MobileIron Platform (MDMPP20 and MDMAEP20) Security Target

Version 1.0
05/27/16

Prepared for:

MobileIron

415 East Middlefield Road
Mountain View, CA 94043

Prepared By:

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is a MobileIron Platform provided by MobileIron. The TOE is being evaluated as a Mobile Device Management (MDM) solution consisting of a MDM server (MobileIron Core) and associated MDM mobile device agents (MobileIron Client).

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a iteration number placed at the end of the component label. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2). Note also that unlike the PP, the elements labels include the iteration value after the component portion of the label rather than at the end of the label (e.g., FDP_ACC.1(1).1 rather than FDP_ACC.1.1(1)).
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – MobileIron Platform (MDMPP20 and MDMAEP20) Security Target

ST Version – Version 1.0

ST Date – 05/27/16

1.2 TOE Reference

TOE Identification – MobileIron Platform

- MobileIron Core, Version 9.0
- MobileIron Client – Mobile@Work for Android, Version 8.6

TOE Developer – MobileIron

Evaluation Sponsor – MobileIron

1.3 TOE Overview

The Target of Evaluation (TOE) the MobileIron Core server and associated MobileIron Client (referred to as Mobile@Work) agents for Android devices that are part of their MobileIron Platform. Note that the MobileIron Platform consists of other components (MobileIron Sentry and additional mobile device applications, e.g., Web@work, Docs@work, AppConnect container and Secure Application Manager) that do not play a role in enforcing the security functions included in this Security Target.

1.4 TOE Description

The TOE is an MDM solution where the claimed security functions are implemented in a central MDM server – MobileIron Core - and distributed MDM agents – MobileIron Client.

MobileIron Core (http://www.mobileiron.com/en/products/core) integrates with backend enterprise IT systems and enables IT to define security and management policies for mobile apps, content and devices independent of the operating system. MobileIron Core enables mobile device, application, and content management.

- Mobile device management capabilities are the primary focus of this evaluation and enable IT to securely manage mobile devices across mobile operating systems and provide secure corporate email, automatic device configuration, certificate-based security, and selective wiping of enterprise data from both corporate-owned as well as user-owned devices.
- Mobile application management capabilities are a secondary focus of this evaluation and help IT manage the entire application lifecycle, from making the applications available in the enterprise app storefront, facilitating deployment of applications to mobile devices, and retiring applications as necessary. Note that this capability is referred to as MAS – Mobile Application Store – Server later in this ST.
- Mobile content management functions are included in the MobileIron Platform, but no claims are made about those capabilities in this Security Target.

MobileIron Client (http://www.mobileiron.com/en/products/client) – also known as Mobile@Work – is an app downloaded by end users onto their mobile devices. It automatically configures the device to function in an enterprise environment by enforcing the configuration and security policies set by the IT department. Once installed, it creates a secure MobileIron container to protect enterprise data and applications.

- The MobileIron Client works with MobileIron Core to configure corporate email, Wi-Fi, VPN, and security certificates and to create a clear separation between personal and business information. This allows IT to selectively wipe only the corporate data on the device if the user leaves the company or if the device falls out of compliance or is lost.
- The MobileIron Client also enables additional enterprise device controls that are not subject to security claims and hence are outside the scope of the evaluation related to this Security Target.

1.4.1 TOE Architecture

MobileIron offers a MobileIron Platform solution comprised of MobileIron Core, MobileIron Sentry, MobileIron Client, and mobile end user products (e.g., smartphones).

Of these components, the TOE is a central MobileIron Core server and MobileIron Clients installed on distributed end user mobile Android devices. MobileIron Sentry is another security product not within scope of this evaluation

(i.e., the MDMPP20 requirements are not applicable), but can freely be used with the TOE in its evaluated configuration.

1.4.1.1 Physical Boundaries

As indicated above the TOE consists of two software components: MobileIron Core and MobileIron Client.

MobileIron Core is a server that is deployed on a CentOS 6.7 Linux operating system (OS) that is included (preconfigured) with the product distribution and runs on an Intel x64 architecture server platform. MobileIron supports the MobileIron Core operating on one of three physical server appliances that they distribute (Mobile Iron M2100, M2200, or M2500) as well as virtual deployments in VMWare ESXi (5.1, 5.5, or 6.0) and Microsoft Hyper-V (Server 2008 R2 or Server 2012 R2). Later in this ST the MobileIron Core component is identified as the "MDM server" and its underlying OS is identified as the "MDM server platform".

The Mobile Iron appliances are all based on Intel Xeon CPUs (E3-1200 or E5-2670) and utilize Intel network adapters (82579LM GbE, 82574L GbE, or Quad I350 GbE) along with SATA disk drives and DRAM from 32-64 GB. Virtual deployments must use a 64-bit x86 virtual machine and an E1000 network adapter. The vendor recommends at least 8 Gb RAM and 80 Gb storage.

MobileIron Client consists of apps deployed on Android mobile devices. These components are identified later in this ST as the "MDM Agent".

There are a number of evaluated Samsung Galaxy mobile Android devices (including Galaxy S4, Galaxy Note 3, Galaxy S5, Galaxy Note 4, and Galaxy S6 models) ranging from Android version 4.4 to 5.0.2 that can be used with the Android version of the MobileIron Client.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by MobileIron Platform:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The MDM Server can generate and store audit records for security-relevant events as they occur. These events are stored and protected by the MDM Server and can be reviewed by an authorized administrator. The MDM Server can be configured to export the audit records in either in CSV (comma separated values) format, text format, or a compressed archive format utilizing TLS for protection of the records on the network. The MDM Server also supports the ability to query information about MDM agents and export MDM configuration information.

The MDM Agent includes the ability to indicate (i.e., respond) when it has been enrolled and when policies are applied to the MDM Agent. The MDM Server can be configured to alert an administrator based on its configuration. For example, it can be configured to alert he administrator when a policy update fails or an MDM Agent has been enrolled.

1.4.1.2.2 Cryptographic support

The MDM Server and MDM Agent both include and/or utilize cryptographic modules or libraries with National Voluntary Laboratory Accreditation Program (NVLAP) Cryptographic Algorithm Validation Program (CAVP) validated algorithms for a wide range of cryptographic functions including: asymmetric key generation and establishment, encryption/decryption, cryptographic hashing and keyed-hash message authentication. These

functions are supported with suitable random bit generation, initialization vector generation, secure key storage, and key and protected data destruction.

The primitive cryptographic functions are used to implement security communication protocols: TLS, HTTPS, and SSH used for communication between the MDM Server and MDM Agent and between the MDM Server and remote administrators.

1.4.1.2.3 Identification and authentication

The MDM Server requires mobile device users (MD users) and administrators to be authenticated prior to allowing any security-related functions to be performed. This includes MD users enrolling their device in the MDM Server using a corresponding MDM Agent as well as an administrator logging on to manage the MDM Server configuration, MDM policies for mobile devices, etc.

In addition, both the MDM Server and MDM Agent utilize X.509 certificates, including certificate validation checking, in conjunction with TLS to secure communications between the MDM Server and MDM Agents as well as between the MDM Server and administrators using a web-based user interface for remote administrative access.

1.4.1.2.4 Security management

The MDM Server is designed to include at least two distinct user roles: administrator and mobile device user (MD user). The former interacts directly with the MDM Server while the latter is the user of a mobile device hosting an MDM Agent. The MDM Server further supports the fine-grain assignment of role (access to management function) to defined users allowing the definition of multiple user and administrator roles with different capabilities and responsibilities.

The MDM Server provides all the function necessary to manage its own security functions as well as to manage mobile device policies that are sent to MDM Agents. In addition, the MDM Server ensures that security management functions are limited to authorized administrators while allowing MD users to perform only necessary functions such as enrolling in the MDM Server.

The MDM Agents provide the functions necessary to securely communicate with and enroll in a MDM Server, implement policies received from and enrolled MDM Server, and report the results of applying policies.

1.4.1.2.5 Protection of the TSF

The MDM Server and MDM Agent work together to ensure that all security related communication between those components is protected from disclosure and modification.

Both the MDM Server and MDM Agent include self-testing capabilities to ensure that they are functioning properly. The MDM Server also has the ability to cryptographically verify during start-up that its executable image has not been corrupted.

The MDM Server also includes mechanisms (i.e., verification of the digital signature of each new image) so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

1.4.1.2.6 TOE access

The MDM Server has the capability to display an advisory banner when users attempt to login in order to manage the TOE using the web-based and command-line based user interfaces.

1.4.1.2.7 Trusted path/channels

The MDM Server uses TLS/HTTPS to secure communication channels between itself and remote administrators accessing the TOE via a web-based user interface. In addition, the MDM Server implements a restricted shell (CLISH) that is accessible via SSH to provide access to low level management functions.

It also uses TLS to secure communication channels between itself and mobile device users (MD users). In this latter case, the protected communication channel is established between the MDM Server and applicable MDM Agent on the user's mobile device.

1.4.2 TOE Documentation

MobileIron has developed an extensive document set for the MobileIron Platform. However, for the purpose of evaluation, the following guides were examined:

- MobileIron® MobileIron Core and Android Client Mobile Device Management Protection Profile Guide, Document version 1.2, May 4, 2016
- MobileIron® On-Premise Installation Guide For MobileIron Core, Sentry, and Enterprise Connector, Core Version 9.0, Revised March 21, 2016
- MobileIron Core 9.0.0.0 Upgrade Guide, March 29, 2016.
- MobileIron® Core Device Management Guide For Android Devices, Core 9.0, Revision March 21, 2016
- MobileIron® Apps@Work Guide, MobileIron Core version 9.0, February 29, 2016
- MobileIron® Core System Manager Guide, Core version 9.0, February 26, 2016
- MobileIron® Core Command Line Interface (CLI) Reference, Core Version 9.0, February 26, 2016
- MobileIron® Core Delegated Administration, Version 9.0, February 26, 2016

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Package Claims:
    - Protection Profile for Mobile Device Management, Version 2.0, 31 December 2014 (MDMPP20)
    - Extended Package for Mobile Device Management Agents, Version 2.0, 31 December 2014 (MDMAEP20)

2.1 Conformance Rationale

The ST conforms to the MDMPP20 and MDMAEP20. As explained below, the security problem definition, security objectives, and security requirements have been drawn from the MDMPP20 and MDMAEP20.

3. Security Objectives

The Security Problem Definition can be found in the Protection Profile for Mobile Device Management, Version 2.0, 31 December 2014 (MDMPP20) and Extended Package for Mobile Device Management Agents, Version 2.0, 31 December 2014 (MDMAEP20) and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The MDMPP20 and MDMAEP20 offers additional information about the identified security objectives, but that has not been reproduced here and the MDMPP20 and MDMAEP20 should be consulted if there is interest in that material.

In general, the MDMPP20 and MDMAEP20 have defined Security Objectives appropriate for a Mobile Device Management (MDM) product and as such are applicable to the MobileIron Platform TOE.

3.1 Security Objectives for the Environment

OE.IT_ENTERPRISE

The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access.

OE.MDM_SERVER_PLATFORM

The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities.

OE.MOBILE_DEVICE_PLATFORM

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.

OE.PROPER_ADMIN

TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.PROPER_USER

Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.

OE.WIRELESS_NETWORK

A wireless network will be available to the mobile devices.

OE.TIMESTAMP

Reliable timestamp is provided by the operational environment for the TOE.

4. Extended Components Definition

All of the extended requirements in this Security Target (ST) have been drawn from the Protection Profile for Mobile Device Management, Version 2.0, 31 December 2014 (MDMPP20) and Extended Package for Mobile Device Management Agents, Version 2.0, 31 December 2014 (MDMAEP20). The MDMPP20 and MDMAEP20 define the following extended requirements and since they are not redefined in this ST the MDMPP20 and MDMAEP20 should be consulted for more information in regard to those Common Criteria (CC) extensions.

Extended SFRs:

- FAU_ALT_EXT.1: Server Alerts
- FAU_ALT_EXT.2: Agent Alerts
- FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration
- FAU_NET_EXT.1: Network Reachability Review
- FAU_STG_EXT.1: External Audit Trail Storage
- FAU_STG_EXT.2: Audit Event Storage
- FCS_CKM_EXT.4: Cryptographic Key Destruction
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_IV_EXT.1: Initialization Vector Generation
- FCS_RBG_EXT.1: Extended: Random Bit Generation
- FCS_SSHS_EXT.1: SSH Protocol
- FCS_STG_EXT.1: Cryptographic Key Storage
- FCS_STG_EXT.2: Encrypted Cryptographic Key Storage
- FCS_STG_EXT.4: Cryptographic Key Storage
- FCS_TLSC_EXT.1: Cryptographic Support (FCS)
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
- FIA_ENR_EXT.2: Enrollment of Mobile Device into Management
- FIA_X509_EXT.1: X509 Validation
- FIA_X509_EXT.2: X509 Authentication
- FMT_SMF_EXT.3: Specification of Management Functions
- FMT_UNR_EXT.1: User Unenrollment Prevention
- FPT_TST_EXT.1: Protection of the TSF (FPT)
- FPT_TUD_EXT.1: Trusted Update

5. Security Requirements

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

The SFRs have all been drawn from the Protection Profile for Mobile Device Management, Version 2.0, 31 December 2014 (MDMPP20) and Extended Package for Mobile Device Management Agents, Version 2.0, 31 December 2014 (MDMAEP20). The refinements and operations already performed in the MDMPP20 and MDMAEP20 are not identified (e.g., highlighted) here, rather the requirements have been copied from the MDMPP20 and MDMAEP20 and any residual operations have been completed herein. Of particular note, the MDMPP20 and MDMAEP20 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the MDMPP20 and MDMAEP20. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the MDMPP20 and MDMAEP20 that serve to ensure corresponding evaluations will yield more practical and consistent assurance. The MDMPP20 and MDMAEP20 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by MobileIron Platform TOE.

Requirement Class	Requirement Component	MDM Component
FAU: Security audit	FAU_ALT_EXT.1: Server Alerts	Server
	FAU_ALT_EXT.2: Agent Alerts	Agent/Server
	FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration	Server
	FAU_GEN.1(1): Audit Data Generation	Server
	FAU_NET_EXT.1: Network Reachability Review	Server
	FAU_SAR.1: Audit Review	Server
	FAU_STG_EXT.1: External Audit Trail Storage	Server
	FAU_STG_EXT.2: Audit Event Storage	Server
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation	Server/Server Platform
	FCS_CKM.1(1): Cryptographic Key Generation	Agent
	FCS_CKM.2: Cryptographic Key Establishment	Server/Server Platform
	FCS_CKM.2(1): Cryptographic Key Establishment	Agent
	FCS_CKM_EXT.4: Cryptographic Key Destruction	Server
	FCS_CKM_EXT.4(1): Cryptographic Key Destruction	Agent
	FCS_COP.1(1): Cryptographic Operation (Confidentiality Algorithms)	Server/Server Platform
	FCS_COP.1(2): Cryptographic Operation (Hashing)	Server/Server Platform
	FCS_COP.1(3): Cryptographic Operation (Digital Signature)	Server/Server Platform
	FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)	Server/Server Platform
	FCS_COP.1(5): Cryptographic Operation (Confidentiality Algorithms)	Agent
	FCS_COP.1(6): Cryptographic Operation (Hashing)	Agent

Requirement Class	Requirement Component	MDM Component
	FCS_COP.1(7): Cryptographic Operation (Digital Signature)	Agent
	FCS_COP.1(8): Cryptographic Operation (Keyed-Hash Message Authentication)	Agent
	FCS_HTTPS_EXT.1: HTTPS Protocol	Server/Server Platform
	FCS_IV_EXT.1: Initialization Vector Generation	Server
	FCS_RBG_EXT.1: Extended: Random Bit Generation	Server/Server Platform
	FCS_RBG_EXT.1(1): Extended: Random Bit Generation	Agent
	FCS_SSHS_EXT.1: SSH Protocol	Server Platform
	FCS_STG_EXT.1: Cryptographic Key Storage	Server
	FCS_STG_EXT.2: Encrypted Cryptographic Key Storage	Server
	FCS_STG_EXT.4: Cryptographic Key Storage	Agent
	FCS_TLSC_EXT.1: Cryptographic Support (FCS)	Agent/Server
	FCS_TLSS_EXT.1: TLS Server Protocol	Server
FIA: Identification and authentication	FIA_ENR_EXT.1: Enrollment of Mobile Device into Management	Server
	FIA_ENR_EXT.2: Enrollment of Mobile Device into Management	Agent
	FIA_UAU.1: Timing of Authentication	Server
	FIA_X509_EXT.1: X509 Validation	Server
	FIA_X509_EXT.1(1): X509 Validation	Agent
	FIA_X509_EXT.2: X509 Authentication	Server
	FIA_X509_EXT.2(1): X509 Authentication	Agent
FMT: Security management	FMT_MOF.1(1): Management of Functions in MDM Server	Server
	FMT_MOF.1(2): Management of Enrollment Function	Server
	FMT_MOF.1(3): Management of Functions in MAS Server	Server
	FMT_MOF.1(4): Management of Download Function in MAS Server	Server
	FMT_SMF.1(1): Specification of Management Functions (Server configuration of Agent)	Server
	FMT_SMF.1(2): Specification of Management Functions (Server Configuration of Server)	Server
	FMT_SMF.1(3): Specification of Management Functions (MAS Server)	Server
	FMT_SMF_EXT.3: Specification of Management Functions	Agent
	FMT_SMR.1(1): Security Management Roles	Server
	FMT_SMR.1(2): Security Management Roles	Server
	FMT_UNR_EXT.1: User Unenrollment Prevention	Agent
FPT: Protection of the TSF	FPT_ITT.1(1): Basic Internal TSF Data Transfer Protection (MDM Server)	Agent/Server
	FPT_ITT.1(2): Basic Internal TSF Data Transfer Protection (Distributed TOE)	Agent/Server
	FPT_ITT.1(3): Basic Internal TSF Data Transfer Protection (MAS Server)	Agent/Server
	FPT_TST_EXT.1: TSF Testing	Server
	FPT_TST_EXT.1(1): TSF Testing	Agent
	FPT_TUD_EXT.1: Trusted Update	Server
FTA: TOE access	FTA_TAB.1: Default TOE Access Banners	Server
FTP: Trusted path/channels	FTP_ITC.1(1): Inter-TSF Trusted Channel (Authorized IT Entities)	Server
	FTP_ITC.1(3): Inter-TSF Trusted Channel (Authorized IT Entities)	Server
	FTP_TRP.1(1): Trusted Path for Remote Administration	Server
	FTP_TRP.1(2): Trusted Path for Enrollment	Server

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Server Alerts (FAU_ALT_EXT.1)

FAU_ALT_EXT.1.1

The MDM Server 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. [***no other events***].

5.1.1.2 Agent Alerts (FAU_ALT_EXT.2)

FAU_ALT_EXT.2.1

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

a. successful application of policies to a mobile device;
b. [***generating***] periodic reachability events;
[***c. change in enrollment state***].

FAU_ALT_EXT.2.2

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

5.1.1.3 Support for Compliance Reporting of Mobile Device Configuration (FAU_CRP_EXT.1)

FAU_CRP_EXT.1.1

The MDM Server shall provide [***an interface that provides responses to queries about the configuration of enrolled devices, an interface that permits the export of data about the configuration of enrolled devices***] to authorized entities over an authenticated [***TLS/HTTPS***] trusted communication channel. The provided information for each enrolled mobile device includes:

a. The current version of the MD firmware/software
b. The current version of the hardware model of the device
c. The current version of installed mobile applications
d. List of MD configuration policies that are in place on the device (as defined in FMT_SMF.1.1(1))
e. [***no other information***].

5.1.1.4 Audit Data Generation (FAU_GEN.1(1))

FAU_GEN.1(1).1

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

a. Start-up and shutdown of the MDM Server software;
b. All administrative actions;
c. Commands issued from the MDM Server to an MDM Agent;
d. Specifically defined auditable events listed in **Table 2 Auditable Events**; and
f. [**MDM agent alerts generated by FAU_ALT_EXT.2.1**].

Requirement	Auditable Events	Additional Content
FAU_ALT_EXT.1	Type of alert.	Identity of Mobile Device that sent alert.
FAU_ALT_EXT.2	None	None
FAU_CRP_EXT.1	None	None

FAU_GEN.1(1)	None	None
FAU_NET_EXT.1	None	None
FAU_SAR.1	None	None
FAU_STG_EXT.1	None	None
FAU_STG_EXT.2	None	None
FCS_CKM.1	Failure of key generation activity for authentication keys.	None
FCS_CKM.1(1)	None	None
FCS_CKM.2	None	None
FCS_CKM.2(1)	None	None
FCS_CKM_EXT.4	None	None
FCS_CKM_EXT.4(1)	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_COP.1(5)	None	None
FCS_COP.1(6)	None	None
FCS_COP.1(7)	None	None
FCS_COP.1(8)	None	None
FCS_HTTPS_EXT.1	Failure of the certificate validity check.	None
FCS_IV_EXT.1	None	None
FCS_RBG_EXT.1	Failure of the randomization process.	None
FCS_RBG_EXT.1(1)	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure. Non-TOE endpoint of connection.
FCS_STG_EXT.1	None	None
FCS_STG_EXT.2	None	None
FCS_STG_EXT.4	None	None
FCS_TLSC_EXT.1	Failure to establish a TLS session. Failure to verify presented identifier.	Reason for failure. Presented identifier and reference identifier.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_ENR_EXT.1	Failure of MD user authentication.	Presented credentials.
FIA_ENR_EXT.2	None	None
FIA_UAU.1	None	None
FIA_X509_EXT.1	Failure to validate X.509 certificate.	Reason for failure.
FIA_X509_EXT.1(1)	None	None
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	None
FIA_X509_EXT.2(1)	None	None
FMT_MOF.1(1)	Issuance of command to perform function. Change of policy settings.	Command sent and identity of MDM Agent recipient. Query responses. Policy changed and value or full policy.
FMT_MOF.1(2)	Enrollment by a user.	Identity of user.
FMT_MOF.1(3)	None	None
FMT_MOF.1(4)	None	None
FMT_SMF.1(1)	None	None
FMT_SMF.1(2)	Success or failure of function.	None
FMT_SMF.1(3)	None	None
FMT_SMF_EXT.3	None	None
FMT_SMR.1(1)	None	None

FMT_SMR.1(2)	None	None
FMT_UNR_EXT.1	None	None
FPT_ITT.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
FPT_ITT.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
FPT_ITT.1(3)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
FPT_TST_EXT.1	Initiation of self-test. Failure of self-test. Detected integrity violation.	Algorithm that caused failure. The TSF code file that caused the integrity violation.
FPT_TST_EXT.1(1)	None	None
FPT_TUD_EXT.1	Success or failure of signature verification.	None
FTA_TAB.1	Change in banner setting.	None
FTP_ITC.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
FTP_ITC.1(3)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
FTP_TRP.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of administrator.
FTP_TRP.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol.

Table 2 Auditable Events

FAU_GEN.1(1).2

Refinement: The [***MDM Server***] 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 2 Auditable Events**,
- [**no other audit relevant information**].

5.1.1.5 Network Reachability Review (FAU_NET_EXT.1)

FAU_NET_EXT.1.1

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

5.1.1.6 Audit Review (FAU_SAR.1)

FAU_SAR.1.1

Refinement: The [***MDM Server***] shall provide Authorized Administrators with the capability to read all audit data from the audit records.

FAU_SAR.1.2

Refinement: The [***MDM Server***] shall provide the audit records in a manner suitable for the Authorized Administrators to interpret the information.

5.1.1.7 External Audit Trail Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1

The [***MDM Server***] shall be able to transmit audit data to an external IT entity using a trusted channel implementing the [***TLS***] protocol.

5.1.1.8 Audit Event Storage (FAU_STG_EXT.2)

FAU_STG_EXT.2.1

The [***MDM Server***] shall protect the stored audit records in the audit trail from unauthorized modification.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (FCS_CKM.1) - Server

FCS_CKM.1.1

Refinement: The [***TSF, TOE platform***] shall 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 meet the following: [FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3],
- ECC schemes using 'NIST curves' P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4,
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1].

5.1.2.2 Cryptographic Key Generation (FCS_CKM.1(1)) - Agent

FCS_CKM.1(1).1

Refinement: The [***TSF***] shall 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 meet the following: [ANSI X9.31-1998, Section 4.1],
- ECC schemes using 'NIST curves' P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4].
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1].

5.1.2.3 Cryptographic Key Establishment (FCS_CKM.2) - Server

FCS_CKM.2.1

Refinement The [***TSF, TOE platform***] shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',
- 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',
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.4 Cryptographic Key Establishment (FCS_CKM.2(1)) - Agent

FCS_CKM.2(1).1

Refinement The [***TSF, TOE platform***] shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',

- 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',
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.5 Cryptographic Key Destruction (FCS_CKM_EXT.4) - Server

FCS_CKM_EXT.4.1

The [***TSF, TOE platform***] shall destroy plaintext keying material and critical security parameters in accordance with the following rules:

- For volatile memory, the destruction shall be executed by a single direct overwrite [***consisting of zeroes***] following by a read-verify.
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1) followed by a read-verify.
- For non-volatile flash memory, the destruction shall be executed [***by a direct overwrite of all copies consisting of zeros followed by a read-verify***].
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

FCS_CKM_EXT.4.2

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

5.1.2.6 Cryptographic Key Destruction (FCS_CKM_EXT.4(1)) - Agent

FCS_CKM_EXT.4(1).1

The [***TSF***] shall destroy plaintext keying material and critical security parameters in accordance with the following rules:

- For volatile memory, the destruction shall be executed by a single direct overwrite [***consisting of zeroes***] following by a read-verify.
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1(1)) followed by a read-verify.
- For non-volatile flash memory that is not wear-leveled, the destruction shall be executed [***by a single direct overwrite consisting of zeros followed by a read-verify***].
- For non-volatile flash memory that is wear-leveled, the destruction shall be executed [***by a single direct overwrite consisting of zeros***].
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

FCS_CKM_EXT.4(1).2

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

5.1.2.7 Cryptographic Operation (Confidentiality Algorithms) (FCS_COP.1(1)) - Server

FCS_COP.1(1).1

Refinement: The [***TSF, TOE platform***] shall perform encryption/decryption in accordance with a specified cryptographic algorithm [***AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode, AES-GCM (as defined in NIST SP 800-38D)***] and cryptographic key sizes 128-bit key sizes and [***256-bit key sizes***].

5.1.2.8 Cryptographic Operation (Hashing) (FCS_COP.1(2)) - Server

FCS_COP.1(2).1

Refinement: The [***TSF, TOE platform***] shall perform cryptographic hashing in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.2.9 Cryptographic Operation (Digital Signature) (FCS_COP.1(3)) - Server

FCS_COP.1(3).1

Refinement: The [***TSF, TOE platform***] shall 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-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5].

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

FCS_COP.1(4).1

Refinement: The [***TSF, TOE platform***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384, SHA-512***] , key sizes [***equal to hash size***] , and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.2.11 Cryptographic Operation (Confidentiality Algorithms) (FCS_COP.1(5)) - Agent

FCS_COP.1(5).1

Refinement: The [***TSF***] shall perform encryption/decryption in accordance with a specified cryptographic algorithm [***AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode, AES-GCM (as defined in NIST SP 800-38D)***] and cryptographic key sizes 128-bit key sizes and [***256-bit key sizes***].

5.1.2.12 Cryptographic Operation (Hashing) (FCS_COP.1(6)) - Agent

FCS_COP.1(6).1

Refinement: The [***TSF***] shall perform cryptographic hashing in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.2.13 Cryptographic Operation (Digital Signature) (FCS_COP.1(7)) - Agent

FCS_COP.1(7).1

Refinement: The [***TSF***] shall 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-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5].

5.1.2.14 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(8)) - Agent

FCS_COP.1(8).1

Refinement: The [***TSF***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384, SHA-512***] , key sizes

[***equal to hash size***] , and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.2.15 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1

The [***MDM Server, MDM Server platform***] shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2

The [***MDM Server, MDM Server platform***] shall implement HTTPS using TLS as specified in FCS_TLSS_EXT.1.

FCS_HTTPS_EXT.1.3

The [***MDM Server, MDM Server platform***] shall [***not establish the connection***] if the peer certificate is deemed invalid.

5.1.2.16 Initialization Vector Generation (FCS_IV_EXT.1)

FCS_IV_EXT.1.1

The MDM Server shall generate IVs in accordance with **Table 3 IV Requirements**.

Cipher Mode	Reference	IV Requirements
Cipher Block Chaining (CBC)	SP 800-38A	IVs shall be unpredictable. Repeating IVs leak information about whether the first one or more blocks are shared between two messages, so IVs should be non-repeating in such situations.
Galois Counter Mode (GCM)	SP 800-38D	IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length).

Table 3 IV Requirements

5.1.2.17 Extended: Random Bit Generation (FCS_RBG_EXT.1) - Server

FCS_RBG_EXT.1.1

The [***TSF, TOE platform***] shall perform all deterministic random bit generation services in accordance with [***NIST Special Publication 800-90A using [HMAC_DRBG (any), CTR_DRBG (AES)[1]]***].

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.

5.1.2.18 Extended: Random Bit Generation (FCS_RBG_EXT.1(1)) - Agent

FCS_RBG_EXT.1(1).1

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

FCS_RBG_EXT.1(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.

[1] The TOE RSA Crypto-J implementation uses HMAC_DRBG with SHA-256 while the TOE platform OpenSSL implementation uses CTR_DRBG with AES-256.

5.1.2.19 SSH Protocol (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1
The [***MDM Server platform***] shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [***5656***, ***6668***].

FCS_SSHS_EXT.1.2
The [***MDM Server platform***] shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHS_EXT.1.3
The [***MDM Server platform***] shall ensure that, as described in RFC 4253, packets greater than [**256K**] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4
The [***MDM Server platform***] shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc, [***no other algorithms***].

FCS_SSHS_EXT.1.5
The [***MDM Server platform***] shall ensure that the SSH transport implementation uses [***ecdsa-sha2-nistp256***, ***ssh-rsa***] and [***no other public key algorithms***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6
The [***MDM Server platform***] shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha2-256, hmac-sha2-512***] and [***no other MAC algorithms***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7
The [***MDM Server platform***] shall ensure that [***diffie-hellman-group14-sha1***] and [***no other methods***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8
The MDM Server shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

5.1.2.20 Cryptographic Key Storage (FCS_STG_EXT.1)

FCS_STG_EXT.1.1
The [***TSF***] shall store persistent secrets and private keys when not in use, in [***platform-provided key storage***].

5.1.2.21 Cryptographic Key Storage (FCS_STG_EXT.4)

FCS_STG_EXT.4.1
The MDM Agent shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.1.2.22 Cryptographic Support (FCS) (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1
The [***TSF***] shall implement [***TLS 1.0 (RFC 3246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

Mandatory Ciphersuites:
- o TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246;

Optional Ciphersuites: [
- ***o TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,***
- ***o TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,***
- ***o TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,***
- ***o TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,***
- ***o TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,***

o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
o TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
o TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSC_EXT.1.2
The [***TSF***] shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3
The [***TSF***] shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.1.4
The [***TSF***] shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.1.5
The [***TSF***] shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1, secp521r1***] and no other curves.

5.1.2.23 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1
The [***MDM Server***] shall implement [***TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
Mandatory Ciphersuites:
o TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246;
Optional Ciphersuites: [
o TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
o TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
o TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
o TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,
o TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSS_EXT.1.2
The [***MDM Server***] shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0 and [***no other TLS version***].

FCS_TLSS_EXT.1.3
The [***MDM Server***] shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.1.4
The [***MDM Server***] shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.5
The [***MDM Server***] shall not establish a trusted channel if the distinguished name (DN) or Subject

Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

FCS_TLSS_EXT.1.6

The [***MDM Server***] shall generate key agreement parameters [***over NIST curves [secp256r1, secp384r1] and no other curves, Diffie-Hellman parameters of size 2048 bits and [3072 bits]***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Enrollment of Mobile Device into Management (FIA_ENR_EXT.1)

FIA_ENR_EXT.1.1

The MDM Server shall authenticate the remote user over a trusted channel during the enrollment of a mobile device.

FIA_ENR_EXT.1.2

The MDM Server shall limit the user's enrollment of devices to [***a number of devices***].

5.1.3.2 Enrollment of Mobile Device into Management (FIA_ENR_EXT.2)

FIA_ENR_EXT.2.1

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

5.1.3.3 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1

Refinement: The [***TSF***] shall allow [***no TSF mediated actions***] on behalf of the user to be performed before the user is authenticated with the Server.

FIA_UAU.1.2

Refinement: The [***TSF***] shall require each user to be successfully authenticated with the Server before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.4 X509 Validation (FIA_X509_EXT.1)- Server

FIA_X509_EXT.1.1

The [***TSF***] shall 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 and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - o Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  - o 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 extendedKeyUsage field.

FIA_X509_EXT.1.2

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

5.1.3.5 X509 Validation (FIA_X509_EXT.1(1)) - Agent

FIA_X509_EXT.1(1).1

The [***TSF***] shall 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 and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - o Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  - o 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 extendedKeyUsage field.

FIA_X509_EXT.1(1).2

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

5.1.3.6 X509 Authentication (FIA_X509_EXT.2) - Server

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

FIA_X509_EXT.2.3

The [***TSF***] shall require a unique certificate for each client device.

5.1.3.7 X509 Authentication (FIA_X509_EXT.2(1)) - Agent

FIA_X509_EXT.2(1).1

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

FIA_X509_EXT.2(1).2

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

FIA_X509_EXT.2(1).3

The [***TSF***] shall require a unique certificate for each client device.

5.1.4 Security management (FMT)

5.1.4.1 Management of Functions in MDM Server (FMT_MOF.1(1))

FMT_MOF.1(1).1

Refinement: The MDM Server 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); and
- listed in FMT_SMF.1(2).

5.1.4.2 Management of Enrollment Function (FMT_MOF.1(2))

FMT_MOF.1(2).1

Refinement: The MDM Server shall restrict the ability to initiate the enrollment process to Authorized Administrators and MD users.

5.1.4.3 Management of Functions in MAS Server (FMT_MOF.1(3))

FMT_MOF.1(3).1

Refinement: The MAS Server shall restrict the ability to configure user groups for user-access to specific applications to the administrator.

5.1.4.4 Management of Download Function in MAS Server (FMT_MOF.1(4))

FMT_MOF.1(4).1

Refinement: The MAS Server shall restrict the ability to download applications to enrolled mobile devices that are compliant with MDM policies and assigned to a user in the application access group.

5.1.4.5 Specification of Management Functions (Server configuration of Agent) (FMT_SMF.1(1))

FMT_SMF.1(1).1

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

1. transition to the locked state, (MDF Function 8)
2. full wipe of protected data, (MDF Function 9)
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 13)
10. install applications, (MDF Function 18)
11. update system software, (MDF Function 17)
12. remove applications, (MDF Function 16)
13. remove Enterprise applications, (MDF Function 19) and

the following commands to the MDM Agent: [

No additional commands] and

the following MD configuration policies:

24. password policy:
    a. minimum password length
    b. minimum password complexity
    c. maximum password lifetime (MDF Function 1)
25. session locking policy:

a. screen-lock enabled/disabled
b. screen lock timeout
c. number of authentication failures (MDF Function 2)

26. wireless networks (SSIDs) to which the MD may connect (MDF Function 6)
27. 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 additional WLAN management functions***] (MDF Function 7)
28. application installation policy by [
    a. specifying authorized application repository(s),
    c. denying application installation] , (MDF Function 10)
29. enable/disable policy for [***camera and microphone***] across MD, [***no other method***], (MDF Function 5) and

the following MD configuration policies: [

33. enable/disable policy for [protocols supporting remote access] (MDF Function 23),
34. enable/disable policy for developer modes (MDF Function 24),
35. enable policy for data-at rest protection (MDF Function 25),
36. enable policy for removable media's data-at-rest protection (MDF Function 26),
46. the unlock banner policy (MDF Function 36),[2]
48. enable/disable [a. USB mass storage mode] (MDF Function 39),
49. enable/disable backup to [remote system] (MDF Function 40),
50. enable/disable
    [a. Hotspot functionality authenticated by [no authentication],
    b. USB tethering authenticated by [no authentication]] (MDF Function 41),
51. enable/disable location services: a. across device [c. no other method] (MDF Function 44),
53. [no other policies]].

5.1.4.6 Specification of Management Functions (Server Configuration of Server) (FMT_SMF.1(2))

FMT_SMF.1(2).1

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

a. configure X.509v3 certificates for MDM Server use
b. configure the [***number of devices***] allowed for enrollment [
d. configure the TOE unlock banner,
e. configure periodicity of the following commands to the agent: [
    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]].

5.1.4.7 Specification of Management Functions (MAS Server) (FMT_SMF.1(3))

FMT_SMF.1(3).1

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

a. Configure application access groups,

[2] This configuration policy is not supported on the evaluated Samsung Galazy S4, Galaxy Note 3, and Galaxy S5 Android mobile devices.

b. Download applications,
c. [***no other functions***].

5.1.4.8 Specification of Management Functions (FMT_SMF_EXT.3)

FMT_SMF_EXT.3.1

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

b. administrator-provided device management functions in MDM PP]

c. Import the certificates to be used for authentication of MDM Agent communications
d. [***no additional functions***].

FMT_SMF_EXT.3.2

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

a. Enroll in management;
b. Configure whether users can unenroll the agent from management
c. [***[configure periodicity of reachability events]***].

5.1.4.9 Security Management Roles (FMT_SMR.1(1))

FMT_SMR.1(1).1

Refinement: The MDM Server shall maintain the roles administrator, MD user, and [***Server primary administrator, Security configuration administrator, Device user group administrator, Auditor***].

FMT_SMR.1(1).2

Refinement: The MDM Server shall be able to associate users with roles.

5.1.4.10 Security Management Roles (FMT_SMR.1(2))

FMT_SMR.1(2).1

Refinement: The MAS Server shall maintain the roles administrator, MD user, enrolled mobile devices, application access groups, and [***Server primary administrator, Security configuration administrator, Device user group administrator, Auditor***].

FMT_SMR.1(2).2

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

5.1.4.11 User Unenrollment Prevention (FMT_UNR_EXT.1)

FMT_UNR_EXT.1.1

The MDM Agent shall provide a mechanism to prevent users from unenrolling the mobile device from management.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Basic Internal TSF Data Transfer Protection (MDM Server) (FPT_ITT.1(1))

FPT_ITT.1(1).1

Refinement: The TSF shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between the MDM Agent and MDM Server.

5.1.5.2 Basic Internal TSF Data Transfer Protection (Distributed TOE) (FPT_ITT.1(2))

FPT_ITT.1(2).1

Refinement: The TSF shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between separate parts of the TOE.

5.1.5.3 Basic Internal TSF Data Transfer Protection (MAS Server) (FPT_ITT.1(3))

FPT_ITT.1(3).1

Refinement: The TSF shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between the MDM Agent and MAS Server.

5.1.5.4 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

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

FPT_TST_EXT.1.2

The [***MDM Server***] shall 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.

5.1.5.5 TSF Testing (FPT_TST_EXT.1(1))

FPT_TST_EXT.1(1).1

The [***MDM Agent***] shall run a suite of self- tests during initial start-up (on power on) to demonstrate correct operation of the TSF.[3]

5.1.5.6 Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

The [***MDM Server***] shall provide Authorized Administrators the ability to initiate updates to TSF software.

FPT_TUD_EXT.1.3

The [***MDM Server***] shall provide a means to verify software updates to the TSF using a digital signature mechanism prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Before establishing a user session, the [***MDM Server, MDM Server Platform***] shall display an Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Inter-TSF Trusted Channel (Authorized IT Entities) (FTP_ITC.1(1))

FTP_ITC.1(1).1

Refinement: The [***MDM Server***] shall use [***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 communication channels and provides

[3] Per direction in a TRRT response to a question regarding to NIAP TD0018, FPT_TST_EXT.1(1).2 has been removed.

assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1(1).2

The TSF shall permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1(1).3

The TSF shall initiate communication via the trusted channel for [***exporting audit logs and LDAP authentication requests***].

5.1.7.2 Inter-TSF Trusted Channel (Authorized IT Entities) (FTP_ITC.1(3))

FTP_ITC.1(3).1

Refinement: The [***MAS Server***] shall use [***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 communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1(3).2

The TSF shall permit the MAS Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1(3).3

The TSF shall initiate communication via the trusted channel for [***exporting audit logs and LDAP authentication requests***].

5.1.7.3 Trusted Path for Remote Administration (FTP_TRP.1(1))

FTP_TRP.1(1).1

Refinement: The [***MDM Server, MDM Server platform***] shall use [***TLS/HTTPS, SSH***] to provide a trusted communication path between itself 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 disclosure and detection of modification of the communicated data.

FTP_TRP.1(1).2

Refinement: The [***MDM Server, MDM Server platform***] shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1(1).3

Refinement: The [***MDM Server, MDM Server platform***] shall require the use of the trusted path for all remote administration actions.

5.1.7.4 Trusted Path for Enrollment (FTP_TRP.1(2))

FTP_TRP.1(2).1

Refinement: The [***MDM Server***] shall use [***TLS/HTTPS***] to provide a trusted communication path between itself 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.

FTP_TRP.1(2).2

Refinement: The [***MDM Server***] shall permit MD users to initiate communication via the trusted path.

FTP_TRP.1(2).3

Refinement: The [***MDM Server***] shall require the use of the trusted path for all MD user actions.

5.2 TOE Security Assurance Requirements

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

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

Table 4 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d
The developer shall provide a functional specification.

ADV_FSP.1.2d
The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

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

ADV_FSP.1.2e
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

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

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

AGD_OPE.1.2c

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

AGD_OPE.1.3c

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

AGD_OPE.1.4c

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

AGD_OPE.1.5c

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

AGD_OPE.1.6c

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

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

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

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

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

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

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

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

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

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

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

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

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

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

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

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

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

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

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

AVA_VAN.1.2e
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

FAU_ALT_EXT.1: The MDM Server component of the TOE can be configured to alert the administrator for device enrollment changes and failed policy deployment. Note that while a device could effectively be unenrolled by being factory reset, there is no explicit unenroll function available to the MDM Agent, though a mobile device can be retired (and hence unenrolled) at the MDM Server. Alerts can be configured and can be displayed on the administrator console, sent to a configured e-mail address, , or sent via a push notification to a configured device.

FAU_ALT_EXT.2: The TOE supports the ability to periodically synchronize the MDM Server and MDM Agents. Using policies, the MDM Server can configure a heartbeat interval that defines the maximum amount of time that can lapse before the MDM Agent will synchronize with the MD Server. The synchronization includes retrieving information about the policies that are installed, thereby ensuring the MDM Server is informed about which policies have been applied. The MDM Server can force a check-in at any time and that would serve to determine MDM Agent connectivity status.

When a mobile device checks in, the MDM Agent performs a compliance check based on configured policies. If any of the settings have not been reported to and acknowledged by the MDM server, the MDM Agent reports those changes. Hence, if something happens, such as a network disruption, that prevents the MDM server from receiving the mobile device compliance information or prevents the MDM Agent from receiving an acknowledgement from the MDM server, that information will be sent the next time the MDM Agent connects until it is finally acknowledged.

FAU_CRP_EXT.1: The MDM Server component of the TOE maintains a configuration database that can be queried to determine the current configuration of mobile devices and can be used to export that configuration information. The MDM server maintains a database of device information including OS versions, device model, installed applications, applied policies, etc. and this information can be exported by an administrator in CSV (comma separated values) format via the administrator web interface.

FAU_GEN.1(1): The MDM Server component of the TOE can generate audit events for a wide range of security-relevant operations including start-up and shutdown, administrator functions, commands sent to MDM Agents, and others (see Table 2 Auditable Events). In addition to identifying the security event, each audit records incudes a time stamp, identify of the responsible user (where applicable), and the outcome of the event (success or failure).

FAU_NET_EXT.1: The MDM Server component of the TOE provides the ability for an administrator to determine the connectivity status of any MDM Agent. Device check-in normally occurs periodically, where an administrator configured the period. Device check-ins can also be initiated by the mobile device user using the MDM Agent or by an administrator using the MDM Server web interface to cause an immediate check-in to ensure or determine the current connectivity status. While an administrator can check the last check-in status of any device via the administrator web interface, the administrator can also configure a policy to send an alert if a device has not checked-in for a configured number of days.

FAU_SAR.1/FAU_STG_EXT.1/FAU_STG_EXT.2: The MDM Server component of the TOE provides the functions necessary for an administrator to review all of the collected audit records, while ensuring that the audit records cannot be modified. The audit records are stored within files on the MDM Server host platform which is

configured to allow only limited direct access (i.e., to the primary security administrator via a restricted SSH shell). Otherwise, the audit files are made available only via MDM Server interfaces that are controlled based on user role. The MDM Server doesn't offer any functions that allow the audit log or individual audit records therein to be modified, inserted, or deleted. The MDM Server component of the TOE also provides the ability to export audit records via a function available in the administrator web interface and the exported records are protected via the HTTPS/TLS connection to that interface. This export transmits audit data in either in CSV (comma separated values) format, text format, or a compressed archive format, depending upon the specific audit data being exported.

6.2 Cryptographic support

FCS_CKM.1(*)/FCS_COP.1(*): The TOE and its platform include and makes use of available cryptographic modules to perform cryptographic operations to support higher level functions (such as communication protocols).

The MDM Server component includes the RSA Crypt-J (6.1) cryptographic library and also utilizes the Red Hat Enterprise Linux OpenSSL Module (1.0.1e with FIPS 3.0) cryptographic functions available in its platform.

The MDM Agent component includes its own OpenSSL (1.0.2d with FIPS 2.0.9) cryptographic library.

The available cryptographic functions are as follows:

Requirement	Component	Function	Details	Algorithm Certificate
FCS_CKM.2	MDM Server	generate asymmetric cryptographic keys used for key establishment	● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes ● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard") ● NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes	CVL #805
FCS_CKM.2	Server Platform	generate asymmetric cryptographic keys used for key establishment	● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes ● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [P-521] (as defined in	CVL #380

Requirement	Component	Function	Details	Algorithm Certificate
			FIPS PUB 186-4, "Digital Signature Standard") ● NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes	
FCS_CKM.1	MDM Server	generate asymmetric cryptographic keys used for authentication	● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [P-521]; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1 for FFC schemes	RSA #1154 ECDSA #357 DSA #701
FCS_CKM.1	Server Platform	generate asymmetric cryptographic keys used for authentication	● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [P-521]; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1 for FFC schemes	RSA #1590 ECDSA #564 DSA #903
FCS_CKM.2(1)	MDM Agent	generate asymmetric cryptographic keys used for key establishment	● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes ● NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard") ● NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer	CVL #801

Requirement	Component	Function	Details	Algorithm Certificate
			Factorization Cryptography" for RSA-based key establishment schemes	
FCS_CKM.1(1)	MDM Agent	generate asymmetric cryptographic keys used for authentication	● ANSI X9.31-1998, Section 4.1 for RSA schemes; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [P-521]; ● FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1 for FFC schemes	RSA #1865 ECDSA #750 DSA #1082
FCS_COP.1(3)	MDM Server	cryptographic signature services	● RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-4, "Digital Signature Standard", ● Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets FIPS PUB 186-4, "Digital Signature Standard" with "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard"),	RSA #1154 ECDSA #357
FCS_COP.1(3)	Server Platform	cryptographic signature services	● RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-4, "Digital Signature Standard", ● Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets FIPS PUB 186-4, "Digital Signature Standard" with "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard"),	RSA #1590 ECDSA #564
FCS_COP.1(4)	MDM Server	keyed-hash message authentication	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 with key sizes equal to hash size	HMAC #1378
FCS_COP.1(4)	Server Platform	keyed-hash message authentication	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 with key sizes equal to hash size	HMAC #1958
FCS_COP.1(1)	MDM	encryption/decryption	● AES-CBC (as defined in NIST SP	AES #2249

Requirement	Component	Function	Details	Algorithm Certificate
	Server		800-38A) mode, ● AES-GCM (as defined in NIST SP 800-38D)	
FCS_COP.1(1)	Server Platform	encryption/decryption	● AES-CBC (as defined in NIST SP 800-38A) mode, ● AES-GCM (as defined in NIST SP 800-38D)	AES #3119
FCS_COP.1(2)	MDM Server	cryptographic hashing	SHA-1, SHA-256, SHA-384, SHA-512	SHS #1938
FCS_COP.1(2)	Server Platform	cryptographic hashing	SHA-1, SHA-256, SHA-384, SHA-512	SHS #2577
FCS_COP.1(7)	MDM Agent	cryptographic signature services	● RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-4, "Digital Signature Standard", ● Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets FIPS PUB 186-4, "Digital Signature Standard" with "NIST curves" P-256, P-384 and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard"),	RSA #1865 ECDSA #750
FCS_COP.1(8)	MDM Agent	keyed-hash message authentication	HMAC-SHA-1, HMAC-SHA-256, HMAC-384, HMAC-512 with key sizes equal to hash size	HMAC #2374
FCS_COP.1(5)	MDM Agent	encryption/decryption	● AES-CBC (as defined in NIST SP 800-38A) mode, ● AES-GCM (as defined in NIST SP 800-38D)	AES #3620
FCS_COP.1(6)	MDM Agent	cryptographic hashing	SHA-1, SHA-256, SHA-384, SHA-512	SHS #3040

FCS_CKM_EXT.4/FCS_CKM_EXT.4(1): The TOE destroys cryptographic keys when they are no longer in use by the MDM Server and MDM Agent. Keys stored in memory are overwritten with zeros, while keys stored on non-volatile media are overwritten with new values as indicated in the table below. Note that in the case of the TOE Platform, the OpenSSL cryptographic module has been designed to destroy keys and other CSPs when no longer required.

The following keys and CSPs are managed by the MDM Server and MDM Agent.

Component	Key or CSP	Where Stored	Destruction
MDM Agent	Certificate private key	Platform key storage	Replaced when a new

			certificate is received
MDM Agent	Certificate private key	Volatile memory (in use)	Overwritten with read verify when TLS session terminates
MDM Agent	TLS session key	Volatile memory	Overwritten with read verify when TLS session terminates
MDM Server	Web Portal certificate private key	Platform key storage	Replaced when a new certificate is loaded
MDM Server	Web Portal certificate private key	Volatile memory (in use)	Overwritten with read verify when TLS session terminates
MDM Server Platform	SSH portal certificate private key	Platform key storage	Replaced when a new certificate is loaded
MDM Server Platform	SSH portal certificate private key	Volatile memory (in use)	Overwritten with read verify when SSH session terminates
MDM Server	Local CA certificate issuing certificate private keys	Platform key storage	Replaced when a new certificate is loaded
MDM Server	Local CA certificate issuing certificate private keys	Volatile memory (in use)	Overwritten with read verify after enrollment is complete (a device certificate has been issued)
MDM Server	TLS session key	Volatile memory	Overwritten with read verify when TLS session terminates
MDM Server Platform	SSH session key	Volatile memory	Overwritten with read verify when SSH session terminates
MDM Server	AES IVs	Volatile memory	Overwritten when TLS session terminates

FCS_HTTPS_EXT.1: The MDM Server component of the TOE includes the ability to support the HTTPS protocol (compliant with RFC 2818) so that remote users can securely connect using the web-based user interface.

FCS_IV_EXT.1: The MDM Server component of the TOE generates initialization vectors (IVs) in accordance with Table 3 IV Requirements. The MDM Server generates IVs for AES CBC using unpredictable (random) IVs drawn from the SHA-256 HMAC_DRBG and AES-256 CTR-DRBG (which meets the "unpredictable" requirement of SP 800-38A), and the Server uses AES CBC encryption for protection of the Server's private keys and user credentials. The MDM Server derives AES CBC and GCM IVs as part of the TLS handshake (which also meets the "unpredictable" and "non-repeating" requirements of SP 800-38A and SP 800-38D respectively).

FCS_RBG_EXT.1: The MDM Server component of the TOE includes a cryptographic module (RSA Crypto-J) and accesses a second cryptographic module (OpenSSL on its platform). In the case of OpenSSL, the library provides an AES-256 CTR_DRBG (cert #631) deterministic random bit generation that is seeded with 384-bits of platform-based entropy with a security strength of 256-bits. In the case of RSA Crypto-J, the MDM Server implements a SHA-256 HMAC-DRBG (cert # 273) also seeded with 384-bits of platform-based entropy with a security strength of 256-bits.

FCS_RBG_EXT.1(1): The MDM Agent component of the TOE incudes a cryptographic library (OpenSSL). The MDM Agent provides an AES-256 CTR-DRBG (cert #950) seeded using 384-bits (also forming a 256-bit entropy input and 128-bit nonce in conformance with SP 800-90A) of platform-based seeding material and further conditioned using an OpenSSL conditioning function.

FCS_SSHS_EXT.1: The MDM Server Platform component of the TOE includes an SSHv2 implementation (that conforms to RFCs 4251, 4252, 4253, 4254, 5656, and 6668) to facilitate remote administration. The MDM Server accepts both user and password and public-key based authentication and once connected any packets greater than 256K will be discarded and after 2^28 packets (or at the request of the client) the connection will be rekeyed. The MDM Server supports the following algorithms:

- Encryption: only AES128-CBC and AES256-CBC;
- Public key: only SSH-RSA;
- MAC: only HMAC-SHA1, HMAC-SHA2-256, and HMAC-SHA2-512; and
- Key exchange: only Diffie-Hellman Group 14 SHA1.

FCS_STG_EXT.1/FCS_STG_EXT.4: Both the MDM Server and MDM Agent components of the TOE use platform provided storage to store persistent and private keys.

The MDM Server stores each of its keys and certificates in flat files that are assigned permissions to restrict those keys to the components/processes that use them. The MDM Agent utilizes the Android keystore on the mobile device to store its keys and certificates.

FCS_TLSC_EXT.1/FCS_TLSS_EXT.1: The MDM Server component of the TOE includes the ability to support the TLS protocol (TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), and TLS 1.2 (RFC 5246)) for the purposed of protecting audit records exported to an external server, protecting communication channels between the MDM Server and MDM Agent, and protecting (in conjunction with HTTPS) remote web-based administrator sessions. It will reject any attempts to use other TLS or SSL versions and TLS sessions will not be established if an applicable certificate if found to be invalid or if the remote network entity doesn't match the distinguished name (DN) in the certificate. The DN can be an exact match or a match with a wildcard in accordance with RFC 6125. Note that when a mobile device is enrolled it is assigned a unique UUID that is associated with the serial number of the device's certificate. When the mobile device connects and presents its certificate, the UUID in the certificate is used to look up the certificate serial number to ensure it matches the certificate that was presented. The MDM Server generates key agreement parameters using NIST curves secp256r1 and secp384r1 and 2048-bit and 3072-bit Diffie-Hellman parameters. Note also that pinned certificates are not supported by the MDM Server or Agent.

The MDM Server component of the TOE also includes an LDAP client that acts as a TLS client. It supports the TLS protocol (TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), and TLS 1.2 (RFC 5246)) for the purpose of protecting the communication channel between the MDM Server and secure LDAP server for authentication. TLS sessions will not be established if an applicable certificate if found to be invalid or if the remote LDAP server doesn't match the distinguished name (DN) in the certificate (i.e., FQDN). The LDAP client supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers.

The MDM Agent component of the TOE includes the ability to support the TLS protocol (TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), and TLS 1.2 (RFC 5246)) for the purposed of protecting the communication channel between the MDM Server and MDM Agent. TLS sessions will not be established if an applicable certificate if found to be invalid or if the remote network entity doesn't match the distinguished name (DN) in the certificate (i.e., FQDN). The MDM Agent supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers. This is the default behavior and there is no means to enable additional curves in the MDM Agent.

The following ciphers are supported by both the MDM Server and MDM Agent:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,

- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289, and
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

The LDAP client supports the ciphers listed above with the exception of the four TLS_DHE ciphers.

6.3 Identification and authentication

FIA_ENR_EXT.1: When a new device is being enrolled, the initial connection is made using TLS where only the MDM Server certificate is authenticated by the MDM Agent. Once the TLS connection is established, the mobile device user must be authenticated with a PIN or username and password that are configured on the MDM Server. Only after the MDM Server determines that the device can be enrolled (including ensuring the user has not exceeded the maximum number of enrolled devices), the device is provisioned with an X.509v3 certificate so that subsequent communication between the MDM Agent and MDM Server will be protected by mutually authenticated TLS. When enrolled (and later when synchronizing), the MDM Server checks the OS version on the mobile device and will quarantine the device if it doesn't meet any configured restrictions. Note that quarantine removes most enterprise configurations, but leaves the ability to check-in. If the device is determined to be in compliance during a subsequent check-in, the quarantined accesses will be restored.

FIA_ENR_EXT.2: During enrollment, the MDM Agent records the unique URL (FQDN) of the MDM Server for future communication purposes. This value is initially configured by the mobile device user when attempting to enroll the mobile device.

FIA_UAU.1: The MDM Server component of the TOE requires users to login (be authenticated) before they can perform any security-related functions. Mobile device users are initially authenticated using a provided PIN or password and after that the MDM Agent uses its provisioned X.509v3 certificate to authenticate itself to the MDM Server. Administrators are assigned usernames and passwords used for authentication when access the MDM Server via the console, web-based user interfaces, or SSH.

FIA_X509_EXT.1/FIA_X509_EXT.2: The MDM Server component of the TOE uses X.509v3 certificates (specifically configured by an administrator) as part of TLS authentication and will not establish TLS sessions if the applicable certificates cannot be determined to be valid. When the MDM Server cannot obtain a CRL and verify the revocation status of a certificate, the certificate is considered to be invalid, the session is not established, the underlying TCP connection is reset and a failed attempt is audited. The same occurs when a CRL is obtained that indicates the certificate is revoked. Only when the CRL indicates the certificate is valid is the TLS session allowed. Additional certificate validation includes checking the validation path, basicConstraints, revocation, and extendedKeyUsage properties (see FIA_X509_EXT.1.1).

FIA_X509_EXT.1(1)/FIA_X509_EXT.2(1): The MDM Agent component of the TOE uses X.509v3 certificates (issued during enrollment) as part of TLS authentication and will not establish TLS sessions if the applicable certificate is not valid. When a session is not established, the underlying TCP connection will be reset, It also performs the validation checks including checking the validation path, basicConstraints, revocation, and extendedKeyUsage properties (see FIA_X509_EXT.1(1).1).

6.4 Security management

FMT_MOF.1(1)/FMT_MOF.1(3)/FMT_MOF.1(4): The MDM Server component of the TOE restricts all security management functions (identified below for FMT_SMF.1(1)/FMT_SMF.1(2)/FMT_SMF.1(3)) to an authorized administrator. This is accomplished by role-based access controls (described below) assigned to each of the functions.

FMT_MOF.1(2): While most security management functions are restricted to an authorized administrator, the authorized administrator can enable mobile device users to enroll their mobile device. An authorized administrator

registers a given device and provides the mobile device user a 6-12 digit PIN or password (that can be more than 15 characters) that will allow them to enroll the device.

FMT_SMF.1(1): A primary function of the MDM Server component of the TOE is to manage mobile devices via installed MDM Agent components. The MDM Server supports the following commands and configuration policies on the mobile devices identified in section 1.4.1.1:

- transition to the locked state,
- full wipe of protected data,
- unenroll from management,
- install policies,
- 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,
- import X.509v3 certificates into the Trust Anchor Database,
- install applications,
- update system software,
- remove applications,
- remove Enterprise applications,
- password Policy:
  - minimum password length
  - minimum password complexity
  - maximum password lifetime
- session locking Policy:
  - screen-lock enabled/disabled
  - screen lock timeout
  - number of authentication failures
- wireless networks (SSIDs) to which the MD may connect
- security Policy for each wireless network:
  - specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)
  - ability to specify security type
  - ability to specify authentication protocol
  - specify the client credentials to be used for authentication
- application installation Policy by
  - specifying authorized application repository(s) (Google Playstore and MDM application server)
  - denying application installation
- enable/disable policy for the camera and microphone
- enable/disable policy for protocols supporting remote access (Hotspot, USB, and Bluetooth tethering)
- enable/disable policy for developer modes
- enable policy for data-at rest protection
- enable policy for removable media's data-at-rest protection
- unlock banner policy
- enable/disable USB mass storage mode
- enable/disable backup to remote system
- enable/disable location services

FMT_SMF.1(2): In addition to managing mobile devices, the MDM Server component of the TOE supports the security management functions to configure and manage itself, including configuring a login banner. Among the available security management functions are the ability to configure X.509v3 certificates, managing the device registration process (enrolling specific devices and limiting the number of devices a user can enroll), 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, and query the current version of installed mobile applications.

FMT_SMF.1(3): Furthermore, in support of application hosting, the MDM server supports the configuration of application groups in the form of labels assigned to individual apps and devices. It also supports the ability to download applications for deployment.

FMT_SMF_EXT.3/FMT_UNR_EXT.1: The MDM Agent component of the TOE can be configured with a X.509v3 certificate suitable to facilitate secure communication with the MDM Server. This certificate is provisioned during device enrollment. The MDM Server can be configured to use an external CA or to use a local CA using an administrator-configured certificate to sign CSRs from the MDM Agent during enrollment. Once secure communication is enabled and the device is enrolled, the MDM Agent accepts commands and policies from the enrolled MDM Server and implements those commands and policies (identified above). The MDM Agent can also be unenrolled from an enrolled MDM Server. This is accomplished by retiring the mobile device on the MDM Server or alternately by using the Android Settings - Device Administrators function to remove the MDM Agent Administrator access on the mobile device. Note that the MDM Agent can restrict that latter ability depending on its configured policies.

Once an MDM Agent is enrolled, it offers a limited set of trouble shooting functions to the mobile device user: force check-in, send agent log, reinstall certificates, and show warnings.

FMT_SMR.1(1)/FMT_SMR.1(2): The MDM Server component of the TOE implements a role-based access mechanism. Initially, there is only one security management role by default – administrator – and additional roles can be configured. The administrator has access to the SSH (or console) command-line interface to perform low-level management functions as well as access to the web-based System Manager and Admin Portals. Within the Admin Portal additional users can be defined and assigned specific user and management roles as follows.

Users can connect to a user portal and perform functions on devices assigned to that user depending on their assigned roles. The following list of roles can be individually assigned to each user:

- Wipe Device (cause the registered device to be reset)
- Lock Device (lock the device)
- Unlock Device (unlock the device)
- Locate Device (retrieve location information from the device)
- Retire Device (unenroll the device)
- Register Device (initiate a PIN registration request or send registration instructions for a device)
- Change Device Ownership (change the device ownership to another defined user)

User added in the Admin Portal cannot access the web-based System Manager or SSH (or console) command-line interfaces. Furthermore, added users can access the Admin Portal only if they have been assigned an administrator role. The following administrator roles can be individually assigned to each user:

- View dashboard, device page, device details
- Manage devices
- Manage devices, restricted
- Wipe device
- Add device
- Manage ActiveSync device
- Manage AppTunnel
- Manage device enrollment (iOS)
- Delete retired device
- View apps in device details
- Locate device
- View label
- Manage Label
- View user
- Manage user

- Manage app
- Apply and remove application label
- View configuration
- Manage configuration
- Apply and remove configuration
- View policy
- Manage policy
- Apply and remove policy label
- View settings
- Manage settings
- View logs and events
- Manage logs and events
- Manage administrators and device spaces
- Manage content
- View content, apply and remove content labels

The role settings allow fine-grained definition of administrative users.

Minimally the following roles are supported in the TOE:

- Administrator – This is the initial administrator that has full control of all functions.
- Mobile device user – This is a user added in the Admin Portal, but not assigned ant administrative roles.
- Enrolled mobile device – This is a mobile device that has been enrolled by a mobile device user.
- Application access group – The TOE allows "Labels" to be defined and apps to be assigned. Devices can also be assigned to labels and that associated services to restrict or allow access to corresponding apps.
- Server primary administrator – Same as Administrator above.
- Security configuration administrator – This is a user with access to the web-based System Manager and SSH (or console) command-line interfaces, as well as most of the Admin Portal manage roles.
- Device user group administrator – This is a user that minimally has the manage user Admin Portal role, but is not assigned the manage administrators and device spaces role.
- Auditor – This is a user that minimally has the manage logs and events Admin Portal role. If necessary for a given deployment, an auditor can be provided access to the System Manager Portal to have access to low level protocol audit records (application logs).

6.5 Protection of the TSF

FPT_ITT.1(1)/ FPT_ITT.1(2)/ FPT_ITT.1(3): The TOE provides a trusted channel between the MDM Server and MDM Agent that is protected through the use of TLS. During enrollment, only the MDM Server is authenticated by the MDM Agent. Once enrolled, all communication between the MDM Server and MDM Agent is protected using this channel using mutual authentication. Note that the MDM server also performs the MAS server functions, so the only distributed TOE components are the MDM server and associated MDM agents.

FPT_TST_EXT.1/FPT_TST_EXT.1(1): Both the MDM Server and MDM Agent components of the TOE utilize cryptographic modules or libraries that include self-tests to ensure the available cryptographic operations (including AES, RSA, ECDSA, SHA, HMAC-SHA functions) are performing correctly when starting up.

The MDM Server leverages its platform RPM Package Manager (RPM) functions to verify the integrity of its own executable files during start-up. The platform maintains a database of all installed RPMs, including the TOE, and during boot each RPM is checked with a 2048-bit RSA signature and non-configuration files inside each RPM are checked with SHA-256 hashes. Any errors are reported on the console and the MDM Server will attempt to restart if an error is detected without intervention by an administrator.

FPT_TUD_EXT.1: The MDM Server component of the TOE provides functions to query and update the MDM Server software version. When updating the MDM Server software, each new update is installed as an RPM package and is verified using a MobileIron digital signature prior to installation.

RPMs are signed with GPG (GNU's PGP version). The MobileIron signing key is RSA 2048 and the CentOS signing key is RSA 4096. When a package is installed, RedHat/CentOS validate the PGP signature on the package. The RPM includes digests of the files within the RPM. These digests are stored in a database on the system during package install. During boot, the contents of each file are verified against the stored digests.

6.6 TOE access

FTA_TAB.1: The MDM Server component of the TOE can be configured to display an administrator-defined message when an administrator is logging onto the MDM Server to access available security functions.

The web-based user interfaces (including the System Manager and Admin Portal) can be configured with a textual banner up to 2048 characters that is displayed on the login screen. The SSH (and console) interface can be configured separately with a textual banner that is displayed immediately upon connecting, but before the user logs in (when using a password).

6.7 Trusted path/channels

FTP_ITC.1(1)/FTP_ITC.1(3): The MDM Server uses TLS to secure communication when exporting audit records (this is really part of FTP_TRP.1(1) below) as well as an external LDAP authentication server.

FTP_TRP.1(1): The MDM Server provides a web-based user interface (including a System Manager and Admin Portal) to the MDM Server for remote administration. Each web-based session can be initiated by administrators and is protected through the use of HTTPS/TLS. The MDM Server Platform also provides a restricted shell (CLISH) for low level management of the server. CLISH is accessible via SSH to ensure a secure communication channel.

FTP_TRP.1(2): The MDM Server provides a TLS interface as described above for FPT_ITT.1(*) that can be accessed by mobile device users via the MDM Agent.