 [MDMPP] FMT\_SMF.1(1)/ANDROID *Specification of Management Functions (Server configuration of Agent) (Android)*

---

**FMT\_SMF.1(1)/ANDROID**

The MDM Server shall be capable of communicating the following commands to the MDM Agent:

1. transition to the locked state (MDF Function 6)
2. full wipe of protected data (MDF Function 7)
3. unenroll from management
4. install policies
5. query connectivity status
6. query the current version of the MD firmware/software
7. query the current version of the hardware model of the device
8. query the current version of installed mobile applications
9. import X.509v3 certificates into the Trust Anchor Database (MDF Function 11)
10. install applications (MDF Function 16)
11. update system software (MDF Function 15)
12. remove applications (MDF Function 14)

and the following commands to the MDM Agent: [

- 13. remove Enterprise applications (MDF Function 17).
- 14. wipe Enterprise data (MDF Function 28).
- 15. remove imported X.509v3 certificates and [
  - no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12).
- 16. alert the user.
- 20. retrieve MD-software integrity verification values (MDF Function 38)]

and the following MD configuration policies:

25. password policy:
  - a. minimum password length
  - b. minimum password complexity
  - c. maximum password lifetime (MDF Function 1)
26. session locking policy:
  - a. screen-lock enabled/disabled
  - b. screen lock timeout
  - c. number of authentication failures (MDF Function 2)
27. wireless networks (SSIDs) to which the MD may connect (MDF Function 2)
28. security policy for each wireless network:
  - a. [
    - specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)]
  - b. ability to specify security type

- c. ability to specify authentication protocol
  - d. specify the client credentials to be used for authentication
  - e. *[no other WLAN management functions]* (WLAN Client Function 1)
29. application installation policy by [
- o specifying authorized application repository(s),
  - o denying application installation], (MDF Function 8)
30. enable/disable policy for *[camera, screen capture]* across device, [
- o no other method], (MDF Function 5)

and the following MD configuration policies: [

- 31. enable/disable policy for the VPN protection across MD, [
  - o on a per-app basis] (MDF Function 3),
- 32. enable/disable policy for *[Bluetooth]*, (MDF Function 4),
- 34. enable/disable policy for *[Wi-Fi tethering, USB tethering, Bluetooth tethering]*, (MDF Function 25),
- 38. enable/disable policy for local authentication bypass (MDF Function 27),
- 40. enable/disable policy for display notification in the locked state of [
  - o email notifications,
  - o calendar appointments,
  - o contact associated with phone call notification,
  - o text message notification,
  - o other application-based notifications](MDF Function 19),
- 47. the unlock banner policy, (MDF Function 36),
- 49. enable/disable [
  - o USB mass storage mode] (MDF Function 39),
- 50. enable/disable backup of [
  - o all applications,
  - o selected applications,
  - o selected groups of applications,
  - o configuration data] to [locally connected system, remote system] (MDF Function 40),
- 51. enable/disable [
  - o Hotspot functionality authenticated by [no authentication],
  - o USB tethering authenticated by [no authentication]] (MDF Function 41),
- 52. enable/disable location services: [
  - o across device] (MDF Function 22),
- 53. enable/disable policy for user unenrollment,
- 54. enable/disable policy for the Always-On VPN protection across device (MDF Function 45),
- 55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23),
- 58. enable/disable automatic updates of system software,
- 60. *[application installation policy by [*

- specifying a set of allowed applications (an application allow list)]]

---

6.3.5.7 [MDMPP] FMT\_SMF.1(1)/IOS *Specification of Management Functions (Server configuration of Agent) (iOS)*

---

### FMT\_SMF.1(1)/IOS

The MDM Server shall be capable of communicating the following commands to the MDM Agent:

1. transition to the locked state (MDF Function 6)
2. full wipe of protected data (MDF Function 7)
3. unenroll from management
4. install policies
5. query connectivity status
6. query the current version of the MD firmware/software
7. query the current version of the hardware model of the device
8. query the current version of installed mobile applications
9. import X.509v3 certificates into the Trust Anchor Database (MDF Function 11)
10. install applications (MDF Function 16)
11. update system software (MDF Function 15)
12. remove applications (MDF Function 14)

and the following commands to the MDM Agent: [

- 13. remove Enterprise applications (MDF Function 17).
- 14. wipe Enterprise data (MDF Function 28).
- 15. remove imported X.509v3 certificates and [
  - no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12).
- 16. alert the user.
- 23. revoke Biometric template]

and the following MD configuration policies:

25. password policy:
  - a. minimum password length
  - b. minimum password complexity
  - c. maximum password lifetime (MDF Function 1)
26. session locking policy:
  - a. screen-lock enabled/disabled
  - b. screen lock timeout
  - c. number of authentication failures (MDF Function 2)
27. wireless networks (SSIDs) to which the MD may connect (MDF Function 2)
28. security policy for each wireless network:
  - a. [
    - specify the CA(s) from which the MD will accept WLAN authentication server certificate(s).

- specify the FQDN(s) of acceptable WLAN authentication server certificate(s)]
  - b. ability to specify security type
  - c. ability to specify authentication protocol
  - d. specify the client credentials to be used for authentication
  - e. *[no other WLAN management functions]* (WLAN Client Function 1)
- 29. application installation policy by [
  - specifying authorized application repository(s), (MDF Function 8)
- 30. enable/disable policy for *[camera, screen capture]* across device, [
  - and no other method], (MDF Function 5)

and the following MD configuration policies: [

- 31. enable/disable policy for the VPN protection across MD
  - [on a per-app basis,
  - no other method] (MDF Function 3),
- 36. enable policy for data-at rest protection, (MDF Function 20),
- 38. enable/disable policy for local authentication bypass (MDF Function 27),
- 39. the Bluetooth trusted channel policy:
  - a. enable/disable the Discoverable mode (for BR/EDR)
  - b. change the Bluetooth device name, [
    - no other Bluetooth configuration] (MDF Function 18),
- 40. enable/disable policy for display notification in the locked state of [
  - email notifications,
  - calendar appointments,
  - contact associated with phone call notification,
  - text message notification,
  - other application-based notifications](MDF Function 19),
- 47. the unlock banner policy, (MDF Function 36),
- 49. enable/disable [
  - USB data transfer without user authentication] (MDF Function 39),
- 50. enable/disable backup of [
  - all applications,
  - configuration data] to [remote system] (MDF Function 40),
- 55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23),
- 60. [application installation policy by [
  - specifying a set of allowed applications (an application allow list)]]
- 61. [iOS Hub agent passcode authentication policy:
  - Minimum Passcode Length]

---

6.3.5.8 [MDMPP] FMT\_SMF.1(2)/ANDROID *Specification of Management Functions (Server Configuration of Server) (Android)*

---

**FMT\_SMF.1.1(2)/ANDROID**

The TSF shall be capable of performing the following management functions:

- a. choose X.509v3 certificates for MDM Server use
- b. configure the [
  - devices specified by [IMEI, [serial number]],
  - specific device models,
  - a number of devices
 ] and [[manufacturer, operating system]] allowed for enrollment,
- c. [
  2. configure the TOE unlock banner,
  3. configure periodicity of the following commands to the agent: [query connectivity status, query the current version of the MD firmware/software, query the current version of the hardware model of the device, query the current version of installed mobile applications],
  4. configure the privacy-sensitive information that will and will not be collected from particular mobile devices]
  6. configure the interaction between TOE components
  8. [configure server administrator login session timeout, configure Enterprise certificate to be used for signing policies, configure MDM Agent/platform to perform a network reachability test, configure transfer of MDM server logs to another server for storage, analysis, and reporting]].

---

6.3.5.9 [MDMPP] FMT\_SMF.1(2)/IOS *Specification of Management Functions (Server Configuration of Server) (iOS)*

---

**FMT\_SMF.1.1(2)/IOS**

The TSF shall be capable of performing the following management functions:

- a. choose X.509v3 certificates for MDM Server use
- b. configure the [
  - devices specified by [[DEP identifier]],
 ] and [no other features] allowed for enrollment
- c. [
  2. configure the TOE unlock banner,
  3. configure periodicity of the following commands to the agent: [query connectivity status, query the current version of the MD firmware/software, query the current version of the hardware model of the device, query the current version of installed mobile applications],
  4. configure the privacy-sensitive information that will and will not be collected from particular mobile devices
  6. configure the interaction between TOE components

8. [configure server administrator login session timeout, configure Enterprise certificate to be used for signing policies, configure MDM Agent/platform to perform a network reachability test, configure transfer of MDM server logs to another server for storage, analysis, and reporting].

---

#### 6.3.5.10 [MDMPP] FMT\_SMF.1(3) *Specification of Management Functions (MAS Server)*

---

##### FMT\_SMF.1.1(3)

The MAS Server shall be capable of performing the following management functions:

- a. Configure application access groups
- b. Download applications
- c. [no other functions]

---

#### 6.3.5.11 [AGENTMOD] FMT\_SMF\_EXT.4 *Specification of Management Functions*

---

##### FMT\_SMF\_EXT.4.1

The MDM Agent shall be capable of interacting with the platform to perform the following functions:

- Import the certificates to be used for authentication of MDM Agent communications,
- [administrator-provided device management functions in MDM PP]
- [no additional functions].

##### FMT\_SMF\_EXT.4.2

The MDM Agent shall be capable of performing the following functions:

- Enroll in management
- Configure whether users can unenroll from management
- [configure periodicity of reachability events].

---

#### 6.3.5.12 [MDMPP] FMT\_SMR.1(1) *Security Management Roles*

---

##### FMT\_SMR.1.1(1)

The TSF shall maintain the roles administrator, MD user, and [[server primary administrator, security configuration administrator, device user group administrator, auditor]].

##### FMT\_SMR.1.2(1)

The TSF shall be able to associate users with roles.

---

#### 6.3.5.13 [MDMPP] FMT\_SMR.1(2) *Security Management Roles (MAS Server)*

---

##### FMT\_SMR.1.1(2)

The TSF shall additionally maintain the roles enrolled mobile devices, application access groups, and [server primary administrator, security configuration administrator, device user group administrator, auditor].

##### FMT\_SMR.1.2(2)

The MAS Server shall be able to associate users with roles.

---

**6.3.5.14 [AGENTMOD] FMT\_UNR\_EXT.1 User Unenrollment Prevention**

---

**FMT\_UNR\_EXT.1.1**

The MDM Agent shall provide a mechanism to enforce the following behavior upon an attempt to unenroll the mobile device from management: [prevent the unenrollment from occurring].

**6.3.6 Class FPT: Protection of the TSF**

---

**6.3.6.1 [MDMPP] FPT\_API\_EXT.1 Use of Supported Services and APIs**

---

**FPT\_API\_EXT.1.1**

The TSF shall use only documented platform API's.

---

**6.3.6.2 [MDMPP] FPT\_ITT.1(2) Internal TOE TSF Data Transfer (To MDM Agent)**

---

**FPT\_ITT.1.1(2)**

The TSF shall [

- invoke platform-provided functionality to use [
  - mutually authenticated TLS,
  - HTTPS]

] to protect all data from disclosure and modification when it is transferred between TSF and MDM Agent.

---

**6.3.6.3 [MDMPP] FPT\_LIB\_EXT.1 Use of Third Party Libraries**

---

**FPT\_LIB\_EXT.1.1**

The MDM software shall be packaged with only *[a list of third-party libraries in Appendix A]*.

---

**6.3.6.4 [MDMPP] FPT\_TST\_EXT.1 Functionality Testing**

---

**FPT\_TST\_EXT.1.1**

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

**FPT\_TST\_EXT.1.2**

The TSF shall [invoke platform-provided functionality] to provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [TOE platform]-provided cryptographic services.



---

**6.3.6.5 [MDMPP] FPT\_TUD\_EXT.1 Trusted Update**

---

**FPT\_TUD\_EXT.1.1<sup>5</sup>**

The TSF shall provide Authorized Administrators the ability to query the current version of the software.

**FPT\_TUD\_EXT.1.2**

The TSF shall [invoke platform-provided functionality, implement functionality] to provide Authorized Administrators the ability to initiate updates to TSF software.

**FPT\_TUD\_EXT.1.3**

The TSF shall [invoke platform-provided functionality] to provide a means to verify software updates to the TSF using a digital signature mechanism prior to installing those updates.

**6.3.7 Class FTA: TOE Access**

---

**6.3.7.1 [MDMPP] FTA\_TAB.1 Default TOE Access Banners**

---

**FTA\_TAB.1.1**

Before establishing a user session, the TSF shall [implement functionality] to display an Administrator-specified advisory notice and consent warning message regarding use of the TOE.

**6.3.8 Class FTP: Trusted Path/Channels**

---

**6.3.8.1 [MDMPP] FTP\_ITC\_EXT.1 Trusted Channel**

---

**FTP\_ITC\_EXT.1.1**

The TSF shall provide a communication channel between itself and [

- an MDM Agent that is internal to the TOE

] that is logically distinct from other communication channels, as specified in [FPT\_ITT.1(2)].

---

**6.3.8.2 [MDMPP] FTP\_ITC.1(1) Inter-TSF Trusted Channel (Authorized IT Entities)**

---

**FTP\_ITC.1.1(1)**

The TSF shall [

- invoke platform-provided functionality to use [
  - mutually authenticated TLS

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [authentication server] that is logically distinct from other

---

<sup>5</sup> TD0438

communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

**FTP\_ITC.1.2(1)**

The TSF shall [invoke platform-provided functionality] to permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

**FTP\_ITC.1.3(1)**

The TSF shall [invoke platform-provided functionality] to initiate communication via the trusted channel for [*sending audit data to the Syslog Server, sending authentication requests to LDAP*].

---

**6.3.8.3 [MDMPP] FTP\_TRP.1(1)*****Trusted Path (for Remote Administration)***

---

**FTP\_TRP.1.1(1)**

The TSF shall [

- invoke platform-provided functionality to use [
  - TLS,
  - HTTPS]

] to provide a trusted communication path between itself as a [server] and remote administrators that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification, disclosure.

**FTP\_TRP.1.2(1)**

The TSF shall [invoke platform-provided functionality] to permit remote administrators to initiate communication via the trusted path.

**FTP\_TRP.1.3(1)**

The TSF shall [invoke platform-provided functionality] to require the use of the trusted path for all remote administration actions.

---

**6.3.8.4 [MDMPP] FTP\_TRP.1(2)*****Trusted Path (for Enrollment)***

---

**FTP\_TRP.1.1(2)**

The TSF shall [

- invoke platform-provided functionality to use [
  - TLS,
  - HTTPS]

] to provide a trusted communication path between itself (as a server) and MD users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure and detection of modification of the communicated data from modification, disclosure.

**FTP\_TRP.1.2(2)**

The TSF shall [invoke platform-provided functionality] to permit MD users to initiate communication via the trusted path.

**FTP\_TRP.1.3(2)**

The TSF shall [invoke platform-provided functionality] to require the use of the trusted path for all MD user actions.

**6.4 Statement of Security Functional Requirements Consistency**

The Security Functional Requirements included in the ST represent all required SFRs specified in the PPs against which exact conformance is claimed and a subset of the selection-based, optional, and objective SFRs. All hierarchical relationships, dependencies, and unfulfilled dependency rationales in the ST are considered to be identical to those that are defined in the claimed PP.

## 7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The SARs which are claimed are in exact conformance with the [MDMPP] and [AGENTMOD].

### 7.1 Class ASE: Security Target evaluation

#### 7.1.1 ST introduction (ASE\_INT.1)

---

##### 7.1.1.1 *Developer action elements:*

---

###### ASE\_INT.1.1D

The developer shall provide an ST introduction.

---

##### 7.1.1.2 *Content and presentation elements:*

---

###### ASE\_INT.1.1C

The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

###### ASE\_INT.1.2C

The ST reference shall uniquely identify the ST.

###### ASE\_INT.1.3C

The TOE reference shall uniquely identify the TOE.

###### ASE\_INT.1.4C

The TOE overview shall summarise the usage and major security features of the TOE.

###### ASE\_INT.1.5C

The TOE overview shall identify the TOE type.

###### ASE\_INT.1.6C

The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

###### ASE\_INT.1.7C

The TOE description shall describe the physical scope of the TOE.

###### ASE\_INT.1.8C

The TOE description shall describe the logical scope of the TOE.

---

##### 7.1.1.3 *Evaluator action elements:*

---

###### ASE\_INT.1.1E

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

**ASE\_INT.1.2E**

The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

**7.1.2 Conformance claims (ASE\_CCL.1)**

---

**7.1.2.1 Developer action elements:**

---

**ASE\_CCL.1.1D**

The developer shall provide a conformance claim.

**ASE\_CCL.1.2D**

The developer shall provide a conformance claim rationale

---

**7.1.2.2 Content and presentation elements:**

---

**ASE\_CCL.1.1C**

The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

**ASE\_CCL.1.2C**

The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

**ASE\_CCL.1.3C**

The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

**ASE\_CCL.1.4C**

The CC conformance claim shall be consistent with the extended components definition.

**ASE\_CCL.1.5C**

The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

**ASE\_CCL.1.6C**

The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

**ASE\_CCL.1.7C**

The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

**ASE\_CCL.1.8C**

The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

**ASE\_CCL.1.9C**

The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

**ASE\_CCL.1.10C**

The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

---

**7.1.2.3 Evaluator action elements:**

---

**ASE\_CCL.1.1E**

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

**7.1.3 Security objectives for the operational environment (ASE\_OBJ.1)**

---

**7.1.3.1 Developer action elements:**

---

**ASE\_OBJ.1.1D**

The developer shall provide a statement of security objectives.

---

**7.1.3.2 Content and presentation elements:**

---

**ASE\_OBJ.1.1C**

The statement of security objectives shall describe the security objectives for the operational environment.

---

**7.1.3.3 Evaluator action elements:**

---

**ASE\_OBJ.1.1E**

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

**7.1.4 Extended components definition (ASE\_ECD.1)**

---

**7.1.4.1 Developer action elements:**

---

**ASE\_ECD.1.1D**

The developer shall provide a statement of security requirements.

**ASE\_ECD.1.2D**

The developer shall provide an extended components definition.

---

**7.1.4.2 *Content and presentation elements:***

---

**ASE\_ECD.1.1C**

The statement of security requirements shall identify all extended security requirements.

**ASE\_ECD.1.2C**

The extended components definition shall define an extended component for each extended security requirement.

**ASE\_ECD.1.3C**

The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

**ASE\_ECD.1.4C**

The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

**ASE\_ECD.1.5C**

The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

---

**7.1.4.3 *Evaluator action elements:***

---

**ASE\_ECD.1.1E**

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

**ASE\_ECD.1.2E**

The evaluator shall confirm that no extended component can be clearly expressed using existing components.

**7.1.5 Stated security requirements (ASE\_REQ.1)**

---

**7.1.5.1 *Developer action elements:***

---

**ASE\_REQ.1.1D**

The developer shall provide a statement of security requirements.

**ASE\_REQ.1.2D**

The developer shall provide a security requirements rationale.

---

**7.1.5.2 *Content and presentation elements:***

---

**ASE\_REQ.1.1C**

The statement of security requirements shall describe the SFRs and the SARs.

**ASE\_REQ.1.2C**

All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

**ASE\_REQ.1.3C**

The statement of security requirements shall identify all operations on the security requirements.

**ASE\_REQ.1.4C**

All operations shall be performed correctly.

**ASE\_REQ.1.5C**

Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

**ASE\_REQ.1.6C**

The statement of security requirements shall be internally consistent.

---

**7.1.5.3 Evaluator action elements:**

---

**ASE\_REQ.1.1E**

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

**7.1.6 TOE summary specification (ASE\_TSS.1)**

---

**7.1.6.1 Developer action elements:**

---

**ASE\_TSS.1.1D**

The developer shall provide a TOE summary specification.

---

**7.1.6.2 Content and presentation elements:**

---

**ASE\_TSS.1.1C**

The TOE summary specification shall describe how the TOE meets each SFR.

---

**7.1.6.3 Evaluator action elements:**

---

**ASE\_TSS.1.1E**

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

**ASE\_TSS.1.2E**



The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

## **7.2 Class ADV: Development**

### **7.2.1 Basic Functional Specification (ADV\_FSP.1)**

---

#### **7.2.1.1 Developer action elements:**

---

##### **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.

---

#### **7.2.1.2 Content and presentation elements:**

---

##### **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.

---

#### **7.2.1.3 Evaluator action elements:**

---

##### **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.

## 7.3 Class AGD: Guidance Documentation

### 7.3.1 Operational User Guidance (AGD\_OPE.1)

---

#### 7.3.1.1 *Developer action elements:*

---

##### **AGD\_OPE.1.1D**

The developer shall provide operational user guidance.

---

#### 7.3.1.2 *Content and presentation elements:*

---

##### **AGD\_OPE.1.1C**

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

##### **AGD\_OPE.1.2C**

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

##### **AGD\_OPE.1.3C**

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

##### **AGD\_OPE.1.4C**

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

##### **AGD\_OPE.1.5C**

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

##### **AGD\_OPE.1.6C**

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

##### **AGD\_OPE.1.7C**

The operational user guidance shall be clear and reasonable.

---

**7.3.1.3 Evaluator action elements:**

---

**AGD\_OPE.1.1E**

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

**7.3.2 Preparative Procedures (AGD\_PRE.1)**

---

**7.3.2.1 Developer action elements:**

---

**AGD\_PRE.1.1D**

The developer shall provide the TOE including its preparative procedures.

---

**7.3.2.2 Content and presentation elements:**

---

**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.

---

**7.3.2.3 Evaluator action elements:**

---

**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.

**7.4 Class ALC: Life Cycle Support**

**7.4.1 Labeling of the TOE (ALC\_CMC.1)**

---

**7.4.1.1 Developer action elements:**

---

**ALC\_CMC.1.1D**

The developer shall provide the TOE and a reference for the TOE.

---

7.4.1.2 *Content and presentation elements:*

---

**ALC\_CMC.1.1C**

The TOE shall be labeled with its unique reference.

---

7.4.1.3 *Evaluator action elements:*

---

**ALC\_CMC.1.1E**

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

**7.4.2 TOE CM Coverage (ALC\_CMS.1)**

---

7.4.2.1 *Developer action elements:*

---

**ALC\_CMS.1.1D**

The developer shall provide a configuration list for the TOE.

---

7.4.2.2 *Content and presentation elements:*

---

**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.

---

7.4.2.3 *Evaluator action elements:*

---

**ALC\_CMS.1.1E**

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

**7.5 Class ATE: Tests**

**7.5.1 Independent Testing - Conformance (ATE\_IND.1)**

---

7.5.1.1 *Developer action elements:*

---

**ATE\_IND.1.1D**

The developer shall provide the TOE for testing.

---

7.5.1.2 *Content and presentation elements:*

---

**ATE\_IND.1.1C**

The TOE shall be suitable for testing.

---

7.5.1.3 *Evaluator action elements:*

---

**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.

## **7.6 Class AVA: Vulnerability Assessment**

### **7.6.1 Vulnerability Survey (AVA\_VAN.1)**

---

7.6.1.1 *Developer action elements:*

---

**AVA\_VAN.1.1D**

The developer shall provide the TOE for testing.

---

7.6.1.2 *Content and presentation elements:*

---

**AVA\_VAN.1.1C**

The TOE shall be suitable for testing.

---

7.6.1.3 *Evaluator action elements:*

---

**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.

## 8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Security Audit, Communication, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access, and Trusted Path/Channels. The following table defines which distributed TOE component(s) perform the capabilities described by the SFR.

Table 15: SFR and TOE Component Mapping

| Requirement                      | UEM Server | Android Hub Agent | iOS Hub Agent |
|----------------------------------|------------|-------------------|---------------|
| [MDMPP] FAU_ALT_EXT.1            | X          |                   |               |
| [AGENTMOD] FAU_ALT_EXT.2/ANDROID |            | X                 |               |
| [AGENTMOD] FAU_ALT_EXT.2/IOS     |            |                   | X             |
| [MDMPP] FAU_GEN.1(1)             | X          | X                 | X             |
| [MDMPP] FAU_GEN.1(2)             | X          |                   |               |
| [AGENTMOD] FAU_GEN.1(2)          |            | X                 | X             |
| [MDMPP] FAU_NET_EXT.1            | X          |                   |               |
| [MDMPP] FAU_SAR.1                | X          |                   |               |
| [AGENTMOD] FAU_SEL.1(2)          |            | X                 | X             |
| [MDMPP] FAU_STG_EXT.1            | X          | X                 | X             |
| [MDMPP] FCO_CPC_EXT.1            | X          | X                 | X             |
| [MDMPP] FCS_CKM.1                | X          | X                 | X             |
| [MDMPP] FCS_CKM.2                | X          | X                 | X             |
| [MDMPP] FCS_CKM_EXT.4            | X          | X                 | X             |
| [MDMPP] FCS_COP.1(1)             | X          | X                 | X             |
| [MDMPP] FCS_COP.1(2)             | X          | X                 | X             |
| [MDMPP] FCS_COP.1(3)             | X          | X                 | X             |
| [MDMPP] FCS_COP.1(4)             | X          | X                 | X             |
| [MDMPP] FCS_RBG_EXT.1            | X          | X                 | X             |
| [MDMPP] FCS_STG_EXT.1            | X          | X                 | X             |
| [AGENTMOD] FCS_STG_EXT.1(2)      |            | X                 | X             |
| [MDMPP] FIA_ENR_EXT.1/ANDROID    | X          |                   |               |
| [MDMPP] FIA_ENR_EXT.1/IOS        | X          |                   |               |
| [AGENTMOD] FIA_ENR_EXT.2         |            | X                 | X             |
| [MDMPP] FIA_UAU.1                | X          |                   |               |
| [MDMPP] FIA_X509_EXT.1(1)        | X          | X                 | X             |
| [MDMPP] FIA_X509_EXT.2           | X          | X                 | X             |
| [MDMPP] FIA_X509_EXT.5           | X          | X                 | X             |
| [MDMPP] FMT_MOF.1(1)             | X          |                   |               |
| [MDMPP] FMT_MOF.1(2)             | X          |                   |               |
| [MDMPP] FMT_MOF.1(3)             | X          |                   |               |
| [MDMPP] FMT_POL_EXT.1            | X          |                   |               |
| [AGENTMOD] FMT_POL_EXT.2         |            | X                 | X             |
| [MDMPP] FMT_SMF.1(1)/ANDROID     | X          |                   |               |
| [MDMPP] FMT_SMF.1(1)/IOS         | X          |                   |               |
| [MDMPP] FMT_SMF.1(2)/ANDROID     | X          |                   |               |

|                          |   |              |              |
|--------------------------|---|--------------|--------------|
| [MDMPP] FMT_SMF.1(2)/IOS | X |              |              |
| [MDMPP] FMT_SMF.1(3)     | X |              |              |
| [AGENTMOD] FMT_SMF_EXT.4 |   | X            | X            |
| [MDMPP] FMT_SMR.1(1)     | X |              |              |
| [MDMPP] FMT_SMR.1(2)     | X |              |              |
| [AGENTMOD] FMT_UNR_EXT.1 |   | X            | X            |
| [MDMPP] FPT_API_EXT.1    | X | X            | X            |
| [MDMPP] FPT_ITT.1(2)     | X | X            | X            |
| [MDMPP] FPT_LIB_EXT.1    | X | X            | X            |
| [MDMPP] FPT_TST_EXT.1    | X | <sup>6</sup> | <sup>7</sup> |
| [MDMPP] FPT_TUD_EXT.1    | X | X            | X            |
| [MDMPP] FTA_TAB.1        | X |              |              |
| [MDMPP] FTP_ITC_EXT.1    | X | X            | X            |
| [MDMPP] FTP_ITC.1(1)     | X |              |              |
| [MDMPP] FTP_TRP.1(1)     | X |              |              |
| [MDMPP] FTP_TRP.1(2)     | X | X            | X            |

Note: SFRs that originate from the Mobile Device Management Protection Profile are denoted by a [MDMPP], and SFRs that originated from the Mobile Device Management Agent PP-Module are denoted by [AGENTMOD].

The minimum configuration for this evaluation is one UEM Server, and one iOS Hub agent installed on an Apple device and/or one Android Hub agent installed on an Android device. Including additional iOS Hub agents installed on multiple Apple devices and additional Android Hub agents installed on multiple Android devices as part of an operational configuration will not affect the validity of the functional claims made within the TSS. All TSS descriptions regarding the role, operation, and management of an iOS Hub agent would be consistent with every additional iOS Hub agent and Apple device added to the minimum evaluated configuration. All TSS descriptions regarding the role, operation, and management of an Android Hub agent would be consistent with every additional Android Hub agent and Android device added to the minimum evaluated configuration. Therefore, all TSS descriptions regarding the iOS Hub agent and Android Hub agent can be read with the understanding that the descriptions would apply to one or more of these TOE components and the method in which the additional TOE components met the SFRs would be the same as their minimum configuration equivalent.

Note: The TSS evaluation activities that apply to only the UEM Server component are denoted by [SERVER] and to only the Hub agent components are denoted by [AGENT]. If the TSS evaluation activity applies to both components, it is not denoted.

## 8.1 Security Audit

### 8.1.1 [MDMPP] FAU\_ALT\_EXT.1

[SERVER] The UEM Server component of the TOE provides Authorized Administrators with the ability to view information about enrolled mobile devices and to generate alerts when various events occur. Alerts

<sup>6</sup> TD0438

<sup>7</sup> TD0438

are generated based on configurable “compliance policies” that can detect when a violation has occurred and to mark the affected device as Not Compliant in the device’s overview in the Admin Console.

Authorized Administrators can view information about the status of managed devices through the UEM Admin Console. Two of the dashboards that are accessible from the Main Menu are “Monitor” and “Devices”. From the “Monitor” section of the Admin Console, Authorized Administrators can view the total number of enrolled and unenrolled devices, the total number of compliance violations, devices that failed to install policies (profiles) and which devices have apps on a deny list, devices without required apps, or devices with apps that are not on an allow list. The Authorized Administrator can also view the applications that are associated with particular devices, including application versions. From the “Devices” section of the Admin Console, the Authorized Administrator can view changes in the enrollment status of a device by viewing the enrollment status and enrollment history information. This also lists devices that are enrolled but do not have policies applied to them. The Authorized Administrator can also view detailed information about any specific device that the UEM Server knows about under this section of the Admin Console.

In addition to being able to review this information on demand, Authorized Administrators can configure the delivery of periodic (daily, weekly, monthly) alert emails from the “Monitor” section of the Admin Console for the following events when they are observed on a device:

- Presence of apps on a deny list
- Presence of apps not on an allow list
- Absence of required apps
- Last time a device communicated with the UEM Server
- Unapproved model (iOS only)
- Unapproved device manufacturer (Android only)
- Unapproved operating system version (greater than, less than, equal to, not equal to specified version)

The UEM Server also provides Authorized Administrators with the ability to view an audit processing failure alert on the Admin Console for when a connection between the UEM Server and Syslog Server cannot be established.

### 8.1.2 [AGENTMOD] FAU\_ALT\_EXT.2/ANDROID

[AGENT] The Android Hub agent component of the TOE provides the ability to alert the UEM Server in the event that certain behavior on the underlying mobile device is observed. For Android devices, all of the alerting is performed by the Android Hub agent. The alert for a device becoming enrolled/unenrolled from management is sent by the Android Hub agent as part of the enrollment/unenrollment process which requires communication with the UEM Server. All other alerts are based upon policies (profiles) being applied to the mobile device and the Android Hub agent collecting information on the device to generate alerts based upon “compliance policies” that detect when a violation has occurred.

The Android Hub agent is configured by the UEM Server to generate periodic reachability events based upon a configured 'sample interval' and 'transmit interval'. The collecting of information on the device by the Android Hub agent occurs every 'sample interval'. The Android Hub agent then queues each sample interval of collected data and will send up to the last 10 sample intervals of collected data to the UEM



Server once the 'transmit interval' is reached. If the connection between the Android Hub agent and UEM Server is down during a 'transmit interval', the Android Hub agent continues to queue sample intervals of collected data until a connection is available for a 'transmit interval'. The maximum amount of storage is 10 sample intervals. The actual amount of storage for alerts depends on the amount of storage space of the device and the amount the device allocates to the Android Hub agent app. The alerts generated by the Android Hub agent based upon the collected data during a 'sample interval' include:

- Successful application of policies
- Generating a periodic reachability event
- Failure to install an application managed by the MAS Server functionality of the UEM Server
- Failure to update an application managed by the MAS Server functionality of the UEM Server
- Detection of apps on a deny list
- Detection of apps not on an allow list
- Required apps missing
- Unapproved device manufacturer
- Unapproved operating system version

For apps, the UEM Server has the ability to specify if a given application is required, denied, or allowed. This assignment can be made both for apps under the control of the UEM Server as well as publicly-available apps on the Google Play Store. If the Android Hub agent detects an app on a deny list upon the policy being applied to the device (i.e. the app's installation predates the policy), the alerting process immediately remediates the non-compliance by disabling the app. Otherwise, if the policy is applied prior to the installation of the app, the installation of the app is prevented. Additionally, when the installation status of an app is not compliant with the device's policies, the Android Hub agent will inform the UEM Server of these alert events.

### 8.1.3 [AGENTMOD] FAU\_ALT\_EXT.2/IOS

[AGENT] The iOS Hub agent component of the TOE provides the ability to alert the UEM Server in the event that certain behavior on the underlying mobile device is observed. For iOS/iPadOS devices, most of the alerting is performed through the iOS/iPadOS MDM protocol rather than through the iOS Hub agent. The iOS Hub agent is responsible for alerting the UEM Server component when the device is enrolled/unenrolled from management. The alert for a device becoming enrolled/unenrolled from management is sent by the iOS Hub agent as part of the enrollment/unenrollment process which requires communication with the UEM Server.

The remaining alerts are generated by the underlying iOS/iPadOS platform. These alerts are generated as part of the request and response relationship of an active connection between the iOS/iPadOS platform and the UEM Server; thus, there are no alerts to be queued when a connection is not available. This includes the iOS/iPadOS platform sending alerts when consuming policies (profiles) assigned to it as well as in response to UEM Server querying the iOS/iPadOS platform based upon the iOS Hub agent's periodic reachability event configuration. Additionally, during the periodic reachability events the iOS/iPadOS platform will collect information on the device's model, operating system version, and on installed apps to generate alerts based upon "compliance policies" that detect when a violation has occurred.

For apps, the UEM Server has the ability to specify if a given application is required, denied, or allowed. This assignment can be made both for apps under the control of the UEM Server as well as publicly-

available apps on the Apple App Store. When the installation status of an app is not compliant with the device’s policies, the iOS/iPadOS platform will inform the UEM Server of these alert events.

8.1.4 [MDMPP] FAU\_GEN.1(1)

The UEM Server, iOS Hub agent, and Android Hub agent components of the TOE generate auditable events for their own behavior. These TOE components also rely on their underlying platform to generate audit events. All TOE components rely on their underlying platforms to generate audit logs for their startup and shutdown. The UEM Server generates audit logs for all administrative actions and all commands that are sent to managed devices from the UEM Server. Audit records are generated by the UEM Server for MDM Agent alerts (FAU\_ALT\_EXT.2). The iOS and Android Hub agents will also generate audit records defined under [AGENTMOD] FAU\_GEN.1(2). The TOE components and the underlying OS platforms also generate audit data for the specific auditable events listed in Table 16 below.

Table 16: Auditable Events by Enforcing Component

| Requirement                   | Auditable Event(s)  | Component Generating Record   |
|-------------------------------|---|---|
| <b>FAU_ALT_EXT.1</b>          | Type of alert.  | UEM Server  |
| <b>FCO_CPC_EXT.1</b>          | Enabling/Disabling communications between a pair of components. | UEM Server  |
| <b>FCS_CKM.1</b>              | Failure of key generation activity for authentication keys.     | Windows Platform<br>iOS/iPadOS Platform<br>Android Platform<br><br>Note: The auditing for this SFR is invoked by the platforms’ cryptographic modules.  |
| <b>FCS_RBG_EXT.1</b>          | Failure of the randomization process.                           | Windows Platform<br>iOS/iPadOS Platform<br>Android Platform<br><br>Note: The auditing for this SFR is invoked by the platforms’ cryptographic modules.  |
| <b>FIA_ENR_EXT.1 /ANDROID</b> | Failure of MD user authentication.                              | UEM Server  |
| <b>FIA_ENR_EXT.1/IOS</b>      | Failure of MD user authentication.                              | UEM Server  |
| <b>FIA_X509_EXT.1(1)</b>      | Failure to validate X.509 certificate.                          | Windows Platform<br>iOS/iPadOS Platform<br>Android Hub Agent<br>Android Platform<br><br>Note: Platform auditing for this SFR for: (1) TLS/HTTPS is invoked by the platforms’ mechanisms which implement TLS/HTTPS communication, (2) UEM Server and Hub agent software signed updates is invoked by the platforms' app software update mechanisms, (3) UEM Server software integrity verification is invoked by Window's Authenticode mechanism, (4) signing profiles is invoked by the Windows |

| Requirement                 | Auditable Event(s)  | Component Generating Record  |
|-----------------------------|---|--|
|                             |   | platform's signature services, and (5) verifying signed profiles is invoked by the iOS/iPadOS platform's signature services.   |
| <b>FIA_X509_EXT.2</b>       | Failure to establish connection to determine revocation status.                 | Windows Platform<br>iOS/iPadOS Platform<br>Android Hub Agent<br>Android Platform<br><br>Note: Platform auditing for this SFR for: (1) TLS/HTTPS is invoked by the platforms' mechanisms which implement TLS/HTTPS communication, (2) UEM Server and Hub agent software signed updates is invoked by the platforms' app software update mechanisms, (3) UEM Server software integrity verification is invoked by Window's Authenticode mechanism, (4) signing profiles is invoked by the Windows platform's signature services, and (5) verifying signed profiles is invoked by the iOS/iPadOS platform's signature services. |
| <b>FMT_MOF.1(1)</b>         | Issuance of command to perform function.<br>Change of policy settings.          | UEM Server   |
| <b>FMT_MOF.1(2)</b>         | Enrollment by a user.   | UEM Server   |
| <b>FMT_SMF.1(2)/ANDROID</b> | Success or failure of function.   | UEM Server   |
| <b>FMT_SMF.1(2)/IOS</b>     | Success or failure of function.   | UEM Server   |
| <b>FPT_ITT.1(2)</b>         | Initiation and termination of the trusted channel.                              | Windows Platform<br>iOS/iPadOS Platform<br>Android Platform<br><br>Note: Platform auditing for this SFR for TLS/HTTPS is invoked by the platforms' mechanisms which implement TLS/HTTPS communication.   |
| <b>FPT_TST_EXT.1</b>        | Initiation of self-test. Failure of self-test.<br>Detected integrity violation. | Windows Platform<br><br>Note: Platform auditing for this SFR for UEM Server software integrity verification is invoked by Window's Authenticode mechanism.   |
| <b>FPT_TUD_EXT.1</b>        | Success or failure of signature verification.                                   | Windows Platform<br>iOS/iPadOS Platform<br>Android Platform<br><br>Note: Platform auditing for this SFR for UEM Server and Hub agent software signed updates is invoked by the platforms' app software update mechanisms.  |
| <b>FTA_TAB.1</b>            | Change in banner setting.   | UEM Server   |

| Requirement         | Auditable Event(s)                                 | Component Generating Record  |
|---------------------|--|--|
| <b>FTP_ITC.1(1)</b> | Initiation and termination of the trusted channel. | Windows Platform<br><br>Note: Platform auditing for this SFR for TLS/HTTPS is invoked by the Windows platform’s mechanism which implements TLS/HTTPS communication.                                    |
| <b>FTP_TRP.1(1)</b> | Initiation and termination of the trusted channel. | Windows Platform<br><br>Note: Platform auditing for this SFR for TLS/HTTPS is invoked by the Windows platform’s mechanism which implements TLS/HTTPS communication.                                    |
| <b>FTP_TRP.1(2)</b> | Initiation and termination of the trusted channel. | Windows Platform<br>iOS/iPadOS Platform<br>Android Platform<br><br>Note: Platform auditing for this SFR for TLS/HTTPS is invoked by the platforms’ mechanisms which implement TLS/HTTPS communication. |

TOE audit records are recorded with the following format:

{Syslog Date and Time} {UEM Server IP Address or Fully Qualified Domain Name} {Date and Time} {UEM Server Name} AirWatch Syslog Details are as follows Event Type: {EventType}; Event: {Event}; User: {User}; Device Name: {DeviceFriendlyName}; EnrollmentUser: {EnrollmentUser}; Event Source: {EventSource}; Event Module: {EventModule}; Event Category: {EventCategory}; Event Data: {EventData}

The audit records that are generated include at least the following information: date and time of the event {Date and Time}, event type {EventType}, subject identity {User}, success or failure of the event {Event}, and where (i.e. Device or Server) the event occurred {EventSource}. When identifying the mobile device this will be in the Device Name: {DeviceFriendlyName} field. Additional contents required by the audit records are usually found in the Event Data: {EventData} field.

The following is an example of an audit record from the UEM Server in order to illustrate the audit record format and the fields contained within these records.

Oct 2 07:21:59 uem.cctl.company.com 2022-10-02T12:20:04.625949Z AirWatch AirWatch Syslog Details are as follows Event Type: Device; **Event: EnrollmentComplete**; User: *sysadmin*; Device Name: Tester1Local Android Android 11.0 RF8M803LV4L; EnrollmentUser: Tester1Local; **Event Source: Server**; Event Module: Enrollment; **Event Category: Enrollment**; Event Data: Url=  
uem.cctl.company.com Event Timestamp: 2022-10-02T12:19:59.413000

The audit records that are generated include at least the following information: date and time of the event (underlined text), event type (**bold text**), subject identity (*italicized text*), success or failure of the event (***bold italicized text***), and where the event occurred (**bold underlined text**). The type of event and additional audit record contents are described in Table 13. For a full list of audit record examples, refer to Section 8.1.3 of the VMware Workspace ONE Unified Endpoint Management v2209 Supplemental Administrative Guidance version 1.0.

8.1.5 [MDMPP] FAU\_GEN.1(2)

[SERVER] The UEM Server component of the TOE, which includes the MAS Server functionality, generates audit records when it is unable to push an application to a managed device or detects that a required application is not present on the device (including failed updates to existing managed applications). This is done by defining a compliance policy that checks for the absence of a specific application which causes the device to generate an audit event on its next periodic check-in if that condition is met. The TOE creates an audit record for each device based upon the compliance policy.

TOE audit records are recorded with the following format:

{Syslog Date and Time} {UEM Server IP Address or Fully Qualified Domain Name} {Date and Time} {UEM Server Name} AirWatch Syslog Details are as follows Event Type: {EventType}; Event: {Event}; User: {User}; Device Name: {DeviceFriendlyName}; EnrollmentUser: {EnrollmentUser}; Event Source: {EventSource}; Event Module: {EventModule}; Event Category: {EventCategory}; Event Data: {EventData}

The audit records that are generated include at least the following information: date and time of the event {Date and Time}, event type {EventType}, subject identity {User}, success or failure of the event {Event}, and where (i.e. Device or Server) the event occurred {EventSource}. When identifying the mobile device this will be in the Device Name: {DeviceFriendlyName} field. Additional contents required by the audit records are usually found in the Event Data: {EventData} field.

The following is an example of an audit record from the UEM Server in order to illustrate the audit record format and the fields contained within these records.

Nov 20 06:56:54 uem.cctl.company.com 2022-10-20T11:54:59.654511Z AirWatch AirWatch Syslog Details are as follows Event Type: Device; Event: InstallApplicationFailed; User: sysadmin; *Device Name: Tester1 iPad iOS 14.6.0 GG7G5FWZQ16M*; EnrollmentUser: Tester1; Event Source: Device; Event Module: CustomDeviceEvents; **Event Category: Command**; Event Data: Event Timestamp: 2022-10-20T11:54:59.080000

The audit records that the MAS Server component creates include the following information: date and time of the event (underlined text), event type (**bold text**), and mobile device identity (*italicized text*). For a full list of audit record examples, refer to Section 8.1.3 of the VMware Workspace ONE Unified Endpoint Management v2209 Supplemental Administrative Guidance version 1.0.

8.1.6 [AGENTMOD] FAU\_GEN.1(2)

[AGENT] The iOS and Android Hub agent component of the TOE and the underlying OS platforms generate audit events for activities on the device. The iOS and Android Hub agents’ underlying platforms generate audit logs for their startup and shutdown. The TOE components and the underlying OS platforms also generate audit data for the specific auditable events listed in Table 17 below.

Table 17: Agent Auditable Events by Enforcing Component

| Requirement            | Auditable Event(s)                | Component Generating Record |
|------------------------|-----------------------------------|-----------------------------|
| FAU_ALT_EXT.2 /ANDROID | Success/failure of sending alert. | Android Hub Agent           |
| FAU_ALT_EXT.2 /IOS     | Success/failure of sending alert. | iOS/iPadOS Platform         |

| Requirement          | Auditable Event(s)  | Component Generating Record  |
|----------------------|---|--|
|                      |   | Note: Platform auditing for this SFR is invoked by the iOS internal MDM agent upon communication with the UEM Server to send alert data.   |
| <b>FAU_SEL.1</b>     | All modifications to the audit configuration that occur while the audit collection functions are operating. | iOS Hub Agent<br>iOS/iPadOS Platform<br>Android Hub Agent<br><br>Note: Platform auditing for this SFR is invoked by the iOS internal MDM agent upon receiving policies from the UEM Server.                                  |
| <b>FIA_ENR_EXT.2</b> | Enrollment in management.   | iOS/iPadOS Platform<br>Android Hub Agent   |
| <b>FMT_POL_EXT.2</b> | Failure of policy validation.   | iOS Hub Agent<br>iOS/iPadOS Platform<br>Android Hub Agent<br><br>Note: Platform auditing for this SFR is invoked by the iOS internal MDM agent upon receiving policies from the UEM Server.                                  |
| <b>FMT_SMF_EXT.4</b> | Outcome (Success/failure) of function.  | iOS Hub Agent<br>iOS/iPadOS Platform<br>Android Hub Agent<br>Android Platform<br><br>Note: Platform auditing for this SFR is invoked by the iOS internal MDM agent upon receiving policies and commands from the UEM Server. |

TOE audit records are recorded with the following format:

{Syslog Date and Time} {UEM Server IP Address or Fully Qualified Domain Name} {Date and Time} {UEM Server Name} AirWatch Syslog Details are as follows Event Type: {EventType}; Event: {Event}; User: {User}; Device Name: {DeviceFriendlyName}; EnrollmentUser: {EnrollmentUser}; Event Source: {EventSource}; Event Module: {EventModule}; Event Category: {EventCategory}; Event Data: {EventData}

The audit records that are generated include at least the following information: date and time of the event {Date and Time}, event type {EventType}, subject identity {User}, success or failure of the event {Event}, and where (i.e. Device or Server) the event occurred {EventSource}. When identifying the mobile device this will be in the Device Name: {DeviceFriendlyName} field. Additional contents required by the audit records are usually found in the Event Data: {EventData} field.

The following is an example of an audit record from the iOS Hub agent in order to illustrate the audit record format and the fields contained within these records.

Dec 14 11:24:10 uem.ctl.company.com 2022-12-14T16:22:15.016288Z 10.137.2.36 AirWatch Syslog Details are as follows Event Type: Device; **Event: InstallProfileConfirmed**; User: sysadmin; Device Name: CCTestAD1 iPhone iOS 14.7.1 F17G80KL0D81; EnrollmentUser: CCTestAD1; Event Source: Device; Event Module: Devices; **Event Category: Command**; Event Data: Profile=Test Case 008 - Initial Policy Event Timestamp: 2022-12-14T16:22:14.297000

The audit records that are generated include at least the following information: date and time of the event (underlined text), event type (**bold text**), subject identity (*italicized text*), and success or failure of the event (***bold italicized text***). The type of event and additional audit record contents are described in Table 14. For a full list of audit record examples, refer to Section 8.1.3 of the VMware Workspace ONE Unified Endpoint Management v2209 Supplemental Administrative Guidance version 1.0.

#### 8.1.7 [MDMPP] FAU\_NET\_EXT.1

[SERVER] Authorized Administrators can use the Admin Console to determine the network connectivity status of devices that have iOS and Android Hub agents installed on them. Each time a device connects to the UEM Server, the UEM Server will record the time of that connection identifying the device's network connectivity. Devices will connect to UEM Server based upon the iOS and Android Hub agents' periodic reachability event configuration or in response to an on-demand request by an Authorized Administrator. Android device periodic reachability events are initiated by the Android Hub agents, iOS device periodic reachability events are initiated by the iOS/iPadOS platform, and the on-demand requests by Authorized Administrators for connectivity status for iOS and Android devices are initiated by the UEM Server.

An Authorized Administrator can make an on-demand request from the Device's dashboard by performing a Device Query. This action will result in the UEM Server determining connectivity status for one or more selected devices by using push notifications. For iOS devices, the TSF uses the Apple Push Notification Service (APNS) to send the request to the device and the device will respond to the request when it has network connectivity. For Android devices, the TSF uses Firebase Cloud Messaging Services (FCM) to send the request to the device and the device will respond to the request when it has network connectivity. Refer to FAU\_ALT\_EXT.2/IOS and FAU\_ALT\_EXT.2/ANDROID for more information about how the iOS and Android Hub agents communicate device statuses back to the UEM Server.

#### 8.1.8 [MDMPP] FAU\_SAR.1

[SERVER] For audited events that are generated by the UEM Server component of the TOE, the Monitor dashboard on the Admin Console provides administrators with the ability to review audit records. This provides a graphical view of the log data in a human-readable format. Audit data can be searched and sorted using this interface.

For cryptographic behavior that is performed by the UEM Server's underlying platform, auditing is stored in the Windows event logs. These records can also be sorted, filtered, and searched, but this activity is performed using the platform since the TSF is not responsible for generating this data.

Table 16 shows the auditable events that are logged by UEM Server versus its underlying platform. The component (i.e. TOE or TOE platform) used to review the audit data is the same as the component that is used to generate the data to be reviewed.

#### 8.1.9 [AGENTMOD] FAU\_SEL.1(2)

[AGENT] There is no specific configuration to turn on and off auditing on the TOE, thus the iOS and Android Hub agents and the underlying OS platforms will always perform auditing. However, an Authorized Administrator creates policies on the UEM Server and will assign them to one or more devices. These policies include requirements for the iOS Hub agent, iOS/iPadOS platform, and Android Hub agent to generate audit records for the functionality configured in the policies. Once the iOS Hub agent,

iOS/iPadOS platform, and Android Hub agent receive and apply a policy requiring auditing, they will always generate the necessary audit records. For example, an Authorized Administrator can create a policy that an app is required. The policy will be sent and applied by the Android Hub agents or iOS/iPadOS platforms on the devices to which the policy has been assigned. If a device does not have the required app, the Android Hub agent or iOS/iPadOS platform will create an audit record based upon that event type as well as success/failure. Table 20 describes if a policy is ‘implemented by’ the iOS Hub agent, iOS/iPadOS platform, or Android Hub agent.

The TSF does not support specification of more complex audit pre-selection criteria, such as multiple attributes or logical expressions using attributes.

#### **8.1.10 [MDMPP] FAU\_STG\_EXT.1**

[SERVER] In the evaluated configuration, audit data managed by the UEM Server will be transmitted from the UEM Server to a remote Syslog Server over a TLS v1.2 encrypted trusted channel as well as to the SQL database. The actual TLS encryption is handled by the underlying Windows Server 2019 platform. The audit data that are transferred include audit records generated by the UEM Server software as well as audit records that are received from iOS and Android Hub agents. This does not include audit data that are generated by these TOE components’ underlying platforms as this audit data are not managed by the TOE’s software boundary. It is therefore the responsibility of the Operational Environment to securely transfer this audit data to a remote location for permanent storage.

The UEM Server is configured to send TOE managed audit data over a specific port to the Syslog Server and if this matches the syslog TLS port, then the Syslog Server will receive a TLS connection initiation. The Syslog Server’s certificate is bound to the port, which is defaulted to port 6514 but can be changed by a System Administrator. Once the connection is established, the TOE managed audit logs are sent to the Syslog Server in real time upon the UEM Server creating them or receiving them from Hub agents. If the Syslog Server connectivity is unavailable, audit records will only be stored in the SQL database while the connection is down. Upon re-establishment of communications with the Syslog Server, new audit records will resume being transmitted to it but the audit records that were generated during the time the Syslog Server connection was down are not sent to the Syslog Server. The SQL database only stores audit records for 30 days before purging them and if maximum capacity for record storage is reached in the SQL database, any new audit records are dropped.

[AGENT] Audit records generated by the iOS and Android Hubs Agents are transmitted to the UEM Server over the HTTPS internal TOE trusted channel. iOS and Android Hubs Agent audit records are generated and sent to the UEM Server in real time. As described above, once these audit records reach the UEM Server, the audit records will be sent to the remote Syslog Server over a TLS v1.2 encrypted trusted channel.

## **8.2 Communication**

### **8.2.1 [MDMPP] FCO\_CPC\_EXT.1**

[SERVER] In the evaluated configuration, the Hub agents that can join the TOE by enrolling with the UEM Server are limited based upon the allowed DEP identifiers for iOS Hub agents and the allowed IMEIs and/or serial numbers for Android Hub agents. The configuration of these enrollment restrictions is



the enablement step by the Authorized Administrator through the Admin Console. The entire enrollment process from the UEM Server's perspective is described under FIA\_ENR\_EXT.1/IOS, FIA\_ENR\_EXT.1/ANDROID, and FIA\_X509\_EXT.5. The UEM Server relies on its underlying platform to provide the secure registration channel to the iOS and Android Hub agents that attempt to enroll and join the TOE. This secure registration channel is described under FTP\_TRP.1(2) and is used for all communications between the UEM Server and Hub agents during the enrollment process. After a Hub agent has enrolled and joined the TOE, an Authorized Administrator can disable the communication between a Hub agent and the UEM Server by wiping the Hub agent's device. This will result in the removal of the Hub agent and the unenrollment of the device which will prevent communication between the device and the UEM Server.

[AGENT] The entire enrollment process from the Hub agent's perspective is described under FIA\_ENR\_EXT.1/IOS, FIA\_ENR\_EXT.1/ANDROID, FIA\_ENR\_EXT.2, and FIA\_X509\_EXT.5. The iOS and Android Hub agent rely on their underlying platform to provide the secure registration channel to the UEM Server when enrolling into management and joining the TOE. This secure registration channel is described under FTP\_TRP.1(2) and is used for all communications between the Hub agents and UEM Server during the enrollment process.

### **8.3 Cryptographic Support**

[SERVER] Cryptographic services for the UEM Server are provided by the underlying Windows server platform. The Windows Server 2019 platform uses Microsoft's SymCrypt to perform all cryptographic services.

[AGENT] Cryptographic services for the iOS and Android Hub agents are mainly provided by the underlying mobile device platforms. The iOS Hub agent uses Apple iOS/iPadOS platforms' CoreCrypto Module to perform all claimed cryptographic services. The Android Hub agent uses the Android platform's SCrypto and BoringSSL cryptographic modules to perform all claimed cryptographic services, except for the policy digital signature validation requirements. The Android Hub agent implements OpenSSL for the specific purpose of performing the policy digital signature validation services.

Refer to the platform Security Targets, listed in Section 1.1.5, for more information about the cryptographic functionality provided by the Windows Server, iOS/iPadOS, and Android platforms and their corresponding cryptographic certificates.

#### **8.3.1 [MDMPP] FCS\_CKM.1**

[SERVER] The UEM Server invokes the Windows Server 2019 platform provided functionality for asymmetric key generation in support of TLS communications. The Windows Server 2019 platform provides functionality to support RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and ECC schemes using "NIST curves" P-256 and P-384 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

[AGENT] The iOS and Android Hub agent's software invokes the platform provided functionality in support of TLS communications. Both iOS and Android platforms support RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and ECC schemes using "NIST curves" P-256 and P-384 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

### 8.3.2 [MDMPP] FCS\_CKM.2

[SERVER] The UEM Server invokes the underlying Window server platform in support of two key establishment schemes for the establishment of TLS communications:

- RSA key establishment conforming to “RSAES-PKCS1-v1\_5 as specified in Section 7.2 of RFC 8017” and
- Elliptic curve-based key establishment conforming to NIST Special Publication 800-56A.

[AGENT] The iOS and Android Hub agent's software relies on the underlying mobile device platform to perform key establishment for TLS communications using the following two key establishment schemes:

- RSA key establishment conforming to “RSAES-PKCS1-v1\_5 as specified in Section 7.2 of RFC 8017” and
- Elliptic curve-based key establishment conforming to NIST Special Publication 800-56A

### 8.3.3 [MDMPP] FCS\_CKM\_EXT.4

[SERVER] The UEM Server invokes the underlying platform’s FIPS cryptographic module to zeroize keys and cryptographic security parameter data when no longer needed. The invoking of key destruction occurs as a result of the UEM Server making a cryptographic function call which requires a key and/or cryptographic security parameter data to be generated by the platform. The platform will therefore perform key destruction when the generated cryptographic data are no longer needed, without requiring a separate function call for key destruction from the UEM Server. All key data maintained by the server platform exists only in volatile memory and are erased by a one-pass overwrite with zeroes followed by a read-verify.

[AGENT] The iOS and Android Hub agents’ software invokes the underlying mobile device platform to perform key destruction. The invoking of key destruction occurs as a result of the iOS or the Android Hub agent making a cryptographic function call which requires a key and/or cryptographic security parameter data to be generated by its platform. The platform will therefore perform key destruction when the generated cryptographic data are no longer needed, without requiring a separate function call for key destruction from the iOS or the Android Hub agent. Key data maintained by the iOS and Android Hub agents’ platform in volatile memory are erased by a one-pass overwrite with zeroes. Key data maintained by the iOS and Android Hub agents’ platforms in non-volatile memory are stored in wear-leveled flash memory and are erased by a one-pass overwrite with ones (i.e. block erase).

### 8.3.4 [MDMPP] FCS\_COP.1(1)

[SERVER] The UEM Server invokes the underlying Windows Server 2019 platform to perform AES encryption/decryption services for TLS communications and protection of data at rest in platform key storage. All data at rest are protected using AES-GCM-256 as defined in NIST SP 800-38D. Data in transit are protected using GCM mode and either 128-bit or 256-bit keys. The UEM Server’s platform conforms to NIST SP 800-38D.

[AGENT] The iOS and Android Hub agents’ software invokes the underlying mobile device platform to perform symmetric encryption/decryption. The iOS and Android Hub agents’ platforms are using GCM mode and either 128-bit or 256-bit keys. The device platforms conform to NIST SP 800-38D.

### 8.3.5 [MDMPP] FCS\_COP.1(2)

[SERVER] The UEM Server invokes the Windows server platform to provide SHA-256, SHA-384 and SHA-512 cryptographic hashing services, conformant to FIPS PUB 180-4, with digest sizes of 256, 384 and 512 bits, respectively.

- SHA-256 and SHA-384 are used by HMAC in message authentication in TLS communication.
- SHA-256 and SHA-384 in support of ECDSA with P-256 and P-384 curves used for digital signature services in support of TLS communications.
- SHA-512 is used in the hashing of passwords and for the digital signing of policies (ESDSA with P-521 curve).

[AGENT] The iOS and Android Hub agents' software invokes the underlying mobile device platform to perform cryptographic hashing in support of TLS communication. The iOS and Android Hub agents' platforms use SHA-256 and SHA-384, conformant to FIPS PUB 180-4, in support of TLS communication.

Additionally, the iOS Hub agent invokes the underlying mobile device iOS/iPadOS platform for SHA-512 hashing services, conformant to FIPS PUB 180-4, for ECDSA policy digital signature verification.

The Android Hub agent implements SHA-512 hashing services, conformant to FIPS PUB 180-4, for ECDSA policy digital signature verification. The CAVP SHS certificate number is #A3270.

### 8.3.6 [MDMPP] FCS\_COP.1(3)

[SERVER] The UEM Server invokes the Windows Server 2019 platform to provide all digital signature services in accordance with FIPS PUB 186-4. RSA with 2048-bit keys and ECDSA with P-256 and P-384 NIST curves are used for digital signature services in support of TLS communication. Additionally, ECDSA with P-521 NIST curve (using SHA-512) is used for policy signature generation.

[AGENT] The iOS and Android Hub agents' software invokes the underlying mobile device platform to provide digital signature services in accordance with FIPS PUB 186-4. RSA with 2048-bit keys and ECDSA with P-256 and P-384 NIST curves are used for digital signature services in support of TLS communication.

Additionally, the iOS Hub agent invokes the underlying mobile device's iOS/iPadOS platform to provide ECDSA with P-521 NIST curve (using SHA-512) services for policy signature verification.

The Android Hub agent implements the ECDSA with P-521 NIST curve (using SHA-512) services for policy signature verification functionality. The CAVP ECDSA certificate number is #A3270.

### 8.3.7 [MDMPP] FCS\_COP.1(4)

[SERVER] The UEM Server invokes the Windows Server 2019 platform to provide keyed-hash message authentication services conformant to FIPS PUBs 180-4 and 198-1. HMAC-SHA-256 and HMAC-SHA-384 are used to perform keyed-hash message authentication, with respective digest sizes of 256 and 384 bits in support of trusted communication. The key used by HMAC is the UUID which is 160 bits. This key is stored on the Server platform.

[AGENT] The iOS and Android Hub agents' software invokes the underlying mobile device platform to provide keyed-hash message authentication services conformant to FIPS PUBs 180-4 and 198-1. HMAC-SHA-256 and HMAC-SHA-384 are used to perform keyed-hash message authentication, with respective

digest sizes of 256 and 384 bits in support of trusted communication. The key used by HMAC is the UUID which is 160 bits.

8.3.8 [MDMPP] FCS\_RBG\_EXT.1

Note: The TOE UEM Server and Hub agent software do not directly invoke their respective platforms’ deterministic random bit generator. Instead, the TOE’s software indirectly invokes their platforms’ deterministic random bit generator by directly invoking platform components, which in turn directly invoke the deterministic random bit generator.

[SERVER] The UEM Server invokes the underlying Windows Server 2019 platform to provide random bit generation services. The platform cryptographic module provides an AES counter DRBG (CTR\_DRBG) that conforms to NIST SP 800-90A. In order to provide sufficient randomness to the keys generated by this DRBG (as specified in NIST SP 800-57), the DRBG is seeded with at least 256 bits of entropy which is gathered from a platform based RBG.

[AGENT] The iOS and Android Hub agents’ software invokes the underlying mobile device platform to provide random bit generation services. Both iOS and Android platforms implement AES counter DRBG conformant to NIST SP 800-90A. The iOS DRBG is seeded with at least 256 bits of entropy from the platform’s software-based noise source. The Android DRBG is seeded with at least 256 bits of entropy from the platform’s hardware-based noise source.

8.3.9 [MDMPP] FCS\_STG\_EXT.1

[SERVER] The TOE platform is responsible for storing keys and relies on the SymCrypt cryptographic library to invoke the storage of persistent secrets and private keys which are produced through their operation. All private keys are stored in the Windows Trust Store and all user credentials are stored in authentication repositories. The SQL database Master Key is stored encrypted using an AES-GCM 256-bit key encryption key (KEK). This KEK is encrypted and stored in the Windows Registry. The policy signing X.509v3 certificate is uploaded by the Authorized Administrator and is stored in the SQL database.

The following table contains the list of keys and CSPs for the UEM Server platform:

Table 18: Keys and CSPs for the UEM Server Platform

| Key/CSP                  | Purpose                         | Origin                      |
|--------------------------|---------------------------------|-----------------------------|
| AES-GCM                  | Data at rest<br>Data in transit | Windows Platform - SymCrypt |
| RSA Public/Private Key   | TLS Key Establishment           | Windows Platform - SymCrypt |
| RSA Public/Private Key   | Digital Signatures              | Windows Platform - SymCrypt |
| ECDH Public/Private Key  | TLS Key Establishment           | Windows Platform - SymCrypt |
| HMAC Key                 | Message Authentication          | Windows Platform - SymCrypt |
| RBG CSPs                 | Random Bit Generation           | Windows Platform - SymCrypt |
| ECDSA Public/Private Key | KeyGen for ECDH                 | Windows Platform - SymCrypt |
| ECDSA Public/Private Key | Digital Signatures              | Windows Platform - SymCrypt |
| X.509v3 Certificates     | Digital Signatures              | CA Server                   |

[AGENT] The FCS\_STG\_EXT.1(2) section describes key storage for the iOS and Android Hub agents.

8.3.10 [AGENTMOD] FCS\_STG\_EXT.1(2)

[AGENT] All iOS Hub agent keys are stored in the Trust Anchor and all the Android Hub agent keys are stored in the Android Trust Store of the device. The iOS Hub agent relies on its platform’s CoreCrypto module to invoke the storage of persistent secrets and private keys which are produced through its operation. The Android Hub agent relies on its platform’s BoringSSL and SCrypto to invoke the storage of persistent secrets and private keys which are produced through their operation. These cryptographic modules are invoked by the platform APIs available to the Hub agents when requesting an encryption function. (see section 8.6.1 for the list of APIs).

The following table contains the list of keys and CSPs for the iOS and Android Hub agents along with their purpose:

**Table 19: Keys and CSPs for the Device**

| Key/CSP                  | Purpose                | Origin   |
|--------------------------|------------------------|--|
| AES-GCM                  | Data in transit        | iOS/iPadOS Platform - CoreCrypto<br>Android Platform - BoringSSL and SCrypto |
| RSA Public/Private Key   | TLS Key Establishment  | iOS/iPadOS Platform - CoreCrypto<br>Android Platform - SCrypto               |
| ECDH Public/Private Key  | TLS Key Establishment  | iOS/iPadOS Platform - CoreCrypto<br>Android Platform - BoringSSL             |
| HMAC Key                 | Message Authentication | iOS/iPadOS Platform - CoreCrypto<br>Android Platform - BoringSSL and SCrypto |
| RBG CSPs                 | Random Bit Generation  | iOS/iPadOS Platform - CoreCrypto<br>Android Platform – BoringSSL and SCrypto |
| ECDSA Public/Private Key | KeyGen for ECDH        | iOS/iPadOS Platform - CoreCrypto<br>Android Platform - BoringSSL and SCrypto |
| X.509v3 Certificates     | Digital Signatures     | CA Server  |

**8.4 Identification and Authentication**

8.4.1 [MDMPP] FIA\_ENR\_EXT.1/ANDROID

[SERVER] In the evaluated configuration of the TOE, a user enrolls their Android mobile device through a series of steps. First, the user powers on the mobile device and follows the standard Android Setup Assistant instructions, including language, country/region, and Wi-Fi network as well as downloading the Android Hub agent from the Google Play Store.

The device user will then enter the UEM Server’s URL into the Android Hub agent which will be used to establish the enrollment connection that is described under FTP\_TRP.1(2). Once the HTTPS/TLS connection between an Android Hub agent and UEM Server is established, the user provides their credentials to authenticate to the UEM Server and then the enrollment process begins. There are two methods of configuring user authentication for device enrollment:

- **Basic:** The account has a username/password defined by an Authorized Administrator on the UEM Server.
- **LDAP:** The UEM Server is connected to an Active Directory/LDAP Server that is used as a third-party identity store.

The UEM Server can limit the user's enrollment of Android devices based on the device's IMEI and/or serial number, the specific model and/or manufacturer of the device, the number of devices enrolled by a user, and the installed operating system version. An authorized Administrator can configure enrollment restriction policies through the Admin Console based upon these factors. The UEM Server will then initiate the process of having a unique X.509v3 certificate being issued to the Android Hub agent per the process described under FIA\_X509\_EXT.5 and will send the MDM profiles (policies) assigned to the device.

In addition, if a device is lost or stolen, a deny list of devices can be created using serial number/IMEI information. Any deny listing of a device will unenroll it, remove all MDM profiles, and prevent re-enrollment until the device is removed from the deny list.

#### 8.4.2 [MDMPP] FIA\_ENR\_EXT.1/IOS

[SERVER] In the evaluated configuration of the TOE, a user enrolls their iOS mobile device through a series of steps. First, an Administrator will enroll the device in Apple DEP which is performed using the device's serial number. Then the user powers on the mobile device and follows the standard iOS Setup Assistant instructions, including language, country/region, and Wi-Fi network. Additionally, the iOS Setup Assistant will continue the enrollment process to the UEM Server through Apple DEP.

As part of enrolling in Apple DEP, the iOS/iPadOS platform will receive the UEM Server's URL which will be used to establish the enrollment connection that is described under FTP\_TRP.1(2). Once the HTTPS/TLS connection between an iOS/iPadOS platform and UEM Server is established, the user provides their credentials to authenticate to the UEM Server. There are two methods of configuring user authentication for device enrollment:

- **Basic:** The account has a username/password defined by an Authorized Administrator on the UEM Server.
- **LDAP:** The UEM Server is connected to an Active Directory/LDAP Server that is used as a third-party identity store.

The UEM Server can limit the user's enrollment of specific iOS devices based on device registration with Apple DEP. The DEP registration list of devices' DEP identifiers effectively acts as an allow list for the device. This is done by an Authorized Administrator specifying Registered Devices Only in the registration settings; the UEM Server will acquire the list of registered devices through periodic synchronization with Apple DEP. Once authentication is successful, the iOS Hub agent is then deployed as a managed app by the UEM Server to the iOS mobile device. The UEM Server will then initiate the process of having a unique X.509v3 certificate being issued to the iOS Hub agent per the process described under FIA\_X509\_EXT.5 and will send the MDM profiles (policies) assigned to the device.

In addition, if a device is lost or stolen, a deny list of devices can be created using UDID information. Any deny listing of a device will unenroll it, remove all MDM profiles, and prevent re-enrollment until the device is removed from the deny list.

### 8.4.3 [AGENTMOD] FIA\_ENR\_EXT.2

[AGENT] During the enrollment process, the iOS and Android Hub agents record the UEM Server's DNS name and full URL with hostname. This is the only reference identifier used for the UEM Server. The iOS and Android Hub agents can only be enrolled with one UEM Server at a time.

### 8.4.4 [MDMPP] FIA\_UAU.1

[SERVER] The UEM Server component of the TOE has a configurable login warning banner which is displayed prior to authentication taking place for both the Admin Console and the Self-Service Portal. There is also a "forgot password" link for the Administrator on the Admin Console that can be used to recover credentials based on username. An email is sent out to the Administrator using the address that is stored by the TOE with a link and once the link is selected, a security question must be answered in order to reset the password. The security questions are supplied over a TLS protected link. The TSF does not actually transmit a password to the Administrator.

An Administrator can also select the language the Admin Console will be displayed in from the pre-authentication homepage. In addition, there is an "About VMware AirWatch" button on the Admin Console pre-authentication homepage that includes the UEM Server software version number, copyright and licensing agreement information.

All other means of interacting with the UEM Server component of the TOE require the Administrator to be authenticated to the Admin Console or the user to be authenticated to the Self-Service Portal.

### 8.4.5 [MDMPP] FIA\_X509\_EXT.1(1) and [MDMPP] FIA\_X509\_EXT.2

[SERVER] The UEM Server relies on the underlying platform to provide X.509v3 certificate services for verification of the code signing of UEM Server software updates, for integrity verification of the UEM Server software, and signing profiles (policies) that are sent to iOS and Android Hub agents. The UEM Server's underlying platform signs profiles (policies) with the certificate that the authorized Administrator uploads to the Admin Console specifically for policy signing. The X.509v3 certificates used for the UEM Server's software integrity verification and software updates are signed using a public CA certificate during the software build. The underlying platform is able to determine which certificate to use for the validity check based upon the presented certificate's data. The UEM Server also relies on its platform to provide its X.509v3 certificate as part of TLS and HTTPS/TLS session establishment in all cases where the UEM Server is the server component in the session (e.g. connections to Hub agents) as well as upon the server component's request where the UEM Server is the client component and mutual authentication has been configured. The UEM Server's platform will validate all certificates it receives as part of TLS and HTTPS/TLS session establishment. The HTTPS/TLS connection between the iOS and Android Hub agents and the UEM Server requires that the administrator bind an X.509v3 certificate to ports 443 and 8443 on the UEM Server.

The UEM Server's platform performs the following checks in order to determine if a certificate is valid:

- Certificate validation and certificate path validation conforms to RFC 5280.
- The certificate path must terminate with a trusted CA certificate.
- All certificates must have the basicConstraints extension present, the CA flag set to TRUE for all CA certificates, and any path constraints are met.

- All CA certificates must have the caSigning purpose in the key usage field.
- The UEM Server's TOE platform uses the OCSP as specified in RFC 6960 to verify revocation status. Revocation checking occurs each time a certificate is presented for a validation check. If the UEM Server's TOE platform cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.
- The extendedKeyUsage field must be valid based on the following rules:
  - Certificates used for trusted updates and executable code integrity verification must 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 must 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 must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - CSP certificates presented for OCSP responses must have the OCSP signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST must have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

Additionally, the UEM Server platform's certificate validation service will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE.

[AGENT] The iOS and Android Hub agents rely on the underlying mobile platform's cryptographic modules to provide X.509v3 certificate services for verification of the code signing of Hub agent software updates and in support of HTTPS/TLS connections to the Hub agents. The Hub agents' software updates are signed using a public CA certificate during the software build. The underlying platform is able to determine which certificate to use for the validity check based upon the presented certificate's data. The Hub agents' platforms will present an X.509v3 certificate as part of the HTTPS/TLS session establishment in all cases where the Hub agent is the client component and mutual authentication has been configured (e.g. connections to UEM Server). Upon request for an X.509v3 certificate, the iOS and Android Hub agents' underlying platforms will present the unique certificate that the mobile device was issued as part of FIA\_X509\_EXT.5.

Additionally, for the iOS Hub agent only, the platform implements X.509v3 certificate services for the verification of signed profiles (policies) received from the UEM Server that will be applied by the iOS Hub agent or iOS/iPadOS underlying platform. The underlying platform is able to determine which certificate to use for the validity check based upon the presented certificate's data.

The iOS and Android Hub agents' platforms perform the following checks in order to determine if a certificate is valid:

- Certificate validation and certificate path validation conforms to RFC 5280.
- The certificate path must terminate with a trusted CA certificate.
- All certificates must have the basicConstraints extension present, the CA flag set to TRUE for all CA certificates, and any path constraints are met.
- All CA certificates must have the caSigning purpose in the key usage field.



- Revocation checking occurs each time a certificate is presented for a validation check.
  - The Android Hub agents' platform uses the OCSP as specified in RFC 6960 to verify revocation status. If the Android Hub agents' platforms cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.
  - The iOS Hub agents' platform uses an OCSP TLS Status Request Extension (i.e., OCSP stapling) as specified in RFC 6066. If the iOS Hub agents' platforms cannot establish a connection to determine the validity of a certificate, then the certificate is accepted.
- The extendedKeyUsage field must be valid based on the following rules:
  - Certificates used for trusted updates and executable code integrity verification must 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 must 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 must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - CSP certificates presented for OCSP responses must have the OCSP signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST must have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

Additionally, the iOS and Android Hub agents platforms' certificate validation services will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE, except for certificates related to signed profiles (policies) processed by the Android Hub agent.

The Android Hub agent's instance of OpenSSL implements X.509v3 certificate services for the verification of signed profiles (policies) received from the UEM Server. The Android Hub agent is able to determine which certificate to use for the validity check based upon the presented certificate's data.

The Android Hub agent's instance of OpenSSL performs the following checks in order to determine if a certificate related to signed profiles (policies) is valid:

- Certificate validation and certificate path validation conforms to RFC 5280.
- The certificate path must terminate with a trusted CA certificate.
- All certificates must have the basicConstraints extension present, the CA flag set to TRUE for all CA certificates, and any path constraints are met.
- All CA certificates must have the caSigning purpose in the key usage field.
- The Android Hub agent uses the OCSP as specified in RFC 6960 to verify revocation status. Revocation checking occurs each time a certificate is presented for a validation check. If the Android Hub agent cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.
- The extendedKeyUsage field must be valid based on the following rules:
  - Certificates used for trusted updates and executable code integrity verification must 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 must 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 must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
- CSP certificates presented for OCSP responses must have the OCSP signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
- Server certificates presented for EST must have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

Additionally, the instance of OpenSSL on the Android Hub agent will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE for certificates related to signed profiles (policies).

#### **8.4.6 [MDMPP] FIA\_X509\_EXT.5**

[SERVER] The UEM Server platform requires each iOS and Android Hub agent device to have a unique X.509v3 certificate which is used by UEM Server to perform the client-side authentication of the device as part of the Hub agent to UEM Server communications that are described under FPT\_ITT.1(2). For Android devices, the UEM Server retrieves a specific SCEP challenge for that device from the CA Server. The UEM Server will then bundle the SCEP challenge and the SCEP Server's URL into a payload that is sent to the Android Hub agent. For iOS devices, the UEM Server will request a unique device certificate over DCOM and include the certificate within the MDM profiles (policies) assigned to the device.

[AGENT] Android devices use SCEP to generate a unique client certificate request. After receiving the payload with the SCEP Server's URL and SCEP challenge, the Android Hub agent will send the payload to the device. The device will then request its unique X.509v3 certificate from the SCEP Server. Each iOS device receives its unique X.509v3 certificate within the MDM profiles (policies) assigned to the device. Once a Hub agent receives its unique X.509v3 certificate and the enrollment process is complete, all subsequent communications between a Hub agent and the UEM Server occurs over the connection described under FPT\_ITT.1(2).

## **8.5 Security Management**

### **8.5.1 [MDMPP] FMT\_MOF.1(1)**

[SERVER] The UEM Server provides the capability to manage its own functionality as well as the behavior of mobile devices that are under management. Authorized Administrators manage the TOE through the UEM Server's Admin Console. The Admin Console allows Administrators to specify the unlock banner for both the Admin Console and Self-Service Portal as well as configure the time period for the periodic checks between the UEM Server and the managed devices, the interaction between TOE components, and the certificate for policy signing (as described in FMT\_SMF.1(2)/ANDROID and FMT\_SMF.1(2)/IOS below). Finally, the Admin Console provides the ability to configure how audit data are stored and provides the MAS Server capabilities (as described in FMT\_SMF.1(3) below).

For the configuration of devices under management, the full list of functions that the TSF can perform and a reference to where in the Admin Console this behavior can be found is described in FMT\_SMF.1(1)/ANDROID and FMT\_SMF.1(1)/IOS below. Note that iOS does not provide the ability for third-party MDM software to configure certain aspects of the device's functionality.

As discussed in FMT\_SMR.1(1) below, the TOE provides the ability to define multiple administrative roles, each with its own set of authorized permissions. Individual accounts can be assigned these administrative roles and can also be scoped to only have the authority to manage certain devices or collections of devices based on group membership. If an administrator belongs to a role that has the privilege to perform a certain action and the target for that action is within the scope of their group membership, they are considered to be an Authorized Administrator for the requested function for the purposes of this SFR.

### **8.5.2 [MDMPP] FMT\_MOF.1(2)**

[SERVER] The UEM Server's Authorized Administrator configures the enrollment process. For its own functionality, the Admin Console provides the ability to configure the certificates that are used by the UEM Server as well as the devices that are permitted to enroll in management. This option also allows a maximum number of devices per user to be configured for enrollment.

For both iOS and Android devices, enrollment is performed by a MD user or Authorized Administrator with physical custody of the mobile device. Note that in order to enroll any device, valid user credentials are required which are verified by the UEM Server and/or the external Active Directory/LDAP Server. Since Authorized Administrators are responsible for the creation of MD user accounts on UEM Server, they are able to perform first-use actions requiring user authentication prior to the MD user accessing the account and changing their password.

Note that for iOS, enrollment of mobile devices is brokered using Apple DEP. In the evaluated configuration, the UEM Server is configured to specify the use of registered devices only. Authorized Administrators ensure that iOS mobile devices are first registered with DEP so that they can be selected for enrollment.

### **8.5.3 [MDMPP] FMT\_MOF.1(3)**

[SERVER] The UEM Server provides two methods of restricting the download of apps: through "smart groups" and through the use of allow and deny lists.

The UEM Server uses "smart groups" to separate devices based on how policies are applied to them. Smart groups can consist of organization groups, user groups, and device characteristics. If a device belongs to the smart group, whether it was individually assigned, has characteristics (such as operating system version or model type) associated with the smart group, or is owned by a user who belongs to the group, an app may be configured to be made available to download on demand from the MAS Server functionality of UEM Server. Authorized Administrators have the ability to construct smart groups from these other groups. For any applications that reside in the UEM Server's MAS Server functionality or public applications that are referenced through external links, the Authorized Administrator has the ability to assign one or more smart groups to the app to push it to a set of devices or make it available to be downloaded by them. This assignment can be used to determine if the app is automatically pushed to certain devices based on smart group membership or if it is available on demand.

Apps can also be assigned to application groups by Authorized Administrators. These groups are used to collect apps into a bundle of apps on an allow list, apps on a deny list, or required apps. The categorization of apps can be made both for apps under the control of the UEM Server (i.e. internal apps) as well as publicly-available apps on the Google Play Store/Apple App Store (i.e. managed apps). The application

group is then associated with one or more users and/or organizational groups in order to specify what devices to which it applies. This categorization of apps and subsequent association with managed devices allows the UEM Server to restrict access to internal apps for downloading as well as allows Android Hub agents and iOS/iPadOS platforms to alert the UEM Server if a managed device has one or more internal or managed apps that are in violation of a compliance policy.

The application groups have the following purposes:

- Allow list: If an app is not on the list, it is not permitted on managed devices. When used with a compliance policy, the device's Hub agent will notify the UEM Server if an app that is absent from the allow list is present on the device.
- Deny list: If an app is on this list, it is not allowed on managed devices. When used with a compliance policy, the device's Hub agent will notify the UEM Server if an app that is present on the deny list is present on the device.
- Required: If an app is on this list, it is required MD users install it on managed devices. When used with a compliance policy, the device's Hub agent will notify the UEM Server if an app that is present on the required list is absent from the device.

Additionally, if the Android Hub agent detects an app on a deny list upon the compliance policy being applied to the device (i.e. the app's installation predates the policy), the alerting process immediately remediates the non-compliance by disabling the app. Otherwise, for both the Android Hub agent or the iOS/iPadOS platform, if the deny list compliance policy is applied prior to the installation of the app, the installation of the app is prevented.

Note that an allow list policy and a deny list policy can both be applied to the same device. In this case, the app allow list acts as an exception to apps on the deny list so they can be installed. This occurs when a device is part of multiple smart groups.

#### **8.5.4 [MDMPP] FMT\_POL\_EXT.1**

[SERVER] The UEM Server provides policies (known as "profiles") that can be assigned to groups of devices. Profiles are transmitted to the assigned devices and are applied to the device by one of the following entities: the iOS Hub agent, the iOS/iPadOS underlying platform, or the Android Hub agent. For iOS devices, the entity applying the policy depends on the type of policy being sent. The UEM Server will digitally sign the policies with an X.509v3 certificate, and the signature will be validated by the entity receiving the policy before to the entity applies the policy to the device. All policies are signed by the UEM Server with a trusted CA certificate using ECDSA with SHA-512. Since the devices have a finite set of trusted root certificates that they permit, the Authorized Administrator is expected to acquire certificates from a trusted third-party (e.g., GoDaddy, VeriSign) for signing the policies.

#### **8.5.5 [AGENTMOD] FMT\_POL\_EXT.2**

[AGENT] Candidate profiles are sent to the device by the UEM Server when they are assigned to that device. These profiles have been signed by the UEM Server and require verification before they are applied on the device. For Android devices, all policies are applied by the Android Hub agent and are signed with a trusted CA certificate using ECDSA with SHA-512. For iOS devices, profiles are applied by either the iOS Hub agent or the iOS/iPadOS underlying platform, and the profiles are signed with a trusted CA certificate using ECDSA with SHA-512. The Android Hub agent will use its instance of OpenSSL for

the verification of signed policies for which it will apply to the device. The iOS Hub agent and iOS/iPadOS underlying platform will use the iOS/iPadOS underlying platform for the verification of signed policies for which they will apply to the device. Verification of a policy’s signature is performed as described under the FIA\_X509\_EXT.1(1) and FIA\_X509\_EXT.2 section. The iOS Hub agent, the iOS/iPadOS underlying platform, and the Android Hub agent will only apply a policy that has been signed by the UEM Server’s certificate and that certificate is determined to be valid. If a policy is received that is not signed, signed by an incorrect certificate, or the certificate is deemed invalid, the policy will not be applied on the device.

8.5.6 [MDMPP] FMT\_SMF.1(1)/ANDROID and [MDMPP] FMT\_SMF.1(1)/IOS

[SERVER] The UEM Server component of the TOE has the ability to issue commands and configuration policies to mobile devices. Depending on the mobile device platform and the function being configured, these may be transmitted to the device itself through the platform’s capabilities or to the iOS and Android Hub agent component of the TOE that resides on the device. This is dependent on what APIs the device operating system makes available to third-party applications to access remotely.

The following table lists the management functions that can be performed by the UEM Server as defined by the MDM PP, how those functions are initiated, as well as whether this behavior is enforced by the iOS and Android Hub agent or by the underlying mobile device platform. Unless specified otherwise, the management function is initiated from the “Device details” view in the Admin Console.

Table 20: UEM Server Management Functions

| iOS   |  |                | Android  |                                   |                |
|---|--|----------------|--|-----------------------------------|----------------|
| Command   | Claimed in VID11146 <sup>8</sup> VID11147 <sup>9</sup> | Implemented By | Command  | Claimed in VID11160 <sup>10</sup> | Implemented By |
| <b>1. transition to the locked state</b> – “Lock” button.   | Yes to both  | Platform       | <b>1. transition to the locked state</b> – “Lock” button.  | Yes                               | Hub Agent      |
| <b>2. full wipe of protected data</b> – “More Actions” button > “Device Wipe”.  | Yes to both  | Platform       | <b>2. full wipe of protected data</b> – “More Actions” button > “Device Wipe”.   | Yes                               | Hub Agent      |
| <b>3. unenroll from management</b> – “More Actions” button > “Device Wipe”.   | No to both   | Platform       | <b>3. unenroll from management</b> – “More Actions” button > “Device Wipe”.  | No                                | Hub Agent      |
| <b>4. install policies</b> – assigned and applied to target devices at the creation or modification of a profile under “Devices” > “Profiles & Resources” > “Profiles”. | No to both   | Platform       | <b>4. install policies</b> – assigned and applied to target devices at the creation or modification of a profile under “Devices” > “Profiles & Resources” > “Profiles”.<br>Note: The Server sends the "public app update policy" to Google Play Store which sends the policy to the device for processing. | No                                | Hub Agent      |

<sup>8</sup> TD0479

<sup>9</sup> TD0479

<sup>10</sup> TD0479

|  |             |          |   |     |           |
|--|-------------|----------|---|-----|-----------|
| <b>5. query connectivity status</b> – “Query” button.  | No to both  | Platform | <b>5. query connectivity status</b> – “Query” button.   | No  | Hub Agent |
| <b>6. query the current version of the MD firmware/software</b> – “Query” button. Status shown in the main detail view page.   | No to both  | Platform | <b>6. query the current version of the MD firmware/software</b> – “Query” button. Status shown in the main detail view page.  | No  | Hub Agent |
| <b>7. query the current version of the hardware model of the device</b> – “Query” button. Status shown in the main detail view page.   | No to both  | Platform | <b>7. query the current version of the hardware model of the device</b> – “Query” button. Status shown in the main detail view page.  | No  | Hub Agent |
| <b>8. query the current version of installed mobile applications</b> – “Query” button. Status shown in the Apps tab under the main detail view page.   | No to both  | Platform | <b>8. query the current version of installed mobile applications</b> – “Query” button. Status shown in the “Apps” tab under the main detail view page.  | No  | Hub Agent |
| <b>9. import X.509v3 certificates into the Trust Anchor Database</b> – assigned and applied to devices as part of a policy under the “Credentials” tab when defining the policy.   | Yes to both | Platform | <b>9. import X.509v3 certificates into the Trust Anchor Database</b> – assigned and applied to devices as part of a policy under the “Credentials” tab when defining the policy.  | Yes | Hub Agent |
| <b>10. install applications</b> – “Apps & Books” > “Applications” > “Native”. Admin will be prompted to define what devices an application is assigned to during definition or modification of the application. When the application is specified as automatic distribution, the installation is initiated by the TSF. | Yes to both | Platform | <b>10. install applications</b> – “Apps & Books” > “Applications” > “Native”. Admin will be prompted to define what devices an application is assigned to during definition or modification of the application. When the application is specified as automatic distribution, the installation is initiated by the TSF.<br>Note: This is for internal apps hosted by the TOE. Apps hosted by Google Play Store can also be managed with the "public app update policy". The Server sends this policy to Google Play Store which sends the policy to the device for processing. | Yes | Hub Agent |
| <b>11. update system software</b> – the UEM Server will send command to the iOS/iPadOS platform to update to the identified OS version, then the iOS/iPadOS platform will reach out to Apple to get the OS version.  | Yes to both | Platform | <b>11. update system software</b> – the UEM Server will send command to the Enterprise Firmware Over the Air (EFOTA) ONE Server to update to the latest OS, then the EFOTA ONE Server will push the updated software to the devices.  | Yes | Platform  |
| <b>12. remove applications</b> – specific managed app from a   | Yes to both | Platform | <b>12. remove applications</b> – specific managed app from a  | Yes | Hub Agent |

|  |             |   |  |     |   |
|--|-------------|---|--|-----|---|
| single device: “Device details” view, “Apps” tab, Remove option (“X”) button for the desired managed app.<br>Note: Managed apps are those from the Apple App Store that are installed on the device due to TOE policies.                               |             |   | single device: “Device details” view, “Apps” tab, Remove option (“X”) button for the desired managed app.<br>Note: Managed apps are those from the Google Play Store that are installed on the device due to TOE policies.   |     |   |
| <b>13. remove Enterprise applications</b> – specific internal app from a single device: “Device details” view, “Apps” tab, Remove option (“X”) button for the desired internal app.<br>Note: Internal apps are those from the UEM Server (MAS Server). | Yes to both | Platform                                      | <b>13. remove Enterprise applications</b> – specific internal app from a single device: “Device details” view, “Apps” tab, Remove option (“X”) button for the desired internal app.<br>Note: Internal apps are those from the UEM Server (MAS Server).   | Yes | Hub Agent                                     |
| <b>14. wipe Enterprise data</b> – “More Actions” button > “Enterprise Wipe”.   | Yes to both | Platform                                      | <b>14. wipe Enterprise data</b> – “More Actions” button > “Device Wipe”.   | Yes | Hub Agent                                     |
| <b>15. remove imported X.509v3 certificates</b> – “Devices” > “Profiles & Resources” > “Profiles” > choose the profile that pushed the certificate > “Devices” > “Remove Profile”  | Yes to both | Platform                                      | <b>15. remove imported X.509v3 certificates</b> – “Devices” > “Profiles & Resources” > “Profiles” > choose the profile that pushed the certificate > “Devices” > “Remove Profile”  | Yes | Hub Agent                                     |
| <b>16. alert the user</b> – “Send” button.<br>Note: This refers to alerting the user of the mobile device, not an Administrator on the UEM Server. This can be sent as an email, SMS, or push notification.  | No to both  | Hub Agent (push notification), Platform (SMS) | <b>16. alert the user</b> – “Send” button.<br>Note: This refers to alerting the user of the mobile device, not an Administrator on the UEM Server. This can be sent as an email, SMS, or push notification.  | No  | Hub Agent (push notification), Platform (SMS) |
|  |             |   | <b>20. retrieve MD-software integrity verification values</b> – “Groups & Settings” > “All Settings” > “Apps” > “Settings & Policies” > “Settings” > “Custom Settings”. Paste and save the following in the Custom Settings field:<br><code>{ "SafetyNetEnabled":true }</code><br>Verify SafetyNet from the Summary tab in the “Device details” view.<br>The TOE uses the SafetyNet Attestation API which provides a cryptographically-signed attestation, assessing the device's integrity. | No  | Hub Agent                                     |
| <b>23. revoke Biometric template</b> – by deleting the passcode, this  | No to both  | Platform                                      |  |     |   |

|  |             |          |   |     |           |
|--|-------------|----------|---|-----|-----------|
| disables the biometric template for use.   |             |          |   |     |           |
| <b>25. password policy</b> – defined in the Passcode properties of a profile.  | Yes to both | Platform | <b>25. password policy</b> – defined in the Passcode properties of a profile.   | Yes | Hub Agent |
| <b>26. session locking policy</b> – Defined in the Passcode properties of a profile.   | Yes to both | Platform | <b>26. session locking policy</b> – Defined in the Passcode properties of a profile.  | Yes | Hub Agent |
| <b>27. wireless networks (SSIDs) to which the MD may connect</b> – Defined under the Wi-Fi properties of a profile.  | Yes to both | Platform | <b>27. wireless networks (SSIDs) to which the MD may connect</b> – Defined under the Wi-Fi properties of a profile.   | Yes | Hub Agent |
| <b>28. security policy for each wireless network</b> – defined in the Wi-Fi properties of a profile; the permitted CA(s) are defined by reference on the Wi-Fi properties to those defined under Credentials.            | Yes to both | Platform | <b>28. security policy for each wireless network</b> – defined in the Wi-Fi properties of a profile; the permitted CA(s) are defined by reference on the Wi-Fi properties to those defined under Credentials. | Yes | Hub Agent |
| <b>29. application installation policy</b> – groups of required, allowed, and/or denied apps can be defined in “Apps & Books” > “Application Settings” > “App Groups”.   | Yes to both | Platform | <b>29. application installation policy</b> – groups of required, allowed, and/or denied apps can be defined in “Apps & Books” > “Application Settings” > “App Groups”.  | Yes | Hub Agent |
| <b>30. enable/disable policy for camera and screen capture across device</b> – defined in the Restrictions properties of a profile.  | Yes to both | Platform | <b>30. enable/disable policy for camera and screen capture across device</b> – defined in the Restrictions properties of a profile.   | Yes | Hub Agent |
| <b>31. enable/disable policy for the VPN across MD</b> – defined in the VPN properties of a profile.<br><b>on a per-app basis</b> – defined in the “Tunnel & Other Attributes” setting for an individual app assignment. | Yes to both | Platform | <b>31. enable/disable policy for the VPN across MD or on a per-app basis</b> – defined in the VPN properties of a profile.  | Yes | Hub Agent |
|  |             |          | <b>32. enable/disable policy for Bluetooth</b> – defined in the “Restrictions” tab of a profile.  | Yes | Hub Agent |
|  |             |          | <b>34. enable/disable policy for Wi-Fi tethering, USB tethering, and/or Bluetooth tethering</b> – defined in the “Restrictions” tab of a profile.   | No  | Hub Agent |
| <b>36. enable policy for data-at-rest protection</b> – For iOS devices, data-at-rest protection is automatically enabled if a passcode is set so this is configured under the Passcode properties of a profile.          | No to both  | Platform |   |     |           |
| <b>38. enable/disable policy for local authentication bypass</b> –   | No to both  | Platform | <b>38. enable/disable policy for local authentication bypass</b> –  | Yes | Hub Agent |



|  |             |          |   |     |           |
|--|-------------|----------|---|-----|-----------|
| Deletes the user’s passcode, allowing the user to access the device: “Device” > “Details View” > “More Actions” > “Clear Passcode Device”.   |             |          | Deletes the user’s passcode, allowing the user to access the device: “Device” > “Details View” > “More Actions” > “Change Device Passcode”.   |     |           |
| <b>39. configure the Bluetooth trusted channel policy</b> – Can enable/disable discoverable mode and change the name of the entire device which will change the Bluetooth device name; defined in the “Restrictions” tab of a profile. | No to both  | Platform |   |     |           |
| <b>40. enable/disable policy for display notification in the locked state</b> – can enable/disable any notification on a per-app basis based upon the bundle ID.   | Yes to both | Platform | <b>40. enable/disable policy for display notification in the locked state</b> – can enable/disable all notification through "Allow Keyguard Notifications" defined in the “Restrictions” properties of a profile. | No  | Hub Agent |
| <b>47. the unlock banner policy</b> – configured through the ‘if lost return’ function under “Lock Screen Message” tab of a profile.   | Yes to both | Platform | <b>47. the unlock banner policy</b> – “Set a Lockscreen Message” under the “Custom Messages” properties of a profile.   | Yes | Hub Agent |
| <b>49. enable/disable USB mass storage mode and/or USB data transfer without user-authentication</b> – defined in the “Restrictions” tab of a profile.   | No to both  | Platform | <b>49. enable/disable USB mass storage mode</b> – defined in the “Restrictions” properties of a profile.  | Yes | Hub Agent |
| <b>50. enable/disable backup</b> – defined in the “Restrictions” tab of a profile under the iCloud subcategory.  | No to both  | Platform | <b>50. enable/disable backup</b> – defined in the “Restrictions” tab of a profile.  | No  | Hub Agent |
|  |             |          | <b>51. enable/disable Hotspot and/or USB tethering</b> – select “Allow All Tethering”, defined in the “Restrictions” tab of a profile   | Yes | Hub Agent |
|  |             |          | <b>52. enable/disable location services across device</b> – defined in the “Restrictions” tab of a profile.   | Yes | Hub Agent |
|  |             |          | <b>53. enable/disable policy for user unenrollment</b> – defined in the “Intelligent Hub Settings”.   | No  | Hub Agent |
|  |             |          | <b>54. enable/disable policy for the Always-On VPN protection across device</b> – defined in the VPN properties of a profile.<br>Note: The TOE must configure the VPN to enforce this policy.                     | Yes | Hub Agent |

|  |             |           |  |     |           |
|--|-------------|-----------|--|-----|-----------|
|  |             |           | If the MD user sets the VPN settings this policy will not be enforced.   |     |           |
| <b>55. enable/disable policy for use of Biometric Authentication Factor</b> – defined in the “Restrictions” tab of a profile.  | Yes to both | Platform  | <b>55. enable/disable policy for use of Biometric Authentication Factor</b> – defined in the Passcode properties of a profile.   | Yes | Hub Agent |
|  |             |           | <b>58. enable/disable automatic updates of system software</b> – managed under “Add Profile” > “Android Profile” > “System Updates”.   | No  | Hub Agent |
| <b>60. application installation policy</b> – groups of required, allowed, and/or denied apps can be defined in “Apps & Books” > “Application Settings” > “App Groups”. Note: Command #60 is an extension of Command #29 since allowed applications are only managed by name and not version. | No for both | Platform  | <b>60. application installation policy</b> – groups of required, allowed, and/or denied apps can be defined in “Apps & Books” > “Application Settings” > “App Groups”. Note: Command #60 is an extension of Command #29 since allowed applications are only managed by name and not version. | Yes | Hub Agent |
| <b>61. iOS Hub agent passcode authentication policy</b> – specifying complexity requirements for authenticating to the Hub agent can be defined in “Settings” > “Apps” > “Settings and Policies” > “Security Policies”.  | No for both | Hub Agent |  |     |           |

8.5.7 [MDMPP] FMT\_SMF.1(2)/ANDROID and [MDMPP] FMT\_SMF.1(2)/IOS

[SERVER] The UEM Server provides the ability to manage its own behavior. Listed below are the internal management functions that are provided along with information about how those functions are performed:

- **Configuration of X.509v3 certificates for UEM Server use:** UEM Server utilizes the X.509v3 certificate which is imported into the “Personal” certificate category for Windows Server 2019 Computer account on which UEM Server is installed.
- **Configure devices permitted for enrollment based on:**
  - **Android Devices: devices specified by IMEI or serial number, specific device models, number of devices, manufacturer, and operating system:** Defined in Groups & Settings > All Settings > Devices & Users > General > Enrollment and choosing the Restrictions selection. The Restrictions tab allows for customize enrollment restriction policies.
  - **iOS Devices: devices specified by DEP identifier:** Defined in Devices & Users > General > Enrollment where Current Setting is set to Override and Devices Enrollment Mode is set to Registered Devices Only.
- **Configuration of TOE unlock banner:** Defined for the Admin Console in Settings under System > Branding.

- **Periodicity of agent communications to query connectivity status, query current version of the MD firmware/software, query the current version of the device hardware model, query the current version of installed mobile apps:**
  - Android Devices: Set globally for the supported Android device platform as follows: Groups and Settings > All Settings > Devices and Users > Android > Intelligent Hub Settings
  - iOS Devices: Set globally for the supported iOS device platform as follows: Groups and Settings > All Settings > Devices and Users > Apple > MDM Sample Schedule.
- **Configure the privacy-sensitive information that will and will not be collected from particular mobile devices:** Located in the Admin Console under Groups & Settings > All Settings > Devices & Users > General > Privacy. The types of data categories collected are GPS, Telecom, Applications, Profiles, and Network.
- **Configure the interaction between TOE components:**
  - Android Devices: allow devices based upon IMEI or serial number: Defined in Groups & Settings > All Settings > Devices and Users > General > Enrollment
  - iOS Devices: allow devices based upon DEP identifier: Defined in Devices & Users > General > Enrollment where Current Setting is set to Override and Devices Enrollment Mode is set to Registered Devices Only.
- **Configure server administrator login session timeout:** Set through the Admin Console under Groups & Settings > All Settings > Admin > Console Security > Session Management
- **Configure Enterprise certificate to be used for signing policies:** Set through the Admin Console under Groups & Settings > All Settings > System > Advanced > Policy Signing Certificate
- **Configure MDM Agent/platform to perform a network reachability test:**
  - Android Devices: Set through the Admin Console under Groups & Settings > All Settings > Devices & Users > Android > Intelligent Hub Settings
  - iOS Devices: Set through the Admin Console under Groups & Settings > All Settings > Devices & Users > Apple > MDM Sample Schedule
- **Configure transfer of MDM server logs to another server for storage, analysis, and reporting:** Syslog can be configured either under Groups & Settings > All Settings > System > Enterprise Integration > Syslog. Or by navigating to: Monitor > Reports & Analytics > Events > Syslog (they point to the same location).

#### 8.5.8 [MDMPP] FMT\_SMF.1(3)

[SERVER] The MAS Server component of the UEM Server provides the ability to configure application access in the Admin Console. The MAS Server is defined in the Admin Console under Apps & Books > Applications > Native. Applications can be added to the MAS Server, either as an individual file that is uploaded to the server itself, or as a URL reference to an externally-stored application such as one that resides within the Apple App Store and the Google Play Store. When an application is defined in the MAS Server, it is assigned to one or more smart groups, which are defined under Groups and Settings in the Admin Console. Smart groups can consist of one or more organization groups, user groups, or devices that share certain characteristics regardless of owner. These assignments can be used to define if the application

is automatically downloaded onto the impacted devices or is simply made available by the MAS Server to be downloaded on demand by the user.

The MAS Server also provides the ability to configure application access groups so that different applications can be flagged as required, allowed, or denied. These application groups are assigned a type (required, allowed, denied) and associated with a smart group in the same way that individual applications are assigned. The UEM Server can then define compliance policies to generate alerts when required groups are missing or when denied/not allowed apps are present.

#### **8.5.9 [AGENTMOD] FMT\_SMF\_EXT.4**

[AGENT] The iOS and Android Hub agents have the ability to interact with the underlying mobile device platform in order to enforce the UEM Server management functions. All commands and configuration policies that are defined in FMT\_SMF.1(1)/IOS and FMT\_SMF.1(1)/ANDROID that are received by the iOS and Android Hub agents respectively, result in the mobile device platform being queried or modified in some way. This also includes the ability to upload certificates into the device's certificate store that are used to establish trusted communications between the iOS and Android Hub agents and the UEM Server as part of the enrollment of the device as described under FIA\_X509\_EXT.5. Information about how the iOS and Android Hub agents may enforce these management functions depends on the mobile device platform and is described under FMT\_POL\_EXT.2. The iOS and Android Agents can also prevent the users from unenrolling from management as described under FMT\_UNR\_EXT.1.

The Android Hub agent is configured by the UEM Server to generate periodic reachability events based upon a configured 'sample interval' and 'transmit interval'. The collecting of information on the device by the Android Hub agent occurs every 'sample interval'. The Android Hub agent then queues each sample interval of collected data, and will send up to the last 10 sample intervals of collected data to the UEM Server once the 'transmit interval' is reached.

The iOS Hub agent is also configured by the UEM Server to generate periodic reachability events based upon a configured 'sample interval'. When the 'sample interval' is reached, the iOS Hub agent will initiate a connection to the UEM Server and the UEM Server communicates with the iOS Hub agent to collect policy/sample data from the device. This is considered to be a reachability event since the outcome of this activity updates the Last Seen time of the device in the Admin Console. iOS Hub agent is based upon request/response with the UEM Server, so there is no way for events to be generated when a connection is not made to the UEM Server.

#### **8.5.10 [MDMPP] FMT\_SMR.1(1)**

[SERVER] Roles and permissions are configurable via the Admin Console. Permissions enable and disable specific access to features within the Admin Console and determines if read/write or read-only access is granted to these features. The Admin Console also defines admin groups based on Active Directory/LDAP group information so that individual Administrator accounts can be scoped to only interact with users and/or devices that match their own group membership. When an Administrator is attempting to perform a management function on the TOE, they will not be considered an "Authorized Administrator" unless both of the following are true:

- They are performing an action that is allowed based on the permissions granted to their assigned admin role.

- They are accessing an object that is within the scope of their admin group membership. If the Administrator has no assigned admin group, all objects are within their authorized scope.

While there are several default admin roles, the Admin Console provides the ability for additional admin roles to be created, each with their own set of allowed privileges. Admin group assignment is done at the account level rather than the role level, so two accounts can be assigned the same admin role but belong to different groups. In the evaluated configuration, the TSF will include the following roles:

- **Server primary administrator:** responsible for server installation, initial configuration, and maintenance functions. Responsible for the setup and maintenance of security configuration administrator and auditor accounts.
- **Security configuration administrator:** responsible for security configuration of the server, setup and maintenance of mobile device profiles, definition of user groups, and setup and maintenance of the device user group administrator role, its members, and its permissions.
- **Device user group administrator:** responsible for maintenance of user accounts, including setup, change of account configurations, and account deletion.
- **Auditor:** responsible for review and maintenance of server and device audit logs.

The specific permissions associated with each of these roles are defined in the supplemental administrative guidance for the TOE.

An administrator account may only be assigned one admin role at a time. The “administrator” role as defined by FMT\_SMR.1.1(1) is intended to encompass any of the individual roles listed above.

Users (or “MD users”) are defined separately from administrators. If someone is defined as a user in the Admin Console, they will not be defined as an administrator. A user is simply an individual who possesses a mobile device that is enrolled in management. A user is able to access the Self-Service Portal but not the Admin Console. However, a user can have multiple roles associated with their account but can only be logged into one role at a time. When the user logs in, they can change their role and do not have to log out of the system and re-authenticate to log back into the system.

#### 8.5.11 [MDMPP] FMT\_SMR.1(2)

[SERVER] The MAS Server is logically integrated with the UEM Server. It is accessed by Administrators using the Apps & Books > Applications > Native tab in the Admin Console. Since this is not accessed separately from the remainder of the UEM Server capabilities, the administrative roles that can interact with the MAS Server are defined in the same manner as for FMT\_SMR.1(1) above. The UEM Server also maintains the roles of enrolled mobile devices and application access groups.

#### 8.5.12 [AGENTMOD] FMT\_UNR\_EXT.1

[AGENT] In the evaluated configuration, the "Block User Unenrollment" will be configured for all Android devices which will prevent the unauthorized removal of the Android Hub agent's software from the mobile device. When configured in this manner, the Android Hub agent will perform the following actions to prevent unenrollment:

- Removes the unenrollment button; and
- Removes the ability to uninstall the Android Hub agent through the Google Play Store.

Additionally, as a managed device the Android platform automatically prevents the user from demoting the Android Hub agent from a device Administrator (preventing uninstall).

For the iOS Hub agent, Apple DEP provides the unenrollment protection mechanism for the TOE through the use of the Lock MDM Profile feature. The iOS Hub agent leverages the functionality provided by the underlying device platform, which has been enrolled in Apple DEP, to prevent the unauthorized removal of the iOS Hub agent software.

## **8.6 Protection of the TSF**

### **8.6.1 [MDMPP] FPT\_API\_EXT.1**

[SERVER] The UEM Server uses only the supported Windows Server 2019 platform APIs listed below in order to function.

- .NET API
- Diagnostic API (Windows event logs)
- Networking and Internet API (Microsoft Internet Information Services (IIS))
- Security and Identity API (Windows Authentication)

[AGENT] When installed on a mobile device with the Android platform, the TOE uses only the supported platform APIs listed below in order to function.

- java.security.KeyStore
- Javax.net.HttpURLConnection
- Javax.net.ssl
- KeyChain API (Android Platform Keystore)
- KeyMaster API
- SCrypto
- BoringSSL

When installed on a mobile device with the iOS/iPadOS platforms, the TOE uses only the supported platform APIs listed below in order to function.

- Common Crypto
- CoreCrypto
- Security.framework

### **8.6.2 [MDMPP] FPT\_ITT.1(2)**

[SERVER] The UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and enrolled iOS and Android Hub agents. These protocols are used to protect the data traversing the channel from disclosure and/or modification. This internal channel is used between the UEM Server and the iOS and Android Hub agents for all communications between these TOE components after device enrollment. Since the MAS Server is logically integrated with the UEM Server, the trusted channel to secure its communication with the iOS and Android Hub agents is the same. The UEM Server's platform identity is validated through its X.509v3 certificate presented during TLS session establishment. The UEM Server's platform is invoked by the Hub agent platforms' making a HTTPS/TLS

connection request. During TLS session establishment, the UEM Server's platform will also validate the iOS or Android platform's presented X.509v3 certificate to validate their identities. In the evaluated configuration, the UEM Server's platform will be configured to allow only the following ciphersuites:

- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289

[AGENT] The iOS and Android Hub agents rely on their underlying platforms to provide the HTTPS/TLS communication path and to validate the UEM Server's X.509v3 certificate. The iOS and Android Hub agents' identities are validated through their X.509v3 certificate presented during TLS session establishment. The iOS and Android Hub agents' platforms always initiate this internal channel based upon the iOS/iPadOS platform or Android Hub agent receiving a reachability request to the UEM Server, or an event occurs requiring the iOS Hub agent (e.g. unenrolled from management) or Android Hub agent (e.g. transmit internal) to initiate a connection.

### 8.6.3 [MDMPP] FPT\_LIB\_EXT.1

The TOE is packaged with several third-party open source libraries in order to function.

[SERVER] The UEM Server uses only the listed third-party dynamic libraries for Windows Server 2019 in Appendix A, Section 9.1, in order to function.

[AGENT] When installed on a mobile device with the Android platform, the TOE uses only the listed third-party dynamic libraries in Appendix A, Section 9.2, in order to function.

When installed on a mobile device with the iOS/iPadOS platforms, the TOE uses only the listed third-party libraries in Appendix A, Section 9.3, in order to function.

### 8.6.4 [MDMPP] FPT\_TST\_EXT.1

[SERVER] The UEM Server .dll files and executable code are digitally signed using an X.509v3 certificate from a public CA certificate. During initial installation of the UEM Server and each time the server application is started, the native Windows Authenticode process is invoked to validate the integrity of the UEM Server. When successful, the TOE will start normally. If the validation fails, the native Windows Authenticode process will terminate the host process and the service will not start.

In addition, the UEM Server relies on the underlying Windows Server 2019 platform's own self-tests to verify that the platform and its underlying hardware are operating correctly. This includes the SymCrypt cryptographic module that belongs to the underlying Windows Server 2019 platform. This module performs its own power-up self-tests upon initial start-up, including cryptographic algorithm known answer tests (KATs) and an integrity verification check. For additional information about Windows Server self-tests, refer to the Windows Server 2019 platform Security Target, listed in Section 1.1.5.

These tests are sufficient to validate the correct operation of the TSF because they verify that the platform's cryptographic module which the UEM Server relies upon is operating correctly, the platform does an integrity check of the UEM Server's software, and that the Windows Server 2019 platform

performs self-tests which confirm that the platform's own functionality as well as its underlying hardware's functionality do not have any anomalies that would cause the TOE's software to be executed in an unpredictable or inconsistent manner.

### **8.6.5 [MDMPP] FPT\_TUD\_EXT.1**

[SERVER] Updates for the UEM Server are downloaded as a zip package from the VMware support website. An Authorized Administrator can login to <https://support.workspaceone.com/>, then navigate to Software > Console, or at <https://resources.workspaceone.com/software/console>. The UEM Server software updates are installed by the underlying platform directly onto the system; the platform does not have an automatic method of pulling down or installing the updates without Authorized Administrator (local authorized administrator of the platform) initiation via the platform. The updates are digitally signed using a Digicert X.509v3 certificate which is installed in the Windows trusted key store on the underlying platform which verifies the software updates. The before and after version numbers of the UEM Server software can be checked by clicking on the "About VMware AirWatch" button on the UEM Server's Admin Console. Each iOS and Android Hub agent's current software version can also be queried through the UEM Server's Admin Console by an Authorized Administrator.

[AGENT] Updates to the Hub agents' software are provided by the Google Play Store (Android), Apple Store (iOS), or the UEM Server store over HTTPS/TLS. The Hub agents' software updates are signed using a public CA certificate during the software build and loaded onto the Google Play Store/Apple App Store. The Google Play Store/Apple App Store will then verify the signature and will sign the update with its own signature. The software update is downloaded onto the device by either the MD user (local authorized administrator of the device) directly, the Hub agent after receiving a command from the UEM Server to update the Hub agent software, or the Hub agent based upon a configured policy which requires the Hub agent software to install updates as soon as they are available. Once the update is downloaded onto the device, the platform will verify the signature from the Google Play Store/Apple App Store/UEM Server app store. Secure communication between the mobile device and the stores is handled by the underlying platform for each Hub agent.

## **8.7 TOE Access**

### **8.7.1 [MDMPP] FTA\_TAB.1**

[SERVER] The UEM Server supports the ability to display a configurable warning banner on the Admin Console and the Self-Service Portal login pages. The warning banner will be displayed to both Administrators and users prior to authenticating to the UEM Server on their respective interfaces. The warning banner can be configured through the Admin Console Warning Banner tab by an Authorized Administrator.

## **8.8 Trusted Path/Channels**

### **8.8.1 [MDMPP] FTP\_ITC\_EXT.1**

[SERVER] The TOE has a communication channel between the UEM Server and one or more iOS and Android Hub agents on mobile devices. The UEM Server relies on its underlying platform to protect all data from disclosure and modification transferred over this internal communication channel. This



communication channel is established once a mobile device has been fully enrolled into management, is logically distinct from other communication channels, and is specified under FPT\_ITT.1(2).

[AGENT] The iOS and Android Hub agents are internal to the TOE. The iOS and Android Hub agents rely on their underlying platforms to protect all data from disclosure and modification transferred over this internal communication channel.

### 8.8.2 [MDMPP] FTP\_ITC.1(1)

[SERVER] The UEM Server communicates with third-party systems that reside in the Operational Environment via trusted channels. In the evaluated configuration, the UEM Server connects with:

- the Syslog Audit Server using TLS v1.2 to encrypt the audit data that traverses the channel, and
- the AD/LDAP authentication server using TLS v1.2 for device enrollment using LDAP and to send authentication requests for an Administrator attempting to authenticate to the Admin Console.

The use of these protocols to establish trusted channels ensures that data in transit will be protected and not subjected to unauthorized modification or disclosure. During TLS session establishment, the UEM Server's platform will validate the third-party systems' presented X.509v3 certificates to validate their identities. If the third-party system is configured for mutual authentication, the UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment. In the evaluated configuration, the UEM Server's platform will be configured to allow only the following ciphersuites:

- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289

The MAS Server is logically integrated with the UEM Server. As a result of this, all remote communications that the MAS Server requires to operate are identical to those described above.

### 8.8.3 [MDMPP] FTP\_TRP.1(1)

[SERVER] The UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and Administrators attempting to connect to the Admin Console for the purposes of remote administration. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the Administrator's identity is validated through their authentication credentials presented to the UEM Server. In the evaluated configuration, the UEM Server's platform will be configured to allow only the following ciphersuites:

- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289

**8.8.4 [MDMPP] FTP\_TRP.1(2)**

[SERVER] The UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and users. These protocols are used to protect the data traversing the channel from disclosure and/or modification. This communication path is used to connect to the Self-Service Portal for the purposes of remote device registration and other self-service tasks as well as the enrollment of the iOS and Android Hub agents into the TOE. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the user's identity is validated through their authentication credentials presented to the UEM Server. In the evaluated configuration, the UEM Server's platform will be configured to allow only the following ciphersuites:

- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289

[AGENT] The iOS and Android Hub agents rely on their underlying platforms to provide the HTTPS/TLS communication path and to validate the UEM Server's X.509v3 certificate. After the enrollment of an iOS or Android Hub agent into the TOE is complete, the initial connection handled by the communication path described above is closed and all future communication between the Hub agent and UEM Server is governed by the internal channel described under the FPT\_ITT.1(2) requirement.

## 9 Appendix A: List of Third-Party Libraries

### 9.1 Windows Server 2019 Libraries

- adaptivasdk
- antlr
- AspectCompiler
- Automapper
- Automatonymous
- bundletransformer.autoprefixer
- bundletransformer.core
- bundletransformer.less
- cete.dynamicpdf
- commandlineparser
- commonservicelocator
- componentspace.saml2
- coverlet.collector
- dapper
- dotless
- DotNetSeleniumExtras.PageObjects
- DotNetSeleniumExtras.WaitHelpers
- edtftpnet-pro
- EgressTenantAdapter.Client
- enterpriselibrary.common
- enterpriselibrary.data
- enterpriselibrary.data.netcore
- entrustv9
- enyimmemcachedcore
- eplusplus
- exceldatareader
- exceldatareader.dataset
- fluentassertions
- fluentvalidation
- google.apis
- google.apis.admin.directory.directory\_v1
- Google.apis.androidenterprise.v1
- google.apis.auth
- google.protobuf
- google.protobuf.tools
- htmlagilitypack
- htmsanitizer
- http2dotnet
- I18NClient.Net
- I18NClient.Net.Abstractions
- identityguardadminservicev8api
- identityguardcommonfailoverapi
- IdentityModel.OidcClient
- interop.cert
- javascriptengineswitcher.v8
- jetbrains.microsoft.deployment.windows.installer
- JsonSubTypes
- justmock
- marvin.jsonpatch
- microsoft.aspnet.mvc
- microsoft.aspnet.razor
- microsoft.aspnet.web.optimization
- microsoft.aspnet.webapi
- microsoft.aspnet.webapi.client
- microsoft.aspnet.webapi.core
- microsoft.aspnet.webapi.cors
- microsoft.aspnet.webapi.owin
- microsoft.aspnet.webapi.owinselfhost
- microsoft.aspnet.webapi.webhost
- microsoft.aspnet.webpages
- microsoft.aspnetcore.http
- microsoft.azure.activedirectory.graphclient
- microsoft.clearscript.v8.native.win-x64
- microsoft.codeanalysis.csharp.scripting
- microsoft.codedom.providers.dotnetcompilerplatform
- microsoft.csharp
- microsoft.deployment.compression
- microsoft.extensions.caching.abstractions
- microsoft.extensions.caching.memory
- microsoft.extensions.dependencyinjection.abstractions
- microsoft.extensions.http
- microsoft.extensions.logging

## *Security Target*

- microsoft.extensions.logging.abstractions
- microsoft.extensions.options
- microsoft.graph
- microsoft.identitymodel.clients.activedirectory
- microsoft.identitymodel.protocols.openidconnect
- microsoft.net.test.sdk
- microsoft.owin
- microsoft.owin.host.httplistener
- microsoft.owin.host.systemweb
- microsoft.owin.hosting
- microsoft.owin.security.jwt
- microsoft.powershell.5.referenceassemblies
- microsoft.sharepointonline.csom
- microsoft.sqlserver.dacfx.x64
- microsoft.visualstudio.threading
- microsoft.visualstudio.threading.analyzers
- microsoft.web.administration
- microsoft.web.infrastructure
- microsoft.wim
- mimekit
- moq
- moq.automock
- mraspect.fody
- netstandard.library
- newtonsoft.json
- newtonsoft.json.bson
- nlog.targets.syslog
- node.js
- nsdepcop
- Polly
- Polly.Contrib.WaitAndRetry
- Portable.BouncyCastle
- qrcoder
- quartz
- quartz.serialization.json
- restease
- restsharp

## *VMware Workspace ONE UEM*

- restsharp.newtonsoft.json.extensions
- security.cryptography
- selenium.phantomjs.webdriver
- selenium.support
- selenium.webdriver
- selenium.webdriver.chromedriver
- selenium.webdriver.geckodriver
- Serilog.Sinks.Console
- Serilog.Sinks.File
- Sharpziplib
- SigningService.Api.Client
- simmetrics.net
- sonaranalyzer.csharp
- specflow
- specflow.tools.msbuild.generation
- specflow.xunit
- StackExchange.Redis
- starksoft.aspen
- stylecop.analyzers
- swashbuckle
- swashbuckle.core
- system.buffer
- system.codedom
- system.componentmodel
- system.componentmodel.annotations
- system.configuration.configurationmanager
- system.data.odbc
- system.data.oledb
- system.diagnostics.tracing
- system.directoryservices.protocols
- system.identitymodel.tokens.jwt
- system.io.abstractions
- system.io.abstractions.testinghelpers
- system.io.filesystem.accesscontrol
- system.management
- system.management.automation
- system.memory
- system.numerics.vectors
- system.reactive.platformservices
- system.reflection.typeextensions
- system.runtime.caching

## *Security Target*

- system.runtime.compilerservices.unsafe
- system.runtime.serialization.plists
- system.security.cryptography.cng
- system.security.cryptography.pkcs
- system.security.cryptography.x509certificates
- system.servicemodel.http
- system.servicemodel.primitives
- system.serviceprocess.servicecontroller
- system.text.encodings.web
- system.threading.tasks.dataflow
- system.threading.tasks.extensions
- system.valuetuple
- ThirdPartyApi.Client
- twilio
- unity

## **9.2 Android 11 Libraries**

- acsdk.jar
- androidx.core:core-ktx
- androidx.multidex:multidex
- apache-commons-net
- AppAuth-Android
- chilkat-ftp-software
- com.airbnb.android:lottie
- com.android.support:multidex
- com.crashlytics.sdk.android
- com.commonware.cwac:saferoom
- com.fasterxml.jackson.core:jackson-core
- com.fasterxml.jackson.core:jackson-databind
- com.fasterxml.jackson.core:jackson-annotations
- com.google.android.play:core
- com.google.code.gson:gson
- com.google.dagger
- com.madgag.spongycastle:core
- com.microsoft.identity.client:msal
- com.mixpanel.android
- com.squareup.moshi:moshi
- com.squareup.moshi:moshi-kotlin
- com.squareup.moshi:moshi-adapters

## *VMware Workspace ONE UEM*

- unity.abstractions
- unity.container
- unity.interception
- unity.registrationbyconvention
- unity.servicelocation
- webapithrottle
- webgrease
- wiremock.net
- wix
- xunit
- xunit.analyzers
- xunit.core
- xunit.runner.msbuild
- xunit.runner.visualstudio
- YamlDotNet

- com.squareup.moshi:moshi-kotlin-codegen
- com.squareup.okhttp3:logging-interceptor
- com.squareup.okhttp3:okhttp
- com.squareup.picasso
- com.squareup.retrofit2:retrofit
- com.squareup.retrofit2:converter-jackson
- cz.msebera.android:httpclient
- dev.samstevens.totp:totp
- cz.msebera.android:httpclient
- Google-DPC-lib
- google-play-services
- gson
- guava
- io.reactivex.rxjava2:rxandroid
- io.reactivex.rxjava2:rxjava
- io.supercharge:shimmerlayout
- jzlib
- kotlin-stdlib-jdk8
- krb5
- littleproxy
- netty
- net.minidev:json-smart

## *Security Target*

- OpenSSL
- org.apache.commons:io
- org.greenrobot:eventbus
- org.greenrobot:greendao
- org.jetbrains.kotlin:kotlin-reflect
- org.jetbrains.kotlin:kotlin-coroutines-android

## **9.3 iOS 14 and iPadOS 14 Libraries**

- Alamofire
- AppAuth
- Base32
- BerTlv
- Boringssl
- CocoaLumberjack
- Duktpe
- FastSocket
- GoogleApiClientForRest
- GoogleAppMeasurement
- GoogleDataTransport
- GoogleSignIn
- GTMAppAuth
- GTMSessionFetcher
- HttpStatusCodes
- JSONAPI
- Jsoncpp

## *VMware Workspace ONE UEM*

- org.jetbrains.kotlin:kotlin-coroutines-core
- pub.devrel:easypermissions
- slf4j
- sqlcipher-for-android
- support-libraries-for-android-api
- zxing

- JsonModel
- JWTDecode
- KeychainAccess
- kissxml
- lottie
- MSAL
- OneTimePassword
- OpenSSL
- plcrashreporter
- reachability
- sfhfkeychainutils
- Starscream
- Sqlcipher
- SQLite3
- SwiftPtoBuf
- Zip