VMware AirWatch Mobile Device Management

Security Target

ST Version: 1.0
December 27, 2016

AirWatch LLC
1155 Perimeter Center West
Suite 100
Atlanta, GA 30338

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
304 Sentinel Drive, Suite 1160
Annapolis Junction, MD 20701

Table of Contents

1 Security Target Introduction 7

1.1 ST Reference 7

1.1.1 ST Identification 7

1.1.2 Document Organization 7

1.1.3 Terminology 8

1.1.4 Acronyms 8

1.1.5 Reference 9

1.2 TOE Reference 10

1.3 TOE Overview 10

1.4 TOE Type 11

2 TOE Description 13

2.1 Evaluated Components of the TOE 13

2.2 Components and Applications in the Operational Environment 13

2.3 Excluded from the TOE 14

2.3.1 Not Installed 14

2.3.2 Installed but Requires a Separate License 14

2.3.3 Installed But Not Part of the TSF 14

2.4 Physical Boundary 14

2.4.1 Hardware 14

2.4.2 Software 14

2.5 Logical Boundary 15

2.5.1 Security Audit 15

2.5.2 Cryptographic Support 15

2.5.3 Identification and Authentication 16

2.5.4 Security Management 17

2.5.5 Protection of the TSF [SERVER] 17

2.5.6 TOE Access [SERVER] 18

2.5.7 Trusted Path/Channels [SERVER] 18

3 Conformance Claims 19

3.1 CC Version 19

3.2 CC Part 2 Conformance Claims 19
3.3 CC Part 3 Conformance Claims 19
3.4 PP Claims 19
3.5 Package Claims 19
3.6 Package Name Conformant or Package Name Augmented 20
3.7 Conformance Claim Rationale 20
4 Security Problem Definition 21
4.1 Threats 21
4.2 Organizational Security Policies 21
4.3 Assumptions 22
4.4 Security Objectives 22
4.4.1 TOE Security Objectives 22
4.4.2 Security Objectives for the Operational Environment 23
4.5 Security Problem Definition Rationale 24
5 Extended Components Definition 25
5.1 Extended Security Functional Requirements 25
5.2 Extended Security Assurance Requirements 25
6 Security Functional Requirements 26
6.1 Conventions 26
6.2 Security Functional Requirements Summary 26
6.3 Security Functional Requirements 28
6.3.1 Class FAU: Security Audit 28
6.3.2 Class FCS: Cryptographic Support 33
6.3.3 Class FIA: Identification and Authentication 37
6.3.4 Class FMT: Security Management 39
6.3.5 Class FPT: Protection of the TSF 43
6.3.6 Class FTA: TOE Access 44
6.3.7 Class FTP: Trusted Path/Channels 44
6.4 Statement of Security Functional Requirements Consistency 46
7 Security Assurance Requirements 47
7.1 Class ADV: Development 47

7.1.1 Basic Functional Specification (ADV_FSP.1) 47
7.2 Class AGD: Guidance Documentation 48
7.2.1 Operational User Guidance (AGD_OPE.1) 48
7.2.2 Preparative Procedures (AGD_PRE.1) 49
7.3 Class ALC: Life Cycle Support 49
7.3.1 Labeling of the TOE (ALC_CMC.1) 49
7.3.2 TOE CM Coverage (ALC_CMS.1) 50
7.4 Class ATE: Tests 50
7.4.1 Independent Testing - Conformance (ATE_IND.1) 50
7.5 Class AVA: Vulnerability Assessment 51
7.5.1 Vulnerability Survey (AVA_VAN.1) 51
8 TOE Summary Specification 52
8.1 Security Audit 52
8.1.1 [SERVER] FAU_ALT_EXT.1: 52
8.1.2 [AGENT] FAU_ALT_EXT.2: 53
8.1.3 [SERVER] FAU_GEN.1(1): 53
8.1.4 [AGENT] FAU_GEN.1(2)/Agent: 55
8.1.5 [SERVER] FAU_GEN.1(2)/Server: 56
8.1.6 [SERVER] FAU_NET_EXT.1: 56
8.1.7 [SERVER] FAU_SAR.1: 57
8.1.8 [SERVER] FAU_STG_EXT.1(1): 57
8.1.9 [SERVER] FAU_STG_EXT.1(2): 57
8.1.10 [SERVER] FAU_STG_EXT.2: 57
8.2 Cryptographic Support 57
8.2.1 FCS_CKM.1: 58
8.2.2 FCS_CKM.2: 58
8.2.3 FCS_CKM_EXT.4: 58
8.2.4 FCS_COP.1(1): 59
8.2.5 FCS_COP.1(2): 59
8.2.6 FCS_COP.1(3): 59
8.2.7 FCS_COP.1(4): 60

8.2.8 [SERVER] FCS_HTTPS_EXT.1 60
8.2.9 FCS_RBG_EXT.1: 60
8.2.10 [SERVER] FCS_STG_EXT.1(1): 60
8.2.11 [AGENT] FCS_STG_EXT.1(2): 61
8.2.12 FCS_TLSC_EXT.1: 61
8.2.13 [SERVER] FCS_TLSS_EXT.1: 62
8.3 Identification and Authentication 62
8.3.1 [SERVER] FIA_ENR_EXT.1 62
8.3.2 [AGENT] FIA_ENR_EXT.2: 63
8.3.3 [SERVER] FIA_UAU.1: 63
8.3.4 FIA_X509_EXT.1: 63
8.3.5 FIA_X509_EXT.2: 64
8.4 Security Management 64
8.4.1 [SERVER] FMT_MOF.1(1): 64
8.4.2 [SERVER] FMT_MOF.1(2): 65
8.4.3 [SERVER] FMT_MOF.1(3): 65
8.4.4 [SERVER] FMT_MOF.1(4): 65
8.4.5 [SERVER] FMT_POL_EXT.1: 66
8.4.6 [AGENT] FMT_POL_EXT.2: 66
8.4.7 [SERVER] FMT_SMF.1(1): 66
8.4.8 [SERVER] FMT_SMF.1(2): 69
8.4.9 [SERVER] FMT_SMF.1(3): 69
8.4.10 [AGENT] FMT_SMF_EXT.3: 70
8.4.11 [SERVER] FMT_SMR.1(1): 70
8.4.12 [SERVER] FMT_SMR.1(2): 71
8.4.13 [AGENT] FMT_UNR_EXT.1: 71
8.5 Protection of the TSF 71
8.5.1 [SERVER] FPT_ITT.1(1): 71
8.5.2 [SERVER] FPT_ITT.1(2): 71
8.5.3 [SERVER] FPT_ITT.1(3): 72
8.5.4 [SERVER] FPT_TST_EXT.1: 72

8.5.5 [SERVER] FPT_TUD_EXT.1: ... 72
8.6 TOE Access ... 72
8.6.1 [SERVER] FTA_TAB.1: ... 72
8.7 Trusted Path/Channels ... 73
8.7.1 [SERVER] FTP_ITC.1(1): ... 73
8.7.2 [SERVER] FTP_ITC.1(2): ... 73
8.7.3 [SERVER] FTP_ITC.1(3): ... 73
8.7.4 [SERVER] FTP_TRP.1(1): ... 73
8.7.5 [SERVER] FTP_TRP.1(2): ... 73

Table of Figures

Figure 1: TOE Boundary ... 11

Table of Tables

Table 1: Customer Specific Terminology ... 8
Table 2: CC Specific Terminology ... 8
Table 3: Acronym Definition ... 9
Table 4: Evaluated Components of the TOE ... 13
Table 5: Evaluated Components of the Operational Environment ... 14
Table 6: Cryptographic Algorithm Table for TOE Platforms ... 16
Table 7: TOE Threats ... 21
Table 8: TOE Organizational Security Policies ... 22
Table 9: TOE Assumptions ... 22
Table 10: TOE Objectives ... 23
Table 11: Operational Environment Objectives ... 23
Table 12: Security Functional Requirements for the TOE ... 28
Table 13: Agent/Server Auditable Events ... 31
Table 14: Server Auditable Events by Enforcing Component ... 54
Table 15: Sample Server Auditable Events ... 55

Table 16: Sample Agent Auditable Events 56

1 Security Target Introduction

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

1.1 ST Reference

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

1.1.1 ST Identification

ST Title:	VMware AirWatch Mobile Device Management Security Target
ST Version:	1.0
ST Publication Date:	December 27, 2016
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

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

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

Chapter 3 describes the conformance claims made by this ST.

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

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

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

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

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

1.1.3 Terminology

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

Term	Definition
Administrator	An individual that has the ability to manage some aspect of mobile device configuration using the Admin Console.
AirWatch Administrator	The class of TOE Administrators that allows comprehensive access to the AirWatch environment, excluding the Administration tab under System Configuration.
Application Management	The class of TOE Administrators that provides the ability to deploy and manage internal and public apps for managed devices.
End User	An individual who possesses a mobile device that is managed by AirWatch and who has limited authority to perform management functions using the Self-Service Portal
Role	The level of access given to Administrator accounts. The TOE comes with pre-defined roles but new roles with custom sets of privileges can be created.
System Administrator	The class of TOE Administrators that have complete access to an AirWatch environment, including access to Password and Security settings, Session Management and AirWatch Admin Console audit information contained in the Administration tab under System Configuration.

Table 1: Customer Specific Terminology

Term	Definition
Authorized Administrator	The claimed Protection Profile defines an Authorized Administrator role that is authorized to manage the TOE and its data. For the TOE, this is considered to be any user with the 'admin' role.
Security Administrator	Synonymous with Authorized Administrator.
Trusted Channel	An encrypted connection between the TOE and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application an Authorized Administrator uses to manage it (web browser, terminal client, etc.).
User	In a CC context, any individual who has the ability to manage TOE functions or data.

Table 2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
CC	Common Criteria
CPU	Central Processing Unit
DEP	[Apple] Device Enrollment Program
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure over a bidirectional TLS encrypted tunnel
IP	Internet Protocol
IT	Information Technology

LDAP	Lightweight Directory Access Protocol
MAS	Mobile Application Store
MDM	Mobile Device Management
NIAP	National Information Assurance Partnership
OS	Operating System
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SSL	Secure Sockets Layer
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

Table 3: Acronym Definition

1.1.5 Reference

[1] Protection Profile for Mobile Device Management, version 2.0 (MDMPP)

[2] Extended Package for Mobile Device Management Agents, version 2.0 (MDMAPP)

[3] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-001

[4] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-002

[5] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-003

[6] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-004

[7] NIST Special Publication 800-56B Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography, August 2009

[8] NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation, December 2001

[9] FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules May 25, 2001

[10] FIPS PUB 180-3 Federal Information Processing Standards Publication Secure Hash Standard (SHS) October 2008

[11] FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) March 2012

[12] FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard July 2013

[13] FIPS PUB 197 Advanced Encryption Standard November 26 2001

[14] FIPS PUB 198-1 Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008

[15] Apple iOS 9.2 MDFPPv2 Security Target (VID 10695)

[16] Mobile Device Management Supplemental Administrative Guidance v1.0

[17] VMware AirWatch iOS Platform Guide AirWatch v9.0

[18] VMware AirWatch Windows Desktop Platform Guide v9.0

[19] DoD Annex for Protection Profile for Mobile Device Management v2.0, Version 1, Release 2, 20 July 2015

[20] DoD Annex for Extended Package for Mobile Device Management Agents v2.0, Version 1, Release 1, 29 April 2015

[21] Microsoft Windows Common Criteria Evaluation Microsoft Windows 10 Microsoft Windows Server 2012 R2 Security Target Version Number 1

[22] Microsoft Windows Common Criteria Evaluation Microsoft Windows 10 Microsoft Windows Server 2012 R2 Common Criteria Supplemental Admin Guidance Version Number 0.09

1.2 TOE Reference

The TOE is the VMware AirWatch Mobile Device Management version 9.1 comprising the VMware AirWatch MDM Server 9.1, and VMware AirWatch MDM Agent version 9.1.

1.3 TOE Overview

The TOE is a Mobile Device Management product and is comprised of two MDM Server components and an MDM Agent component. Two MDM Server components exist because the evaluated configuration of the TOE is to deploy it in an on-premises configuration, which requires a secondary instance of the MDM Server residing outside the organization’s firewall in a demilitarized zone (DMZ) where it is exposed to external network traffic from the internet. The MDM Server component provides a centralized enterprise level management capability for a collection of mobile devices running the VMware AirWatch MDM Agents. It also provides a Mobile Application Store (MAS) Server that allows managed devices to download apps from a trusted repository that resides within the organization managing the device. The management functionality includes management of Administrators and users, mobile device enrollment, mobile device status, mobile device compliance and policy management, and application management. Administrators access the VMware AirWatch MDM Server through the Admin Console interface in order to manage users, policies, and devices. Users access the VMware AirWatch MDM Server through the Self-Service Portal, which allows them to perform administrative functions relating to their own devices.

The MDM Server runs on a Microsoft Windows Server 2012 R2 operating system and authentication to the MDM Server is provided by the Active Directory/LDAP service. The MDM Server also provides local authentication for initial setup and to mitigate a denial of service in the event the Active Directory/LDAP service is unavailable. The MDM Server stores audit data locally in a SQL database but can send audit records via a TLS encrypted trusted channel to a remote Syslog Server for remote storage.

In the evaluated configuration, the MDM Agent runs on a mobile device running one Apple iOS 9 or 10. The communication channel between the MDM Agent and the MDM Server is protected by TLS. Apple DEP is used to enroll the device with the MDM Server so that it can be managed by the MDM Server. Also, the MDM Agent provides status and policy information about the mobile device to the MDM Server. Figure 1 depicts the network configuration of the TOE.

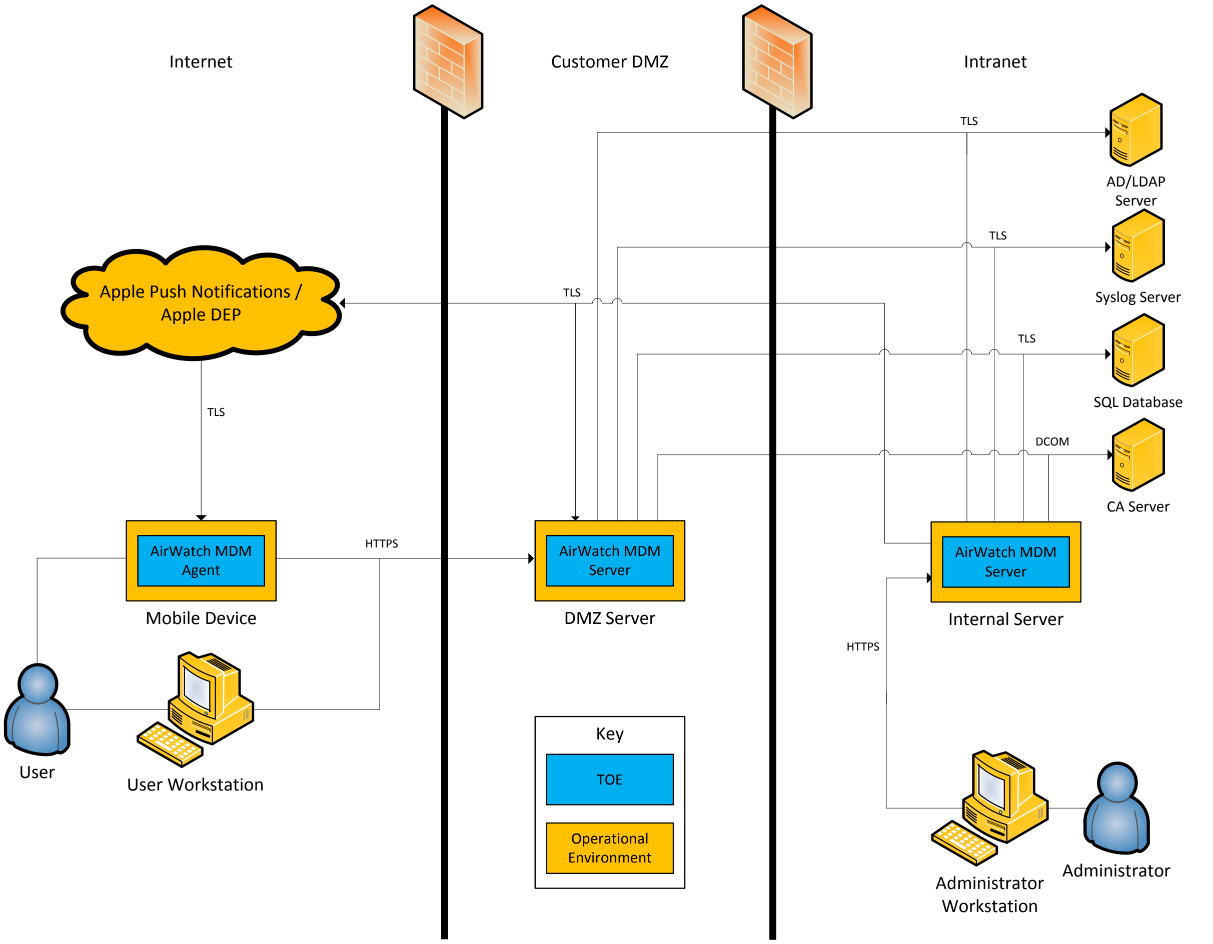

Figure 1: TOE Boundary

As depicted in Figure 1, the TOE consists of two VMware AirWatch MDM Server instances and one or more instances of the VMware AirWatch MDM Agent running on mobile devices. The expected deployment of the TOE is to have an on-premises deployment with two instances of the VMware AirWatch MDM Server running: one in the deploying organization's DMZ and one behind the firewall. The external server hosts the Self-Service Portal so that enterprise users can enroll and manage their own devices from outside the firewall and the Admin Console resides behind the firewall. The internal server communicates indirectly with the DMZ server through querying/modifying the SQL database used by both instances. The connection between the MDM Agent devices and the MDM Server is also protected by HTTPS. The connections between the MDM Server and the Syslog Server/RDBMS and between the MDM Server and AD/LDAP are protected with TLS. The MDM Server is also connected to a CA server in the internal network via DCOM for the purposes of performing mutual authentication.

1.4 TOE Type

The TOE is a Mobile Device Management product consisting of MDM Server software and the MDM Agent which runs on mobile devices. The MDMPP states:

“The MDM Server is an application on a general-purpose platform or on a network device, executing in a trusted network environment. The MDM Server provides administration of the mobile device policies and reporting on mobile device behavior. The MDM Server is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status, and sending commands to the Agents. The platform on which the MDM Server software runs is either a general-purpose platform or a network device.”

The MDM Server TOE type is justified because the TOE software provides centralized enterprise level management capabilities for MDM Agents running on mobile devices, including enrollment, policy management and device status and the MDM Server runs on Microsoft Windows 2012 R2, which is a general purpose platform.

The MDMPP also states:

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

This statement is re-iterated in the MDMAPP. The MDM Agent TOE type is justified because the TOE Agent software is installed on a mobile device as an application developed by AirWatch and establishes a secure connection back to the MDM Server protected by TLS.

2 TOE Description

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

2.1 Evaluated Components of the TOE

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

Component	Definition
VMware AirWatch MDM Server on Microsoft Windows 2012 R2 (including MAS)	MDM Server Component
VMware AirWatch MDM Agent on Apple iOS 9* (VID 10695)	MDM Agent Device
VMware AirWatch MDM Agent on Apple iOS 10**	MDM Agent Device

Table 4: Evaluated Components of the TOE

*VID 10695 certified iOS 9.2. However, since minor releases are covered by Assurance Maintenance and best practice is to use patched software, iOS 9.3 was tested.

**At the time of publication, Apple iOS 10 has not been certified on the NIAP PCL. The TOE was tested on this OS platform in order to ensure its compatibility with future certified products.

As shown in Figure 1, the TOE boundary on the end user mobile devices includes only the VMware AirWatch MDM Agent itself; the actual devices (using the A7 processor) have been evaluated against the Mobile Device Fundamentals Protection Profile under the Validation ID number identified in Table 4 above.

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
Certification Authority (CA) Server	The MDM Server component of the TOE connects to the CA Server via DCOM during device enrollment so that the TOE can provide each device with a unique certificate generated by the CA Server.
Microsoft SQL Enterprise	The TOE's RDBMS database, used to store configuration settings.
Syslog Server	The MDM Server component of the TOE connects to the Syslog Server to persistently store audit data for the MDM Server's own operation as well as the audit data collected from the MDM Agents that it manages.
Windows Server 2012 R2 Active Directory Certificate Services	Certificate Authority providing certificate services for the TOE.
Windows Server 2012 R2 Active Directory / LDAP Server	Identity store that defines users for device enrollment and administrator accounts for access to the Admin Console.

Table 5: Evaluated Components of the Operational Environment

2.3 Excluded from the TOE

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

2.3.1 Not Installed

There are no components that are not installed.

2.3.2 Installed but Requires a Separate License

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

2.3.3 Installed But Not Part of the TSF

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

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

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

2.4 Physical Boundary

2.4.1 Hardware

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

2.4.2 Software

The VMware AirWatch MDM Server runs on Windows Server 2012 and includes the Admin Console, User Self-Service, MDM Server, and MAS Server functions.

The VMware AirWatch MDM Agent runs on the Apple iOS operating system in its evaluated configuration.

2.5 Logical Boundary

The TOE is comprised of several security features. Each of these security features consists of several security functionalities, as identified below. This ST includes both MDM Agent and MDM Server functionality. The security requirements that apply to only the MDM Server component are denoted by [SERVER] and to only the MDM Agent component are denoted by [AGENT]. If the security requirement applies to both components, it is not denoted.

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

2.5.1 Security Audit

[SERVER] The MDM Server component of the TOE creates audit records for all of the auditable events found in Table 13 that are relevant to the MDM Server. It also audits its own startup and shutdown and administrative activities. Audit data is generated for configuration of the MDM Server itself as well as Server-initiated management activities that affect one or more managed mobile devices. The MAS Server also generates audit records when it experiences a failure to push or update an application on a managed mobile device. The audit records contain the subject identity and date and time of the auditable event. Other security-relevant information is included in the audit records as needed, depending on the function that is being audited. The audit records are stored locally in an SQL database and are transferred to a remote Syslog database over a TLS encrypted trusted channel. Audit records can be viewed on the Administrator Console.

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

[AGENT] MDM Agent audit records are created as long as the underlying mobile device is powered on. The MDM Agent generates audit records for the activities it performs as a result of its interactions with the MDM Server or as a result of stored policy information. The MDM Agent generates audit records for security-relevant activity whenever the underlying mobile device is powered on. The MDM Agent facilitates alerting by providing data to the MDM Server on a periodic basis. The MDM Server can then analyze this data (or the absence of data in the case of periodic reachability events) in order to determine if anomalous behavior is occurring.

2.5.2 Cryptographic Support

The MDM Server and the MDM Server platform use cryptography provided by the cryptographic algorithms found in the CNG.sys (CMVP certificate #2356) and BCryptPrimitives.dll (CMVP certificate #2357) cryptographic modules for the Windows Server 2012 platform. The MDM Server uses cryptography to establish TLS and TLS/HTTPS trusted channels and paths to ensure secure communications of data in transit. This includes the use of RSA, Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) key establishment techniques. The MAS Server is integrated with the MDM Server so it uses the same cryptography.

The following table contains the CAVP algorithm certificates corresponding to the server CMVP certificates #2357 and #2356, as well as the agent CMVP certificates #2594 and #2827. In the evaluated configuration, the MDM Agent was tested on devices using the A7 processor running Apple iOS 9 and 10 (both 64-bit). The algorithm certificates are the same for cert. #2357 and #2356 except where indicated:

Algorithm	CAVP Cert. # (Server)	CAVP Cert. # (iOS 9)	CAVP Cert. # (iOS 10)
AES	2832	3686	4255
DRBG	489	993	1353
DSA (certificate #2357 only)	855	N/A	N/A
ECDSA	505	781	1003
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384	1773	2432	2829
KAS ECC, KAS FFC	47	N/A	N/A
RSA	1487, 1493, 1519	1908	2314
SHA-1, SHA-256, SHA-384, SHA-512	2373	3100	3531

Table 6: Cryptographic Algorithm Table for TOE Platforms

2.5.3 Identification and Authentication

[SERVER] The MDM Agent registers with the MDM Server so that it can be enrolled into management by the MDM Server. The user that is performing the enrollment must have a user account on the MDM Server so that they can access the Self-Service Portal and so that they are recognized by the system. This can either be a basic username/password defined on the MDM Server or a centrally defined Active Directory/LDAP credential.

Administrators (through the Admin Console) and users (through the Self-Service Portal) cannot access the MDM Server without being authenticated. Administrators and users can view the configured pre-authentication warning banner and query the MDM Server’s software version number prior to authentication, but all other TSF-mediated actions require authentication. In the event of a forgotten password, the TSF provides the ability to send a password reset email which requires a security question to be answered.

The MDM Server interfaces with the underlying Windows Server 2012 platform to provide certificate validation services via Microsoft Authenticode. Certificates are used for TLS/HTTPS authentication, code signing for software updates, code signing for integrity verification, and signing of MDM policies. In the evaluated configuration, the TOE will be loaded with organizational certificates that were generated in the Operational Environment; the TSF is not responsible for certificate generation.

[AGENT] During the enrollment process, the MDM Agent records the MDM Server’s DNS name and full URL with hostname. The MDM Agent also receives a certificate that is used to validate signed policies that are transmitted from the MDM Server. Similar to the MDM Server, the MDM Agent relies on the underlying platform to perform certificate validation.

2.5.4 Security Management

[SERVER] The TSF provides separate administrative interfaces for Administrators and for users. Administrators use the Admin Console to manage users, policies, and devices, while users use the Self-Service Portal to perform actions related to their own devices. Device enrollment can be initiated by either Administrators or by users. Regardless of the administrative interface that is used to perform an action that affects an MDM Agent, the interface the MDM Server uses to communicate with the MDM Agent is the same. The MDM Server can be used to transmit specific commands to a managed device such as forcibly locking the device, initiating a wipe operation, or sending a push notification. The MDM Server can also define policies (known as profiles) that specify the configuration settings for a device. These configuration settings can include functionality such as configuration of the password policy and what settings are applied to WiFi connections. The functionality that is configurable by the MDM Server is dependent on what the iOS platform provides the ability for third-party software to configure. The MDM Server also may transmit these policies either to the MDM Agent residing on the managed device or directly to the device itself, depending on the functionality being configured. All policy data sent from the MDM Server is signed using ECDSA with either SHA-256 or SHA-512 depending on whether it goes to the OS platform itself or to the TOE MDM Agent that resides there.

The MDM Server also allows for configuration of its own functions. This includes functionality such as defining Administrators and groups, defining the types of devices that are permitted to enroll, defining the communications period for periodic reachability events with managed devices, and configuration of the pre-authentication warning banner. The MDM Server also provides Administrators with the ability to query audit records as well as information about the managed devices.

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

[AGENT] The MDM Agent communicates with the underlying OS platform to validate signed policy updates when they are received. Apple DEP is responsible for enrolling the device into management and preventing user-directed unenrollment. The MDM Agent is also responsible for receiving policy updates and forwarding them to the underlying platform, depending on what function is being managed by the update. Some MDM Server-initiated functionality is communicated directly to the platform rather than through the MDM Agent.

2.5.5 Protection of the TSF [SERVER]

The communications between the MDM Server and MDM Agent are protected using HTTPS. If the TOE's operational environment is such that multiple copies of the same MDM Server are deployed inside and outside of an organization's DMZ, then the communications between these components is also protected by HTTPS.

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

The TOE performs updates of its software and verifies the digital signatures of the updates prior to installing them.

2.5.6 TOE Access [SERVER]

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

2.5.7 Trusted Path/Channels [SERVER]

The trusted communication channels between the MDM Server and the device running the MDM Agent, the syslog audit server, AD/LDAP authentication server and the SQL database server are trusted communications channels which make use of TLS or TLS/HTTPS as the protection mechanism, depending on the interface.

The MDM Server platform uses TLS/HTTPS to provide a trusted path between itself and remote Administrators (through the Admin Console) and users (through the Self-Service Portal).

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 4 September 2012.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through December 27, 2016.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 to include all applicable NIAP and International interpretations through December 27, 2016.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- Protection Profile for Mobile Device Management, version 2.0 [MDMPP]
- Extended Package for Mobile Device Management Agents, version 2.0 [MDMAPP]

3.5 Package Claims

The TOE claims exact compliance to the Protection Profile for Mobile Device Management and Extended Package for Mobile Device Management Agents, which are conformant with CC Part 3.

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

MDMPP:

- FAU_GEN.1(2)/Server
- FAU_SAR.1
- FAU_STG_EXT.1(2)
- FAU_STG_EXT.2
- FCS_TLSC_EXT.1
- FMT_MOF.1(3)
- FMT_MOF.1(4)
- FMT_SMF.1(3)
- FMT_SMR.1(2)
- FPT_ITT.1(1)
- FPT_ITT.1(2)
- FPT_ITT.1(3)
- FTA_TAB.1
- FTP_ITC.1(2)
- FTP_ITC.1(3)

MDMAPP:

- FAU_GEN.1(2)/Agent
- FMT_POL_EXT.2

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

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE claim exact conformance to the MDMPP and MDMAPP.

3.7 Conformance Claim Rationale

The MDMPP states the following:

“The MDM Server is an application on a general-purpose platform or on a network device, executing in a trusted network environment. The MDM Server provides administration of the mobile device policies and reporting on mobile device behavior. The MDM Server is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status, and sending commands to the Agents. The platform on which the MDM Server software runs is either a general-purpose platform or a network device.”

The MDMAPP states the following:

“The MDM Agent, which is the focus of this EP, is installed on a mobile device as an application or is part of the mobile device’s OS. The MDM Agent establishes a secure connection back to the MDM Server controlled by an enterprise Administrator. Optionally, the MDM Agent interacts with the Mobile Application Store (MAS) Server to download and install enterprise-hosted application.

The MDM Agent must closely interact with or be part of (as depicted by the dotted red/blue line in Figure 1 [of the MDMAPP]) the mobile device’s platform to establish policies and perform queries about device status. The mobile device, in turn, has its own security requirements specified in the MDF PP against which the mobile device must be evaluated either concurrently with or before the MDM Agent evaluation.

If the MDM Agent is part of the mobile device’s OS, the agent may present multiple interfaces for configuring the mobile device, such as a local interface and a remote interface. Agents conforming to this profile must at least offer an interface with a trusted channel that serves as one piece of an MDM system. Compliant agents may also offer other interfaces, and the configuration aspects of these additional interfaces is in scope of this EP.”

The MDM Server component of the TOE is designed to provide centralized management capabilities of the MDM Agent components of the TOE which reside on mobile devices. The MDM Agent communicates with the MDM Server over a secure channel. Therefore the conformance claim is appropriate.

4 Security Problem Definition

4.1 Threats

Note: Unless otherwise stated Threats, Organizational Security Policies, Assumptions and Security Objectives apply to both the Agent and Server. If they are only applicable to the Agent they are denoted by [AGENT] and if only to the Server they are denoted by [SERVER].

This section identifies the threats against the TOE. These threats have been taken from the MDMPP and MDMAPP.

Threat	Threat Definition
T.MALICIOUS_APPS	An administrator of the MDM or mobile device user may inadvertently import malicious code, or an attacker may insert malicious code into the TOE or OE, resulting in the compromise of TOE or TOE data.
T.NETWORK_ATTACK	An attacker may masquerade as MDM Server and attempt to compromise the integrity of the mobile device by sending malicious management commands.
T.NETWORK_EAVESDROP	Unauthorized entities may intercept communications between the MDM and mobile devices to monitor, gain access to, disclose, or alter remote management commands. Unauthorized entities may intercept unprotected wireless communications between the mobile device and the Enterprise to monitor, gain access to, disclose, or alter TOE data.
T.PHYSICAL_ACCESS	The mobile device may be lost or stolen, and an unauthorized individual may attempt to access OE data.

Table 7: TOE Threats

4.2 Organizational Security Policies

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

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

Table 8: TOE Organizational Security Policies

4.3 Assumptions

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

Assumption	Assumption Definition
A.CONNECTIVITY	The TOE relies on network connectivity to carry out its management activities. The TOE will robustly handle instances when connectivity is unavailable or unreliable.
A.MDM_SERVER_PLATFORM	[SERVER] The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities. The MDM server relies on the this platform to provide a range of security-related services including reliable timestamps, user and group account management, logon and logout services via a local or network directory service, remote access control, and audit log management services to include offloading of audit logs to other servers. The platform is expected to be configured specifically to provide MDM services, employing features such as a host-based firewall, which limits its network role to providing MDM functionality.
A.PROPER_ADMIN	One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the TOE Administrators, and do so using and abiding by guidance documentation.
A.PROPER_USER	Mobile device users are not willfully negligent or hostile, and use the device within compliance of a reasonable Enterprise security policy.
A.MOBILE_DEVICE_PLATFORM	[AGENT] The MDM Agent relies upon Mobile platform and hardware evaluated against the MDFPP and assured to provide policy enforcement as well as cryptographic services and data protection. The Mobile platform provides trusted updates and software integrity verification of the MDM Agent.

Table 9: TOE Assumptions

4.4 Security Objectives

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

4.4.1 TOE Security Objectives

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

Objective	Objective Definition
O.APPLY_POLICY	The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the [AGENT: mobile OS and] MDM Agent. This will include the initial enrollment of the device into management, through its lifecycle including policy

	updates and through its possible unenrollment from management services.
O.ACCOUNTABILITY	The TOE must provide logging facilities which record management actions undertaken by its administrators.
O.DATA_PROTECTION_TRANSIT	Data exchanged between the MDM Server and the MDM Agent and between the MDM Server and its operating environment must be protected from being monitored, accessed and altered.
O.MANAGEMENT	[SERVER] The TOE provides access controls around its management functionality.
O.INTEGRITY	[SERVER] The TOE will provide the capability to perform self-tests to ensure the integrity of critical functionality, software/firmware and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates.

Table 10: TOE Objectives

4.4.2 Security Objectives for the Operational Environment

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

Objective	Objective Definition
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	[SERVER] The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities.
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	[SERVER] Reliable timestamp is provided by the operational environment for the TOE.
OE.MOBILE_DEVICE_PLATFORM	[AGENT] 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.

Table 11: Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profiles to which the TOE claims conformance. The associated mappings of assumptions to environmental objectives, SFRs to TOE objectives, and OSPs and objectives to threats are therefore identical to the mappings that are specified in the claimed Protection Profiles.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

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

5.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

6 Security Functional Requirements

6.1 Conventions

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

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized text*.
- **Refinement:** allows the addition of details. Indicated with **bold text**.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR. If the same SFR (including iteration number) is defined in multiple claimed PPs/EPs, the iteration is uniquely identified through the use of a slash followed by a unique identifier for the PP, e.g. /Server or /Agent. Also note that if one claimed PP/EP does not iterate a given SFR but the same SFR appears iterated in another claimed PP/EP, an iteration is added to the original SFR for the purposes of consistency (e.g. FCS_STG_EXT.1 is defined in MDMPP but FCS_STG_EXT.1(2) is defined in MDMAPP, FCS_STG_EXT.1 is therefore changed to be FCS_STG_EXT.1(1)).

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

If SFR text is reproduced verbatim from text that was formatted in a claimed PP (such as if the PP's instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE. SFRs that only apply to the Server are denoted by [SERVER], SFRs that only apply to the Agent are denoted by [AGENT] and SFRs that apply to both are unqualified. Note: the following classes only apply to the Server: Protection of the TSF, TOE Access and Trusted Path/Channels.

Class Name	Component Identification	Component Name
Security Audit	FAU_ALT_EXT.1 [SERVER]	Server Alerts
	FAU_ALT_EXT.2 [AGENT]	Agent Alerts
	FAU_GEN.1(1) [SERVER]	Audit Data Generation
	FAU_GEN.1(2)/Agent [AGENT]	Audit Data Generation
	FAU_GEN.1(2)/Server [SERVER]	Audit Generation (MAS Server)
	FAU_NET_EXT.1 [SERVER]	Network Reachability Review
	FAU_SAR.1 [SERVER]	Audit Review
	FAU_STG_EXT.1(1) [SERVER]	External Audit Trail Storage
	FAU_STG_EXT.1(2) [SERVER]	External Audit Trail Storage (MAS Server)

Class Name	Component Identification	Component Name
	FAU_STG_EXT.2 [SERVER]	Audit Event Storage
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM_EXT.4	Cryptographic Key Destruction
	FCS_COP.1(1)	Cryptographic Operation (Confidentiality Algorithms)
	FCS_COP.1(2)	Cryptographic Operation (Hashing)
	FCS_COP.1(3)	Cryptographic Operation (Digital Signature)
	FCS_COP.1(4)	Cryptographic Operation (Keyed-Hash Message Authentication)
	FCS_HTTPS_EXT.1 [SERVER]	HTTPS Protocol
	FCS_RBG_EXT.1	Extended: Random Bit Generation
	FCS_STG_EXT.1(1) [SERVER]	Cryptographic Key Storage
	FCS_STG_EXT.1(2) [AGENT]	Cryptographic Key Storage
	FCS_TLSC_EXT.1	TLS Client Protocol
	FCS_TLSS_EXT.1 [SERVER]	TLS Server Protocol
Identification and Authentication	FIA_ENR_EXT.1 [SERVER]	Enrollment of Mobile Device into Management
	FIA_ENR_EXT.2 [AGENT]	Enrollment of Mobile Device into Management
	FIA_UAU.1 [SERVER]	Timing of Authentication
	FIA_X509_EXT.1	X509 Validation
	FIA_X509_EXT.2	X509 Authentication
Security Management	FMT_MOF.1(1) [SERVER]	Management of Functions in MDM Server
	FMT_MOF.1(2) [SERVER]	Management of Enrollment Function
	FMT_MOF.1(3) [SERVER]	Management of Functions in MAS Server
	FMT_MOF.1(4) [SERVER]	Management of Download Function in MAS Server
	FMT_POL_EXT.1 [SERVER]	Trusted Policy Update
	FMT_POL_EXT.2 [AGENT]	Trusted Policy Update
	FMT_SMF.1(1) [SERVER]	Specification of Management Functions (Server Configuration of Agent)
	FMT_SMF.1(2) [SERVER]	Specification of Management Functions (Server Configuration of Server)
	FMT_SMF.1(3) [SERVER]	Specification of Management Functions (MAS Server)
	FMT_SMF_EXT.3 [AGENT]	Specification of Management Functions
	FMT_SMR.1(1) [SERVER]	Security Management Roles
	FMT_SMR.1(2) [SERVER]	Security Management Roles
	FMT_UNR_EXT.1 [AGENT]	User Unenrollment Prevention
Protection of the TSF [SERVER]	FPT_ITT.1(1)	Basic Internal TSF Data Transfer Protection (MDM Server)
	FPT_ITT.1(2)	Basic Internal TSF Data Transfer Protection (Distributed TOE)

Class Name	Component Identification	Component Name
	FPT_ITT.1(3)	Basic Internal TSF Data Transfer Protection (MAS Server)
	FPT_TST_EXT.1	TSF Testing
	FPT_TUD_EXT.1	Trusted Update
TOE Access **[SERVER]**	FTA_TAB.1	Default TOE Access Banners
Trusted Path/Channels **[SERVER]**	FTP_ITC.1(1)	Inter-TSF Trusted Channel (Authorized IT Entities)
	FTP_ITC.1(2)	Inter-TSF Trusted Channel (MDM Agent)
	FTP_ITC.1(3)	Inter-TSF Trusted Channel (Authorized IT Entities)
	FTP_TRP.1(1)	Trusted Path for Remote Administration
	FTP_TRP.1(2)	Trusted Path for Enrollment

Table 12: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 Class FAU: Security Audit

6.3.1.1 *[SERVER] FAU_ALT_EXT.1 Server Alerts*

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. [[*detection of blacklisted apps, required apps missing, jailbroken or rooted device, unapproved model/version*]]

6.3.1.2 *[AGENT] FAU_ALT_EXT.2 Agent Alerts*

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. [receiving] periodic reachability events; [

c. change in enrollment state,

d. failure to install an application from the MAS server,

e. failure to update an application from the MAS server,

f. [*detection of blacklisted apps, required apps missing, jailbroken or rooted device, unapproved model/version*]]

FAU_ALT_EXT.2.2

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

6.3.1.3 *[SERVER] FAU_GEN.1(1) Audit Data Generation*

FAU_GEN.1.1(1)

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 **13**; and

e. [*MDM agent alerts generated by FAU_ALT_EXT.2.1 function f*].

FAU_GEN.1.2(1)

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

- date and time of the event,
- type of event,
- subject identity,
- (if relevant) the outcome (success or failure) of the event,
- additional information in Table **13,**
- [*no other information*].

Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_ALT_EXT.1 [SERVER]	Type of alert	Identity of Mobile Device that sent alert
FAU_ALT_EXT.2 [AGENT]	Type of alert.	No additional information
FAU_GEN.1(1) [SERVER]	None.	N/A
FAU_GEN.1(2)/Agent [AGENT]	None.	N/A
FAU_GEN.1(2)/Server [SERVER]	None.	N/A
FAU_NET_EXT.1 [SERVER]	None.	N/A
FAU_SAR.1 [SERVER]	None.	N/A
FAU_STG_EXT.1(1) [SERVER]	None.	N/A
FAU_STG_EXT.1(2) [SERVER]	None.	N/A
FAU_STG_EXT.2 [SERVER]	None.	N/A
FCS_CKM.1	Failure of key generation activity for authentication keys.	No additional information
FCS_CKM.2	None.	N/A
FCS_CKM_EXT.4	None	N/A
FCS_COP.1(1)	None.	N/A
FCS_COP.1(2)	None.	N/A

Requirement	Auditable Event(s)	Additional Audit Record Contents
FCS_COP.1(3)	None.	N/A
FCS_COP.1(4)	None.	N/A
FCS_HTTPS_EXT.1 [SERVER]	Failure of certificate validity check.	No additional information
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information
FCS_STG_EXT.1(1) [SERVER]	None.	N/A
FCS_STG_EXT.1(2) [AGENT]	None	N/A
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 [SERVER]	Failure to establish a TLS session.	Reason for failure.
FIA_ENR_EXT.1 [SERVER]	Failure of MD user authentication.	Presented credentials.
FIA_ENR_EXT.2 [AGENT]	Enrollment in management.	Reference identifier of MDM Server.
FIA_UAU.1 [SERVER]	None.	N/A
FIA_X509_EXT.1	Failure to validate X.509 certificate.	Reason for failure
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	No additional information.
FMT_MOF.1(1) [SERVER]	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) [SERVER]	Enrollment by a user.	Identity of user.
FMT_MOF.1(3) [SERVER]	None.	N/A
FMT_MOF.1(4) [SERVER]	None.	N/A
FMT_POL_EXT.1 [SERVER]	None.	N/A
FMT_POL_EXT.2 [AGENT]	Failure of policy validation.	Reason for failure of validation.
FMT_SMF.1(1) [SERVER]	None.	N/A
FMT_SMF.1(2) [SERVER]	Success or failure of function.	No additional information.
FMT_SMF.1(3) [SERVER]	None.	N/A
FMT_SMF_EXT.3 [AGENT]	Success or failure of function.	No additional information.
FMT_SMR.1(1) [SERVER]	None.	N/A
FMT_SMR.1(2) [SERVER]	None.	N/A
FMT_UNR_EXT.1 [AGENT]	Attempt to unenroll.	No additional information.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FPT_ITT.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. [SERVER] Identity of initiator and recipient. [AGENT] Non-TOE endpoint of connection.
FPT_ITT.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. [SERVER] Identity of initiator and recipient. [AGENT] Non-TOE endpoint of connection.
FPT_ITT.1(3)	None.	N/A
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_TUD_EXT.1	Success of failure of signature verification.	No additional information.
FTA_TAB.1	Change in banner setting.	No additional information.
FTP_ITC.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint connection.
FTP_ITC.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint connection.
FTP_ITC.1(3)	None.	N/A
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 13: Agent/Server Auditable Events

Application Note: The table above is merged from the auditable events tables in the MDM Server PP and MDM Agent EP. The SFRs that belong to a particular PP are denoted with [AGENT] or [SERVER] as is the case with how these SFRs are identified elsewhere in this ST. All SFRs in the FPT, FTA, and FTP classes are applicable to the MDM Server component. Any other SFRs that are not denoted with a particular PP are applicable to both claimed PPs.

6.3.1.4 *[AGENT] FAU_GEN.1(2)/Agent Audit Data Generation*

FAU_GEN.1.1(2)/Agent

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

- Start-up and Shutdown of the audit functions;
- Change in MDM policy; and
- Any modification commanded by the MDM Server,
- Specifically defined auditable events listed in Table **13,**
- [*no other events*].

FAU_GEN.1.2(2)/Agent

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

- date and time of the event,
- type of event,
- subject identity,
- (if relevant) the outcome (success or failure) of the event,
- additional information in Table **13,**
- [*no other events*].

6.3.1.5 *[SERVER] FAU_GEN.1(2)/Server Audit Generation (MAS Server)*

FAU_GEN.1.1(2)/Server

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

a. Failure to push a new application on a managed mobile device;

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

FAU_GEN.1.2(2)/Server

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

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

6.3.1.6 *[SERVER] FAU_NET_EXT.1 Network Reachability Review*

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.

6.3.1.7 *[SERVER] FAU_SAR.1 Audit Review*

FAU_SAR.1.1

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

FAU_SAR.1.2

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

6.3.1.8 *[SERVER] FAU_STG_EXT.1(1) External Audit Trail Storage*

FAU_STG_EXT.1.1(1)

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

6.3.1.9 *[SERVER] FAU_STG_EXT.1(2) External Audit Trail Storage (MAS Server)*

FAU_STG_EXT.1.1(2)

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

6.3.1.10 *[SERVER] FAU_STG_EXT.2 Audit Event Storage*

FAU_STG_EXT.2.1

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

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)*

FCS_CKM.1.1

The [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 [no other curves] 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].

Application Note: Only the Server component is capable of generating FFC keys.

6.3.2.2 *FCS_CKM.2 Cryptographic Key Establishment*

FCS_CKM.2.1

The [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”;

].

Application Note: Only the Server component is capable of performing finite field-based key establishment.

6.3.2.3 *FCS_CKM_EXT.4 Cryptographic Key Destruction*

FCS_CKM_EXT.4.1

The [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 block erase 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.

6.3.2.4 *FCS_COP.1(1) Cryptographic Operation (Confidentiality Algorithms)*

FCS_COP.1.1(1)

The [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].

6.3.2.5 *FCS_COP.1(2) Cryptographic Operation (Hashing)*

FCS_COP.1.1(2)

The [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.

6.3.2.6 *FCS_COP.1(3) Cryptographic Operation (Digital Signature)*

FCS_COP.1.1(3)

The [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 [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5].

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

FCS_COP.1.1(4)

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

6.3.2.8 *[SERVER] FCS_HTTPS_EXT.1 HTTPS Protocol*

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

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

FCS_HTTPS_EXT.1.3

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

6.3.2.9 *FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)*

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

6.3.2.10 *FCS_TLSC_EXT.1 TLS Client Protocol*

FCS_TLSC_EXT.1.1

The [TOE platform] shall implement [TLS 1.0 (RFC 3246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [Optional Ciphersuites:
  - 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_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
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSC_EXT.1.2

The [TOE platform] shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The [TOE platform] shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.1.4

The [TOE platform] shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.1.5

The [TOE platform] shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1] and no other curves.

6.3.2.11 *[SERVER] FCS_TLSS_EXT.1 TLS Server Protocol*

FCS_TLSS_EXT.1.1

The [MDM Server platform] shall implement [TLS 1.0 (RFC 3246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [Optional Ciphersuites:
  - 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_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
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSS_EXT.1.2

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

FCS_TLSS_EXT.1.4

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

FCS_TLSS_EXT.1.5

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

6.3.2.12 *[SERVER] FCS_STG_EXT.1(1) Cryptographic Key Storage*

FCS_STG_EXT.1.1(1)

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

6.3.2.13 *[AGENT] FCS_STG_EXT.1(2) Cryptographic Key Storage*

FCS_STG_EXT.1.1(2)

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

6.3.3 Class FIA: Identification and Authentication

6.3.3.1 *[SERVER] FIA_ENR_EXT.1 Enrollment of Mobile Device into Management*

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 [specific devices].

6.3.3.2 *[AGENT] FIA_ENR_EXT.2 Enrollment of Mobile Device into Management*

FIA_ENR_EXT.2.1

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

6.3.3.3 *[SERVER] FIA_UAU.1 Timing of Authentication*

FIA_UAU.1.1

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

FIA_UAU.1.2

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.

6.3.3.4 *FIA_X509_EXT.1 X509 Validation*

FIA_X509_EXT.1.1

The [TOE platform] 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:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

6.3.3.5 *FIA_X509_EXT.2 X509 Authentication*

FIA_X509_EXT.2.1

The [TOE platform] shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS, HTTPS], and [code signing for system software updates, code signing for integrity verification, policy signing].

FIA_X509_EXT.2.2

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

FIA_X509_EXT.2.3

The [TOE platform] shall require a unique certificate for each client device.

6.3.4 Class FMT: Security Management

6.3.4.1 *[SERVER] FMT_MOF.1(1) Management of Functions in MDM Server*

FMT_MOF.1.1(1)

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)
- listed in FMT_SMF.1(2)

to Authorized Administrators.

6.3.4.2 *[SERVER] FMT_MOF.1(2) Management of Enrollment Function*

FMT_MOF.1.1(2)

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

6.3.4.3 *[SERVER] FMT_MOF.1(3) Management of Functions in MAS Server*

FMT_MOF.1.1(3)

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

6.3.4.4 *[SERVER] FMT_MOF.1(4) Management of Download Function in MAS Server*

FMT_MOF.1.1(4)

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.

6.3.4.5 *[SERVER] FMT_POL_EXT.1 Trusted Policy Update*

FMT_POL_EXT.1.1

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

6.3.4.6 *[AGENT] FMT_POL_EXT.2 Trusted Policy Update*

FMT_POL_EXT.2.1

The MDM Agent shall only accept policies and policy updates digitally signed by the Enterprise.

FMT_POL_EXT.2.2

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

6.3.4.7 *[SERVER] FMT_SMF.1(1) Specification of Management Functions*

FMT_SMF.1.1(1)

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: [

<u>14. wipe Enterprise data (MDM Function 28)</u>
<u>15. remove imported X.509v3 certificates and [*no other X.509v3 certificates*] in the Trust Anchor Database, (MDF Function 14)</u>
<u>16. alert the administrator,</u>
<u>22. place applications into application process groups based on [*profile assignment*] (MDF Function 43),</u>
<u>23. [*no other management functions*]]</u>

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 certificates]
    b. ability to specify security type
    c. ability to specify authentication protocol
    d. specify the client credentials to be used for authentication
    e. [*no other WLAN management functions*] (MDF Function 7)
28. application installation policy by [
    b. specifying a set of allowed applications and versions (an application whitelist)]
    (MDF Function 10)
29. enable/disable policy for [*camera, microphone*] across MD, [no other method], (MDF Function 5)

and the following MD configuration policies: [

30. enable/disable policy for the VPN across MD, [on a per-app basis], (MDF Function 3)
35. enable policy for data-at-rest protection, (MDF Function 25)
49. enable/disable backup to [remote system] (MDF Function 40)
53. *the following additional policies:*
    b. enable/disable authentication mechanisms providing user access to protected data other than a Password Authentication Factor (e.g., using a fingerprint);
    c. policies for which there are required configuration values in the mobile operating system STIG relevant to the MD.
    d. full wipe of all user data and applications (applications not included in the out-of-the-box install).

].

6.3.4.8 *[SERVER] FMT_SMF.1(2) Specification of Management Functions*

FMT_SMF.1.1(2)

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

a. configure X.509v3 certificates for MDM Server use

b. configure the [specific devices] allowed for enrollment [

d. configure the TOE unlock banner,

e. configure periodicity of the following commands to the agent: [*query connectivity status, query the current version of the MD firmware/software, query the current version of the hardware model of the device, query the current version of installed mobile applications, read audit logs kept by the MD*],

f. [*store MD audit logs, transfer MD audit logs to another server*]].

6.3.4.9 *[SERVER] FMT_SMF.1(3) Specification of Management Functions (MAS Server)*

FMT_SMF.1.1(3)

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

a. Configure application access groups,

b. Download applications,

c. [no other functions].

6.3.4.10 *[AGENT] FMT_SMF_EXT.3 Specification of Management Functions*

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].

6.3.4.11 *[SERVER] FMT_SMR.1(1) Security Management Roles*

FMT_SMR.1.1(1)

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

FMT_SMR.1.2(1)

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

6.3.4.12 *[SERVER] FMT_SMR.1(2) Security Management Roles*

FMT_SMR.1.1(2)

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)

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

6.3.4.13 *[AGENT] FMT_UNR_EXT.1 User Unrollment Prevention*

FMT_UNR_EXT.1.1

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

6.3.5 Class FPT: Protection of the TSF

6.3.5.1 *[SERVER] FPT_ITT.1(1) Basic Internal TSF Data Protection (MDM Server)*

FPT_ITT.1.1(1)

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

6.3.5.2 *[SERVER] FPT_ITT.1(2) Basic Internal TSF Data Protection (Distributed TOE)*

FPT_ITT.1.1(2)

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

6.3.5.3 *[SERVER] FPT_ITT.1(3) Basic Internal TSF Data Protection (MAS Server)*

FPT_ITT.1.1(3)

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

6.3.5.4 *[SERVER] FPT_TST_EXT.1 TSF Testing*

FPT_TST_EXT.1.1

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

6.3.5.5 *[SERVER] FPT_TUD_EXT.1 Trusted Update*

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 platform] shall provide Authorized Administrators the ability to initiate updates to TSF software.

FPT_TUD_EXT.1.3

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

6.3.6 Class FTA: TOE Access

6.3.6.1 *[SERVER] FTA_TAB.1 Default TOE Access Banners*

FTA_TAB.1.1

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

6.3.7 Class FTP: Trusted Path/Channels

6.3.7.1 *[SERVER] FTP_ITC.1(1) Inter-TSF Trusted Channel*

FTP_ITC.1.1(1)

The [MDM Server platform] shall use [TLS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [authentication server, [*database 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.1.2(1)

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

FTP_ITC.1.3(1)

The TSF shall initiate communication via the trusted channel for [*connection to audit server, authentication server, database server*].

6.3.7.2 *[SERVER] FTP_ITC.1(2) Inter-TSF Trusted Channel (MDM Agent)*

FTP_ITC.1.1(2)

The TSF shall use [TLS, HTTPS] to provide a trusted communication channel between itself and another trusted IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

FTP_ITC.1.2(2)

The TSF shall permit the TSF and MDM Agent to initiate communication via the trusted channel.

FTP_ITC.1.3(2)

The TSF shall initiate communication via the trusted channel for all communication between the MDM Server and the MDM Agent and [no other communication].

6.3.7.3 *[SERVER] FTP_ITC.1(3) Inter-TSF Trusted Channel (Authorized IT Entities)*

FTP_ITC.1.1(3)

The [MAS Server platform] shall use [TLS, TLS/HTTPS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [authentication server, [*database 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.

6.3.7.4 *[SERVER] FTP_TRP.1(1) Trusted Path for Remote Administration*

FTP_TRP.1.1(1)

The [MDM Server platform] shall use [TLS/HTTPS] 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.1.2(1)

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

FTP_TRP.1.3(1)

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

6.3.7.5 *[SERVER] FTP_TRP.1(2) Trusted Path for Enrollment*

FTP_TRP.1.1(2)

The [MDM Server platform] 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)

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

FTP_TRP.1.3(2)

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

6.4 Statement of Security Functional Requirements Consistency

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

7 Security Assurance Requirements

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

7.1 Class ADV: Development

7.1.1 Basic Functional Specification (ADV_FSP.1)

7.1.1.1 *Developer action elements:*

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

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

7.1.1.2 *Content and presentation elements:*

ADV_FSP.1.1C

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

ADV_FSP.1.2C

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

ADV_FSP.1.3C

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

ADV_FSP.1.4C

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

7.1.1.3 *Evaluator action elements:*

ADV_ FSP.1.1E

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

ADV_ FSP.1.2E

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

7.2 Class AGD: Guidance Documentation

7.2.1 Operational User Guidance (AGD_OPE.1)

7.2.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.2.1.2 *Content and presentation elements:*

AGD_OPE.1.1C

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

AGD_OPE.1.2C

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

AGD_OPE.1.3C

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

AGD_OPE.1.4C

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

AGD_OPE.1.5C

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

AGD_OPE.1.6C

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

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

7.2.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

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

7.2.2 Preparative Procedures (AGD_PRE.1)

7.2.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.2.2.2 *Content and presentation elements:*

AGD_ PRE.1.1C

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

AGD_ PRE.1.2C

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

7.2.2.3 *Evaluator action elements:*

AGD_ PRE.1.1E

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

AGD_ PRE.1.2E

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

7.3 Class ALC: Life Cycle Support

7.3.1 Labeling of the TOE (ALC_CMC.1)

7.3.1.1 *Developer action elements:*

ALC_CMC.1.1D

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

7.3.1.2 *Content and presentation elements:*

ALC_CMC.1.1C

The TOE shall be labeled with its unique reference.

7.3.1.3 *Evaluator action elements:*

ALC_CMC.1.1E

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

7.3.2 TOE CM Coverage (ALC_CMS.1)

7.3.2.1 *Developer action elements:*

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

7.3.2.2 *Content and presentation elements:*

ALC_CMS.1.1C

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

ALC_CMS.1.2C

The configuration list shall uniquely identify the configuration items.

7.3.2.3 *Evaluator action elements:*

ALC_CMS.1.1E

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

7.4 Class ATE: Tests

7.4.1 Independent Testing - Conformance (ATE_IND.1)

7.4.1.1 *Developer action elements:*

ATE_IND.1.1D

The developer shall provide the TOE for testing.

7.4.1.2 *Content and presentation elements:*

ATE_IND.1.1C

The TOE shall be suitable for testing.

7.4.1.3 *Evaluator action elements:*

ATE_IND.1.1E

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

ATE_IND.1.2E

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

7.5 Class AVA: Vulnerability Assessment

7.5.1 Vulnerability Survey (AVA_VAN.1)

7.5.1.1 *Developer action elements:*

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 *Content and presentation elements:*

AVA_VAN.1.1C

The TOE shall be suitable for testing.

7.5.1.3 *Evaluator action elements:*

AVA_VAN.1.1E

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

AVA_VAN.1.2E

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

AVA_VAN.1.3E

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

8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Security Audit, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access, and Trusted Path/Channels.

Note: If a TSS SFR section only applies to the Server it is denoted by [SERVER], if only to the Agent it is denoted by [AGENT] and if both Server and Agent then it is not qualified. All requirements for Protection of the TSF, TOE Access, and Trusted Path/Channels apply only to the Server component.

8.1 Security Audit

8.1.1 [SERVER] FAU_ALT_EXT.1:

The MDM Server component of the TOE provides administrators with the ability to view information about enrolled mobile devices and to generate alerts when various events occur.

TOE Administrators can view information about the status of managed devices through the AirWatch Admin Console. Two of the dashboards that are accessible from the Main Menu are “Hub” and “Devices”. Under “Hub”, administrators can view the total number of enrolled and unenrolled devices, the total number of compliance violations and which devices have blacklisted apps, devices without required apps, or devices with apps that are not whitelisted. The administrator can also view the applications that are associated with particular devices, including application versions. Under “Devices”, the administrator can view changes in the enrollment status of a device by viewing the enrollment status and enrollment history information. This also lists devices that are enrolled but do not have policies applied to them. The TOE can also view the list of compromised (jailbroken/rooted) devices under this section of the Admin Console as well as detailed information about any specific device that the Server knows about.

In addition to being able to review this information on demand, administrators can configure the delivery of periodic (daily, weekly, monthly) alert emails from the “Hub” section of the Admin Console. Alerts are generated based on configurable “compliance policies” that can detect when certain events are observed on a device:

- Presence of blacklisted apps
- Presence of non-whitelisted apps
- Absence of required apps
- Compromised (jailbroken or rooted) device
- Last time a device communicated with the MDM Server
- Unapproved model
- Unapproved operating system version (greater than, less than, equal to, not equal to specified version)

Any compliance policy violations that are related to apps can be applied to apps that are maintained by the MAS Server component of the TOE. In addition to displaying on the Hub, a compliance policy can be configured to send an email when a violation is detected and to mark the affected device as Not Compliant in the devices overview of the Admin Console.

8.1.2 [AGENT] FAU_ALT_EXT.2:

The Agent component of the TOE provides the ability to alert the MDM Server in the event that certain behavior on the underlying mobile device is observed.

For iOS devices, most of the alerting is done through the iOS MDM protocol rather than through the TOE. The TOE is responsible for alerting the Server component when a jailbreak is detected and when the device is unenrolled from management. The remaining alerts are generated by the underlying platform. Jailbreak detection is performed when the mobile device user launches an app with the AirWatch jailbreak detection capability. In the event that a jailbreak is detected and the Agent is unable to communicate with the Server, the notification is continuously retried until communications have been re-established.

For iOS applications, the MAS Server has the ability to specify if a given application is required, blacklisted, or whitelisted. This assignment can be made both for apps under the control of the MAS Server as well as publicly-available apps on the Apple Store. iOS does not provide the ability to force a device to receive a required app from the MAS Server, so alerts are only generated if a compliance policy is written to detect the presence of blacklisted or non-whitelisted apps, or the absence of required apps. This is detected through the TSF collecting a list of the installed applications on the device during the periodic check-in.

8.1.3 [SERVER] FAU_GEN.1(1):

The Server component of the TOE generates auditable events for its own behavior. Some of these activities are generated and stored within the TOE boundary while others are logged as events on the underlying Windows Server 2012 platform. The MDM Server generates audit logs for its own startup and shutdown, all administrative actions, and all commands that are sent to managed devices from the MDM Server. The TSF and the underlying OS platform also generate audit data for the specific auditable events listed in the table below. The type of event and specific audit record contents are described in Table 13.

Requirement	Auditable Event(s)	Audit Method
FAU_ALT_EXT.1 [SERVER]	Type of alert	TOE
FCS_CKM.1	Failure of key generation activity for authentication keys.	Platform
FCS_HTTPS_EXT.1 [SERVER]	Failure of certificate validity check.	Platform
FCS_RBG_EXT.1	Failure of the randomization process.	Platform
FCS_TLSC_EXT.1	Failure to establish a TLS session. Failure to verify presented identifier. In addition [AGENT] also has: Establishment/termination of a TLS session.	Platform

Requirement	Auditable Event(s)	Audit Method
FCS_TLSS_EXT.1 [SERVER]	Failure to establish a TLS session.	Platform
FIA_ENR_EXT.1 [SERVER]	Failure of MD user authentication.	TOE
FIA_X509_EXT.1	Failure to validate X.509 certificate.	Platform
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	Platform
FMT_MOF.1(1) [SERVER]	Issuance of command to perform function. Change of policy settings.	TOE
FMT_MOF.1(2) [SERVER]	Enrollment by a user.	TOE
FMT_SMF.1(2) [SERVER]	Success or failure of function.	TOE
FPT_ITT.1(1)	Initiation and termination of the trusted channel.	Platform
FPT_ITT.1(2)	Initiation and termination of the trusted channel.	Platform
FPT_TST_EXT.1	Initiation of self-test. Failure of self-test. Detected integrity violation.	Platform
FPT_TUD_EXT.1	Success of failure of signature verification.	Platform
FTA_TAB.1	Change in banner setting.	TOE
FTP_ITC.1(1)	Initiation and termination of the trusted channel.	Platform
FTP_ITC.1(2)	Initiation and termination of the trusted channel.	Platform
FTP_TRP.1(1)	Initiation and termination of the trusted channel.	Platform
FTP_TRP.1(2)	Initiation and termination of the trusted channel.	Platform

Table 14: Server Auditable Events by Enforcing Component

The following table provides example audit records for each of the server auditable events that are audited by the TOE in order to illustrate the audit record format and the fields contained within these records. **Bold text** is used to highlight the specific event type in order to show correspondence with the SFR.

Requirement	Sample Record
FAU_ALT_EXT.1 [SERVER]	Nov 2 10:51:44 172.16.72.19 November 02 14:51:44 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: **MDMEnrollmentComplete**User: sysadminEvent Source: ServerEvent Module: EnrollmentEvent Category: EnrollmentEvent Data: Device Friendly Name: niapuser1 iPad iOS 9.3.5 FP84Enrollment User: niapuser1
FIA_ENR_EXT.1 [SERVER]	(in GUI) Information / 11/17/2016 10:43 AM / Device / Enrollment / Authentication / **User Enrollment Authentication Failure** / User Enrollment Name : chris4

Requirement	Sample Record
FMT_MOF.1(1) [SERVER]	Nov 3 09:02:09 172.16.72.19 November 03 13:02:10 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: **DeviceLockConfirmed**User: sysadminEvent Source: DeviceEvent Module: DevicesEvent Category: CommandEvent Data: Device Friendly Name: niapuser1 iPad iOS 9.3.5 FP84Enrollment User: niapuser1 3;EnableGeofencing=N/A;EnableScheduling=N/A;RemovalDate=N/Aawsso\JoBrownawsso\JoBrownN/AProfileModifiedawsso\JoBrown
FMT_MOF.1(2) [SERVER]	Nov 10 09:20:28 172.16.72.19 November 10 14:20:29 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: **MDMEnrollmentComplete**User: sysadminEvent Source: ServerEvent Module: EnrollmentEvent Category: EnrollmentEvent Data: Device Friendly Name: niaptestAD iPad iOS 9.3.4 FP84Enrollment User: niaptestAD
FMT_SMF.1(2) [SERVER]	Nov 10 16:20:36 172.16.72.18 November 10 21:20:37 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: ConsoleEvent: **AppleMdmSampleScheduleSettingChangedSuccess**User: AdministratorEvent Source: ServerEvent Module: AdministrationEvent Category: SystemSettingsEvent Data: LoginSessionID=bigij2deztokDevice Friendly Name: N/AEnrollment User: N/A
FTA_TAB.1	Nov 15 13:42:32 172.16.72.18 November 15 18:42:33 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: ConsoleEvent: **BrandingChanged**User: AdministratorEvent Source: ServerEvent Module: SettingsEvent Category: SystemSettingsEvent Data: LoginSessionID=mnwzy1x4m5ioDevice Friendly Name: N/AEnrollment User: N/A

Table 15: Sample Server Auditable Events

8.1.4 [AGENT] FAU_GEN.1(2)/Agent:

The MDM Agent component of the TOE generates audit events for its own activities. Specifically, all activities that are listed in Table 13 that are signified with [AGENT] are audited by the TSF. SFRs that are implemented by both the Agent and the Server (e.g. FCS_CKM.1, FCS_TLSC_EXT.1) are performed by the underlying mobile device platform, so the platform is responsible for logging of these functions as well. The MDM Agent also audits startup and shutdown of itself, any change in policy, and any modification that is directed by the MDM Server.

The table below lists the SFRs that are relevant to the MDM Agent's functionality that have auditable events associated with them and are audited by the TSF. It also provides sample audit records to illustrate the format and contents of this data. Note that since some iOS management functions are enforced by the iOS native MDM agent rather than the TOE, the agent logging for those events are not within the scope of the TSF.

Requirement	Sample Record
FAU_ALT_EXT.2 [AGENT]	Nov 9 15:45:03 172.16.72.18 November 09 20:45:05 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: ConsoleEvent: DeviceQueryRequestedUser: testadminEvent Source: ServerEvent Module: DeviceDetailsEvent Category: DeviceEvent Data: Device=niaptestAD iPad iOS 9.3.4 FP84;LoginSessionID=mchytqm4w4wdDevice Friendly Name: niaptestAD iPad iOS 9.3.4 FP84Enrollment User: niaptestAD
FIA_ENR_EXT.2 [AGENT]	Nov 10 09:20:28 172.16.72.19 November 10 14:20:29 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: MDMEnrollmentCompleteUser: sysadminEvent

	Source: ServerEvent Module: EnrollmentEvent Category: EnrollmentEvent Data: Device Friendly Name: niaptestAD iPad iOS 9.3.4 FP84Enrollment User: niaptestAD
FMT_POL_EXT.2 [AGENT]	Nov 10 09:20:37 172.16.72.19 November 10 14:20:38 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: InstallProfileFailedUser: sysadminEvent Source: ServerEvent Module: DevicesEvent Category: CommandEvent Data: ErrorCode=1000 Invalid Profile;Profile=FAU_ALT_EXT.1.1 - 002 Part 2Device Friendly Name: niaptestAD iPad iOS 9.3.4 FP84Enrollment User: niaptestAD
FMT_SMF_EXT.3 [AGENT]	Nov 4 07:52:20 172.16.72.19 November 04 11:52:21 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: CertificateListSampleConfirmedUser: sysadminEvent Source: DeviceEvent Module: DevicesEvent Category: CommandEvent Data: Device Friendly Name: niap iPad iOS 10.0.0 HGJ1Enrollment User: niap
FMT_UNR_EXT.1 [AGENT]	Nov 11 09:15:53 172.16.72.19 November 11 14:15:55 AirWatch NIAP AirWatch Syslog Details are as follows Event Type: DeviceEvent: BreakMDMConfirmedUser: sysadminEvent Source: DeviceEvent Module: DevicesEvent Category: CommandEvent Data: Device Friendly Name: niaptestlimit iPad iOS 9.3.4 FP84Enrollment User: niaptestlimit

Table 16: Sample Agent Auditable Events

8.1.5 [SERVER] FAU_GEN.1(2)/Server:

The MAS Server component of the TOE generates audit records when it is unable to push an application to a managed device or detects that a required application is not present on the device. This is done by defining a compliance policy that checks for the absence of a specific application which causes the device to generate an audit event on its next periodic check-in if that condition is met. iOS does not provide a mechanism to allow the MAS Server to push an application to the device but an audit event will be generated if the TSF detects that a required app is not present. The following is a sample record of a failed MAS operation:

```
Nov  3 09:33:57 172.16.72.19 November 03 13:33:57 AirWatch NIAP  AirWatch Syslog Details are as follows Event Type: DeviceEvent: InstallApplicationFailedUser: sysadminEvent Source: ServerEvent Module: DevicesEvent Category: CommandEvent Data: ErrorCode=Pending;Application=;ApplicationVersion=;ApplicationType=;BytesReceived=0Device Friendly Name: niapuser1 iPad iOS 9.3.5 FP84Enrollment User: niapuser1
```

8.1.6 [SERVER] FAU_NET_EXT.1:

Authorized Administrators can use the Admin Console to determine the network connectivity status of devices that have MDM Agents installed on them. An ad hoc query for a specific device is initiated from the Devices dashboard. In this query, the Hit result provides the last time that the device connected to the MDM Server.

An administrator can also make an on-demand request to get connectivity information from one or more devices simultaneously using push notifications. For iOS devices, the TSF uses the Apple Push Notification service (APN) and responses are sent back to the TOE via the platform's messaging service. Refer to the discussion on FAU_ALT_EXT.1 for more information about how the MDM Agents communicate device status back to the MDM Server.

Periodic reachability events will be initiated by the MDM Agent and ad hoc requests for connectivity status will be initiated by the MDM Server.

8.1.7 [SERVER] FAU_SAR.1:

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

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

Table 14 shows the auditable events that are logged by the TSF versus the underlying platform. The component used to review the audit data is the same as the component that is used to generate the data to be reviewed.

8.1.8 [SERVER] FAU_STG_EXT.1(1):

AirWatch Console audit data can be transmitted from the MDM Server to a Syslog Server over a TLS 1.0 encrypted trusted channel. The MDM Server can be configured to send syslog data over a specific port and if this matches the syslog TLS port, then the Syslog Server will respond with a TLS connection initiation. The Syslog certificate is bound to the port, which is defaulted to port 443 but can be changed by a System Administrator. Once the connection is established, the audit logs are sent to the Syslog Server. Each time an event is created, the audit data is sent to the Syslog Server. The actual TLS encryption is handled by the underlying Windows Server platform.

As stated in Table 14, some audit data is generated by the underlying platform and is not stored within the TOE boundary. It is therefore the responsibility of the Operational Environment to securely transfer this data to a remote location for permanent storage.

8.1.9 [SERVER] FAU_STG_EXT.1(2):

The MAS Server is an integrated component of the MDM Server and does not exist as a distinct entity. Therefore, the same TLS syslog channel that is used to export MDM Server data is used for any audit records that are related to the MAS Server's behavior.

8.1.10 [SERVER] FAU_STG_EXT.2:

An SQL database is pre-installed on the MDM Server platform. The MDM Server creates tables inside of the SQL database where audit records can be stored during initial setup of the TOE. The operating system on the MDM Server is responsible for securing the audit records stored in the database. The SQL roles of db_owner/sql_admin are required in order to modify the audit records. AirWatch Console Administrators do not have direct access to the database.

8.2 Cryptographic Support

On the MDM Server, the TOE and TOE platform use Microsoft's BCryptPrimitives.dll and CNG.sys cryptographic modules on the underlying Windows Server 2012 platform to provide all of the cryptographic algorithms. Both of these have been FIPS 140-2 validated with CMVP certificates #2357 and #2356, respectively. All of the CAVP algorithm certificates listed below correspond to these CMVP certificates. Unless otherwise specified, BCryptPrimitives.dll and CNG.sys use the same underlying algorithm implementation so their CAVP algorithm certificates do not differ.

Cryptographic services for the MDM Agent are provided by the underlying platform. The MDM Agent uses the Apple iOS CoreCrypto Module (CMVP certificate #2594) to perform the cryptographic functionality it requires in order to satisfy the claimed SFRs.

For more information about the cryptographic functionality provided by these modules and their corresponding cryptographic algorithm certificates, refer to the Security Targets for the underlying platforms listed in section 1.1.5.

8.2.1 FCS_CKM.1:

[SERVER] RSA, Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) public/private key pairs are used in TLS for key establishment. This functionality is implemented by the MDM Server's underlying OS platform and is invoked by the MDM Server as needed. The RSA-2048 bit key pairs are generated according to FIPS PUB 186-4 using CAVP RSA cert. #1487. The ECDH ECC schemes use NIST curves P-256 and P-384 for ECDH and are generated according to FIPS PUB 186-4 with CAVP ECDSA cert. #505. The DH FFC schemes use key sizes of 2048 and 3072 bits and are generated according to FIPS PUB 186-4 with CAVP DSA Cert. #855 (done by CNG.sys only).

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform asymmetric key generation for key establishment. Both RSA and ECDSA key generation functionality conform to FIPS PUB 186-4. The CAVP certificates for the RSA and ECDSA key generation functions for the iOS platform are the same as the certificates for signature functions. These certificates are defined in section 8.2.6.

8.2.2 FCS_CKM.2:

[SERVER] The underlying platform of the MDM Server supports three types of key establishment techniques used in the TLS protocol: RSA, DH, and ECDH. The RSA key establishment implementation conforms to NIST SP 800-56B and is vendor asserted affirmed under FIPS 140-2 IG D.4. The DH and ECDH key establishment methods both conform to SP 800-56A with CAVP KAS Cert. #47.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform key establishment using ECDSA. The mobile device platform claims to implement ECDSA key establishment in accordance with NIST SP 800-56A.

8.2.3 FCS_CKM_EXT.4:

[SERVER] The MDM Server relies on the underlying FIPS cryptographic modules to zeroize keys and cryptographic security parameter data when no longer needed. All key data maintained by the server platform exists only in volatile memory and is erased by a one-pass overwrite with zeroes followed by a read-verify.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform key destruction. The specific cryptographic implementation for the iOS platform can be found in the Apple Security Target documentation referenced in section 1.1.5 of this document.

8.2.4 FCS_COP.1(1):

[SERVER] The MDM Server relies on the underlying OS platform to perform AES encryption/decryption services for TLS communications and protection of data at rest in platform key storage. All data at rest is protected using AES-CBC-256 as defined in FIPS PUB 197 and NIST SP 800-38A. Data in transit is protected using either of CBC or GCM modes and either 128-bit or 256-bit keys. When operating in GCM mode, the TOE conforms to NIST SP 800-38D. The AES implementation used by the TOE is certified under CAVP, AES certificate #2832.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform symmetric encryption/decryption. Similar to the MDM Server component, the MDM Agent uses 256-bit AES-CBC or 128-bit AES-GCM. These implementations conform to NIST SP 800-38A and 800-38D, respectively. The CAVP certificates for the AES implementation are #3686 for iOS 9 and #4255 for iOS 10.

8.2.5 FCS_COP.1(2):

[SERVER] The MDM Server relies on the TOE Platform to provide SHA-1, SHA-256, SHA-384 and SHA-512 cryptographic hashing services, conformant to FIPS PUB 180-4, with digest sizes of 160, 256, 384 and 512 bits, respectively. SHA-1, SHA-256 and SHA-384 are used by HMAC in message authentication in TLS. SHA-512 is used in the hashing of passwords. SHA-512 is also used by ECDSA in the digital signing of policies. The SHA implementation has CAVP SHS cert. #2373.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform cryptographic hashing. Similar to the MDM Server component, the MDM Agent uses either of SHA-256 or SHA-384, conformant to FIPS PUB 180-4. The CAVP certificates for the SHS implementation are #3100 for iOS 9 and #3531 for iOS 10.

8.2.6 FCS_COP.1(3):

[SERVER] The MDM Server relies on the TOE platform to provide digital signature services in accordance with FIPS PUB 186-4. The TSF uses RSA with 2048 bit keys for its digital signature generation and verification, including the verification of certificates. The TOE also uses ECDSA for digitally signing and verifying policies, with support for both P-256 and P-384. RSA has CAVP RSA Cert. #1487, 1493 and 1519, and ECDSA has CAVP ECDSA Cert. #505.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform digital signature services. Similar to the MDM Server component, the MDM Agent uses either of RSA (with 2048-bit keys) or ECDSA (with P-256 or P-384) conformant to FIPS PUB 186-4. The CAVP certificates for the RSA implementation are #1908 for iOS 9 and #2341 for iOS 10. The CAVP certificates for the ECDSA implementation are #781 or iOS 9 and #1003 for iOS 10.

8.2.7 FCS_COP.1(4):

[SERVER] The TOE platform uses HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384, conformant to FIPS PUBs 180-1 and 198-1, with respective digest sizes of 160, 256 and 384 bits. The key used by HMAC is the UUID which is 160 bits. This key is stored on both the Server and the Agent platforms. HMAC has CAVP HMAC cert. #1773.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to perform keyed-hash message authentication. Similar to the MDM Server component, the MDM Agent uses either of HMAC-SHA-256 or HMAC-SHA-384 in accordance with FIPS PUBs 180-1 and 198-1. The CAVP certificates for the HMAC implementation are #2432 for iOS 9 and #2829 for iOS 10. As stated previously, the 160-bit UUID used as the HMAC key is stored on the underlying platform of the Agent.

8.2.8 [SERVER] FCS_HTTPS_EXT.1

HTTPS communication on the MDM Server platform is compliant with RFC 2818 and is handled by IIS 8.5 (although versions as low as IIS 7.5 are also supported) on the Windows 2012 R2 Server. If the peer certificate is invalid, the connection is not established.

8.2.9 FCS_RBG_EXT.1:

[SERVER] The MDM Server relies on the underlying OS platform to provide random bit generation services. The platform cryptographic module provides an AES counter DRBG (CTR_DRBG) that conforms to NIST SP 800-90A. This implementation has CAVP DRBG certificate #489. In order to provide sufficient randomness to the keys generated by this DRBG (as specified in NIST SP 800-57), the RBG is seeded with at least 256 bits of entropy which is gathered from the environmental hardware by the underlying OS.

[AGENT] The MDM Agent software relies on the underlying mobile device platform to provide random bit generation services. The iOS platform cryptographic implementation supports the AES counter DRBG conformant to NIST SP 800-90A. The CAVP certificates for the HMAC implementation are #993 for iOS 9 and #1353 for iOS 10.

8.2.10 [SERVER] FCS_STG_EXT.1(1):

The TOE platform is responsible for storing keys. All private keys are stored in the Windows Trust Store and all user credentials are stored in authentication repositories. The database Master Key is stored encrypted using an AES-CBC key encryption key (KEK). This KEK is encrypted and stored in the registry.

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

Key/CSP	Purpose
AES-CBC	Data at rest Data in transit
AES-GCM	Data in transit
RSA Public/Private Key	TLS Key Establishment
RSA Public/Private Key	Digital Signatures
DH Public/Private Key	TLS Key Establishment
DSA Public/Private Key	KeyGen for DH
ECDH Public/Private Key	TLS Key Establishment

HMAC Key	Message Authentication
RBG CSPs	Random Bit Generation
ECDSA Public/Private Key	KeyGen for ECDH
ECDSA Public/Private Key	Digital Signatures
X.509 Certificates	Digital Signatures

8.2.11 [AGENT] FCS_STG_EXT.1(2):

All MDM Agent keys for iOS are stored in the Trust Anchor of the device.

The following table contains the list of keys and CSPs for the MDM Agent along with their purpose:

Key/CSP	Purpose
AES-CBC	Data at rest Data in transit
AES-GCM	Data in transit
ECDH Public/Private Key	TLS Key Establishment
HMAC Key	Message Authentication
RBG CSPs	Random Bit Generation
ECDSA Public/Private Key	KeyGen for ECDH
ECDSA Public/Private Key	Digital Signatures
X.509 Certificates	Digital Signatures

8.2.12 FCS_TLSC_EXT.1:

[SERVER] The MDM Server is able to act as a TLS client by relying on the TLS capabilities provided by the underlying platform. All of TLS 1.0, 1.1, and 1.2 are supported. When the MDM Server platform is acting as a TLS client, it will only allow the following ciphersuites to be used in the evaluated configuration:

- 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_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
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

When TLS 1.2 is used, the TOE platform verifies the reference identifier during the key exchange process. The reference identifier is the server URL and is not configurable. The TOE platform supports mutual TLS authentication using X.509v3 certificates. The TLS channel will only be established if the peer certificate has been validated. This validation includes checks for the validity of the Distinguished Name (DN), Subject Name (SN), and/or Subject Alternative Name (SAN) along with any extendedKeyUsage field information. The DN and any SAN information is checked against the identity of the remote computer's DNS entry. A certificate can be used for authentication if a wildcard is used in the leftmost portion of the resource identifier. Certificate pinning is not supported. Depending on the ciphersuite used to establish the trusted communications, parameters for any NIST curves include secp256r1 or secp384r1.

[AGENT] Similar to the MDM Server, the MDM Agent relies on the underlying mobile OS platforms to establish TLS communications as a client. Note that unlike with the MDM Server, the iOS MDM Agent only supports TLS 1.2. It also requires the use of either the TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 or the TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ciphersuite. The specific TLS implementation details for iOS can be found in the Apple Security Target documentation referenced in section 1.1.5 of this document.

8.2.13 [SERVER] FCS_TLSS_EXT.1:

The MDM Server is able to act as a TLS server by relying on the TLS capabilities provided by the underlying platform. This functionality is used to support secure web access to the Admin Console and Self-Service Portal, and connections from the client device. All of TLS 1.0, 1.1, and 1.2 are supported. When the MDM Server platform is acting as a TLS server, it will only allow the following ciphersuites to be used in the evaluated configuration:

- 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_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
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The specific OS component used to implement TLS server capabilities is Microsoft IIS 8.5. This can be configured to reject any connections using SSL versions 1.0, 2.0, or 3.0. The MDM Server platform supports mutual authentication of TLS clients using X.509v3 certificates and will not establish a connection if the certificate is invalid. Client certificates are generated by the environmental CA server and delivered to the client device upon enrollment. IIS verifies the distinguished name (DN) and Subject Alternative Name (SAN) and will not establish a connection if the validation fails. The TOE platform is also responsible for generating the key agreement parameters. Depending on the ciphersuite used to establish the trusted communications, parameters for any NIST curves include secp256r1 or secp384r1; or Diffie-Hellman parameters of either 2048 or 3072 bits.

8.3 Identification and Authentication

8.3.1 [SERVER] FIA_ENR_EXT.1

In the evaluated configuration of the TOE, a user enrolls their mobile device through a series of steps. First, the user powers on the mobile device and follows the standard iOS Setup Assistant instructions, including language, country/region, and Wi-Fi network. Once the device has been set up, it will communicate with Apple DEP in order to establish a connection to the MDM Server.

Once the TLS connection between the mobile device and Server is established, the user provides their credentials to authenticate to the MDM Server and then the enrollment process begins. There are two methods of configuring user authentication for device enrollment:

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

The TLS certificate that is used for authentication of this trusted channel is provided by an external Certificate Authority and the binding step is performed on the MDM Server platform through Microsoft Internet Information Services (IIS). This can be done before AirWatch is installed.

In the deployment of corporate owned devices, a whitelist of approved devices can be created. The MDM Server can limit the user's enrollment of specific devices based on device registration with Apple DEP (which is performed using device serial number). The DEP registration list effectively acts as a device whitelist. This is done by specifying Registered Devices Only in the registration settings; the TOE will acquire the list of registered devices through periodic synchronization with DEP.

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

8.3.2 [AGENT] FIA_ENR_EXT.2:

During the enrollment process, the MDM Agent records the MDM Server's DNS name and full URL with hostname. This is the only reference identifier used for the MDM Server. An MDM Agent can only be enrolled with one MDM Server at a time.

8.3.3 [SERVER] FIA_UAU.1:

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

In addition, there is an "about" button on the Admin Console homepage that includes the MDM Server software version number, copyright and licensing agreement information.

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

8.3.4 FIA_X509_EXT.1:

[SERVER] Certificate validation for the MDM Server is the responsibility of the underlying Windows Server 2012 platform. The certificate validation service will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE. Certificate status is validated using CRLs. The certificate validation service will also ensure that the extendedKeyUsage field is properly set for all certificates depending on their intended usage.

[AGENT] Certificate validation for the MDM Agent is the responsibility of the underlying iOS platform. The validity of a certificate will be checked in the same manner as for the MDM Server; the only difference is the specific component that is responsible for doing this check.

8.3.5 FIA_X509_EXT.2:

[SERVER] The MDM Server relies on the TOE platform to provide X.509v3 certificate services for code signing of system software updates, for integrity verifications, and for policy signing. X.509v3 certificates are also used for all TLS and HTTPS authentication. Certificates used for integrity verification and TOE updates are signed using a public CA certificate during the software build. Third-party .dll files used by the TOE are signed using a self-signed certificate during code build and maintained by AirWatch. The TLS connection between managed devices and the MDM Server is provided by the customer binding a certificate to port 443 on the Device Services server via Windows IIS.

If the TOE platform cannot establish a connection to determine the validity of a certificate, then the certificate is accepted. The MDM Server platform requires each MDM Agent device to have a unique certificate. This is done by invoking the environmental CA Server via DCOM and having it generate a unique client certificate during the process of device enrollment. The certificate is transmitted to the managed device and installed, after which all subsequent communications between the MDM Server and the managed device require certificate-based authentication.

[AGENT] The managed device uses certificates for HTTPS authentication and for verification of policy updates. This functionality is provided by the MDM Agent or by the underlying mobile OS, depending on the function.

On iOS devices, the certificate verification is performed by the MDM Agent and iOS native MDM policies will be validated by the OS. Regardless of the type of profile that is sent to an iOS device, the underlying platform is invoked to validate the certificate chain. The TLS trusted channel is verified by the underlying OS platform regardless of the type of device. A unique certificate is issued by the external CA. If the TOE platform cannot establish a connection to determine the validity of a certificate, then the certificate is accepted.

8.4 Security Management

8.4.1 [SERVER] FMT_MOF.1(1):

The MDM Server provides the capability to manage its own functionality as well as the behavior of mobile devices that are under management. For its own functionality, the Admin Console provides the ability to configure the certificates that are used by the MDM Server as well as the devices that are permitted to enroll in management. This option also allows a maximum number of devices per user to be configured as well as a valid time period for enrollment. The Admin Console also allows Administrators to specify the unlock banner for both the Admin Console and Self-Service Portal and the time period for the MDM Server to do its periodic checks of the managed devices to check its characteristics, installed applications, and audit logs. Finally, the Admin Console provides the ability to configure how audit data is stored.

For the configuration of devices under management, the full list of functions that the TSF can perform and a reference to where in the Admin Console this behavior can be found is described in section 8.4.7

below. Note that iOS does not provide the ability for third-party MDM software to configure certain aspects of the device's functionality.

As discussed in section 8.4.11, the TOE provides the ability to define multiple administrative roles, each with its own set of authorized permissions. Individual accounts can be assigned these administrative roles and can also be scoped to only have the authority to manage certain devices or collections of devices based on group membership. If an administrator belongs to a role that has the privilege to perform a certain action and the target for that action is within the scope of their group membership, they are considered to be an Authorized Administrator for the requested function for the purposes of this SFR.

8.4.2 [SERVER] FMT_MOF.1(2):

Enrollment of mobile devices is brokered using Apple DEP. In its evaluated configuration, the MDM Server is configured to specify the use of registered devices only. Administrators ensure that mobile devices are first registered with DEP so that they can be selected for enrollment. Enrollment is then performed by a user or administrator with physical custody of the mobile device. Note that in order to enroll the device, valid user credentials are required. Since administrators are responsible for the creation of user accounts, they are able to perform first-use actions requiring user authentication prior to the user accessing the account and changing their password.

8.4.3 [SERVER] FMT_MOF.1(3):

The MDM Server uses "smart groups" to separate devices based on how policies are applied to them. Smart groups can consist of organization groups, user groups, and device characteristics. Administrators have the ability to construct smart groups from these other groups. For any applications that reside in the MAS Server or public applications that are referenced through external links, the Administrator has the ability to assign one or more smart groups to the app to push it to a set of devices or make it available to be downloaded by them. This assignment can be used to determine if the app is automatically pushed to certain devices based on smart group membership or if it is available on demand. Applications can also be assigned to application groups. These groups are used to collect applications into a bundle of required apps, blacklisted apps, or whitelisted apps. The application group is then associated with one or more user and/or organizational groups in order to specify what devices it applies to. iOS does not provide the ability to automatically push apps onto a device or to block unapproved apps so all enforcement of required/whitelisted/blacklisted apps is handled in a reactive manner. If an iOS application is specified as Auto, the device user will be prompted to accept the application before it is downloaded and installed.

8.4.4 [SERVER] FMT_MOF.1(4):

The MDM Server provides two methods of restricting the download of applications. As stated in FMT_MOF.1(3) above, individual applications can be assigned to smart groups. If a device belongs to the smart group, whether it was individually assigned, has characteristics (such as operating system version or model type) associated with the smart group, or is owned by a user who belongs to the group, an application may be configured to be made available to download on demand from the MAS Server.

The TOE also provides the ability to restrict the download of applications through the use of whitelisting and blacklisting. As stated in section 8.4.3 above, applications can be assigned to application groups which identify group members as required apps, blacklisted apps, or whitelisted apps. These can be thought of as application access groups. Application groups can then be assigned to smart groups which determines what devices they are applied to. iOS devices cannot proactively enforce mandatory installation of required apps or mandatory prohibition of blacklisted or non-whitelisted apps. However, this categorization of apps and subsequent association with managed devices allows the MDM Agent TOE to alert the MDM Server if a managed device has an application loadout that is in violation of policy.

8.4.5 [SERVER] FMT_POL_EXT.1:

The MDM Server provides policies (known as “profiles”) that can be assigned to groups of devices. Profiles are transmitted to these assigned devices and applied to them using some combination of the device’s MDM Agent and its underlying platform, depending on the settings that are configured and the device’s platform OS.

For iOS devices, all profiles sent to the iOS native MDM Agent are signed with a trusted root certificate using ECDSA with SHA-256 or SHA-512. Since iOS has a finite set of trusted root certificates it permits, the customer is expected to acquire certificates from a trusted third party (e.g. GoDaddy, VeriSign). Profiles that are consumed by the TOE MDM Agent are signed in the same manner but ECDSA with SHA-512 is always used.

8.4.6 [AGENT] FMT_POL_EXT.2:

Candidate profiles are received by MDM Agents from the MDM Server when they are created. These profiles are signed using ECDSA with SHA-256 or SHA-512. The only time ECDSA with SHA-256 is used is when profiles are sent to the iOS native MDM agent. The MDM Agent verifies the digital signature and rejects the policy if the signature or certificate is invalid. If the certificate is validated, the profile is applied as defined. The MDM Server certificate that is used for validation is installed on the MDM Agent as part of the enrollment process.

8.4.7 [SERVER] FMT_SMF.1(1):

The MDM Server component of the TOE has the ability to issue commands and configuration policies to mobile devices. Depending on the mobile device platform and the function being configured, these may be transmitted to the device itself through its native MDM capabilities or to the VMware AirWatch MDM Agent component of the TOE that resides on the device. This is dependent on what APIs the device operating system makes available to third-party applications to access remotely.

The following table lists the management functions that can be performed by the MDM Server as defined by the MDM PP, how those functions are initiated, as well as whether this behavior is enforced by the VMware AirWatch MDM Agent or by the underlying iOS platform. Unless specified otherwise, the management function is initiated from the device Details View in the Admin Console.

Command	Implemented By
1. transition to the locked state – “Lock” button.	Platform
2. full wipe of protected data – “More Actions” button > Device Wipe.	VMware AirWatch MDM Agent
3. unenroll from management – “More Actions” button > Device Wipe.	VMware AirWatch MDM Agent

4. install policies – assigned and applied to target devices at the creation or modification of a profile under Devices > Profiles.	Platform
5. query connectivity status – “Query” button.	VMware AirWatch MDM Agent (initiator) Platform (response)
6. query the current version of the MD firmware/software – “Query” button. Status shown in the main detail view page.	VMware AirWatch MDM Agent (initiator) Platform (response)
7. query the current version of the hardware model of the device – “Query” button. Status shown in the main detail view page.	VMware AirWatch MDM Agent (initiator) Platform (response)
8. query the current version of installed mobile applications – “Query” button. Status shown in the Apps tab under the main detail view page.	VMware AirWatch MDM Agent (initiator) Platform (response)
9. import X.509v3 certificates into the Trust Anchor Database – assigned and applied to devices as part of a policy under the “Credentials” tab when defining the policy.	Platform
10. install applications – Apps and Books tab, Details View. Admin will be prompted to define what devices an application is assigned to during definition or modification of the application. When the application is specified as automatic distribution, the installation is initiated by the TSF.	Platform
11. update system software – this is the responsibility of the carrier and not the TOE.	N/A
12. remove applications – iOS does not currently provide an interface for MDM applications to initiate the removal of non-Enterprise applications.	N/A
13. remove Enterprise applications – can be removed in several ways: - specific application from a single device: Device details, Apps tab, Remove option (“X”) button for the desired application. - specific application from all devices: Apps and Books, App details, “Remove From All” button.	Platform
14. wipe Enterprise data – “More Actions” button > Device Wipe.	VMware AirWatch MDM Agent
15. remove imported X509v3 certificates – “More” tab > “Certificates”, Revoke option.	Platform
16. alert the administrator – “Send” button. Note that this refers to alerting the administrator of the mobile device, not the Administrator for the MDM Server. This can be sent as an email, SMS, or push notification.	Platform

22. place applications into application process groups – Apps & Books > Applications Settings > App Groups.	VMware AirWatch MDM Server

Configuration Policy	iOS Implementation
24. password policy – defined in the Passcode properties of a profile.	Platform
25. session locking policy – Defined in the Passcode properties of a profile.	Platform
26. wireless networks (SSIDs) to which the MD may connect – Defined under the Wi-Fi properties of a profile. Note that a profile specifies only a single permitted SSID so if multiple SSIDs are permitted, multiple profiles must be assigned to the device.	Platform
27. security policy for each wireless network – defined in the Wi-Fi properties of a profile, except for the permitted CA(s) which is specified under Credentials.	Platform
28. application installation policy – groups of required, whitelisted, and/or blacklisted apps can be defined in Apps & Books > App Groups. Note that iOS does not provide a mechanism to pre-emptively enforce application whitelisting/blacklisting but the TOE can take corrective action if a compliance policy is defined to detect the presence of a blacklisted or non-whitelisted app.	Platform
29. enable/disable policy for camera and microphone across MD – defined in the Restrictions properties of a profile.	Platform
30. enable/disable policy for the VPN across the mobile device and on a per-app basis – defined in the VPN properties of a profile or in the “VPN Access” setting for an individual app assignment.	Platform
35. enable policy for data-at-rest protection – For Apple devices, data-at-rest protection is automatically enabled if a passcode is set so this is configured under the Passcode properties of a profile.	Platform
49. enable/disable backup – Defined in the Restrictions properties of a profile under the iCloud subcategory.	Platform
53b. enable/disable authentication mechanisms providing user access to protected data other than a Password Authentication Factor (e.g. using a fingerprint) – Defined in the Restrictions	Platform

properties of a profile under the Device Functionality subcategory.	
53c. policies for which there are required configuration values in the mobile operating system STIG relevant to the MD – The act of defining profiles in general allows relevant STIG configuration values to be applied.	Platform
53d. full wipe of all user data and applications not included in the out-of-the-box install – Accomplished through Factory Reset of the device from the Device Details view.	VMware AirWatch MDM Agent

8.4.8 [SERVER] FMT_SMF.1(2):

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

- **Configuration of X.509v3 certificates for MDM Server use:** Certificates used for the MDM server are distributed to devices via profiles. This is specified in the Credentials section of a profile.
- **Configure devices permitted for enrollment based on specific devices:** Defined in Devices & Users > General > Enrollment where Current Setting is set to Override and Devices Enrollment Mode is set to Registered Devices Only
- **Configuration of TOE unlock banner:** Defined for both the Admin Console and Self-Service Portal in Settings under System > Branding.
- **Periodicity of agent communications:** Set globally for the supported device platforms as follows: Groups and Settings > All Settings > Devices and Users > Apple > MDM Sample Schedule.
- **Storage of device audit log data:** Stored in the environmental SQL database and displayed automatically in the Admin Console under Hub > Events > Device Events.
- **Remote transmission of stored audit log data:** Use of Syslog configured in Settings under System > Enterprise Integration > Syslog.

8.4.9 [SERVER] FMT_SMF.1(3):

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

The MAS Server also provides the ability to configure application access groups so that different applications can be flagged as required, whitelisted, or blacklisted. These application groups are assigned a type (requires, whitelisted, blacklisted) and associated with a smart group in the same way that individual applications are assigned. The MDM Server can then define compliance policies to generate alerts when required groups are missing or when prohibited/non-whitelisted apps are present.

8.4.10 [AGENT] FMT_SMF_EXT.3:

The MDM Agent has the ability to interact with the underlying mobile device platform in order to enforce the MDM Server management functions. All commands and configuration policies that are defined in FMT_SMF.1(1) that are received by the MDM Agent result in the mobile device platform being queried or modified in some way. This also includes the ability to upload certificates into the device's certificate store that are used to establish trusted communications between the MDM Agent and the MDM Server as part of the initial installation/configuration of the MDM Agent. Information about how the MDM Agent may enforce these management functions depending on the mobile device platform is provided in section 8.4.5 above.

Through Administrator-initiated registration via DEP, the MDM Server has the ability to select a mobile device that has been registered through DEP for enrollment. The MDM Agent installs itself with administrative privileges on the underlying mobile device so that it has sufficient privileges to exercise any management functions that are directed by the MDM Server. Unenrollment protection is enforced through the Lock MDM Profile feature. This blocks all methods of unenrollment short of factory reset of the device itself.

The MDM Agent also has the ability to enforce the periodicity of reachability events by enforcing the sampling interval values that are configured on the MDM Server. When the MDM Server communicates with an MDM Agent for policy/sample data, this is considered to be a reachability event since the outcome of this activity updates the Last Seen time of the device in the Admin Console.

8.4.11 [SERVER] FMT_SMR.1(1):

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

- They are performing an action that is allowed based on the permissions granted to their assigned admin role.
- They are accessing an object that is within the scope of their admin group membership. If the Administrator has no assigned admin group, all objects are within their authorized scope.

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

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

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

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

Users (or "MD users") are defined separately from administrators. If someone is defined as a user in the Admin Console, they will not be defined as an administrator. A user is simply an individual who possesses a mobile device that is enrolled in management. A user is able to access the Self-Service Portal but not the Admin Console.

8.4.12 [SERVER] FMT_SMR.1(2):

The MAS Server is logically integrated with the MDM Server. It is accessed by Administrators using the Apps & Books tab in the Admin Console. Since this is not accessed separately from the remainder of the MDM Server capabilities, the administrative roles that can interact with the MAS Server are defined in the same manner as for FMT_SMR.1(1) above.

8.4.13 [AGENT] FMT_UNR_EXT.1:

Apple DEP provides the unenrollment protection mechanism for AirWatch through the use of the Lock MDM Profile feature. AirWatch leverages the functionality provided by the underlying platform to prevent the unauthorized removal of the MDM Agent software.

Unenrollment from mobile device management is prevented through the use of Apple DEP.

8.5 Protection of the TSF

8.5.1 [SERVER] FPT_ITT.1(1):

All communications between the MDM Server and MDM Agents are protected using TLS/HTTPS. This is true regardless of the underlying mobile device platform for the MDM Agent. Remote connectivity to the Admin Console and Self-Service Portal are also secured using TLS/HTTPS. This is sufficient to protect the data in transit from unauthorized modification and/or disclosure.

8.5.2 [SERVER] FPT_ITT.1(2):

In the evaluated configuration, the TOE is deployed in an on-premise multiple server mode such that one instance of the MDM Server is located in the network DMZ and a second instance is located behind the organization’s firewall. The external instance runs the Self-Service Portal while the internal instances runs the Admin Console. The Self-Service Portal relays device status information and audit logs to the Admin Console so that this information is kept current and audit data resides in a single location. There are no direct communications between these two components; instead, they share information through querying/modifying shared data sources using the trusted channels defined in FTP_ITC.1(1). The MAS Server component of the TOE is part of the same server application as the MDM Server so there is no remote communication channel between these two components. The only other internal TSF remote communications are between the MDM Server and the MDM Agents, which is discussed in FPT_ITT.1(1) above.

8.5.3 [SERVER] FPT_ITT.1(3):

The MAS Server is logically integrated with the MDM Server. Therefore, the TLS/HTTPS trusted channel used to secure communications between the MDM Server and the MDM Agents is also used for MAS Server communications.

8.5.4 [SERVER] FPT_TST_EXT.1:

The MDM Server .dll files and executable code are digitally signed using an X.509v3 certificate from a trusted third party. During initial installation of the MDM Server and each time the server application is started, the native Windows Authenticode process is invoked to validate the integrity of the MDM Server.

In addition, the MDM Server uses the FIPS 140-2 validated cryptographic modules that belong to the underlying server platform (CMVP certificates #2357 and #2356). These modules each perform their own power-up self-tests upon initial start-up, including cryptographic algorithm known answer tests (KATs) and an integrity verification check.

8.5.5 [SERVER] FPT_TUD_EXT.1:

Updates for the MDM Server are downloaded as a zip package from the AirWatch support website at www.AirWatch.com. The updates are digitally signed using a Verisign X.509v3 certificate which is installed in the Windows trusted key store on the underlying server platform which verifies the software updates. The updates are installed through the platform directly onto the MDM Server which does not have a method of pulling down or installing the updates itself. The before and after version numbers of the MDM Server software can be checked by clicking on the “About” button on the MDM Server Admin Console.

8.6 TOE Access

8.6.1 [SERVER] FTA_TAB.1:

The MDM Server supports the ability to apply custom branding to the Admin Console and the Self-Service Portal login pages. This includes the ability to display text of the Administrator’s choosing, which allows for a configurable warning banner to be displayed to both Administrators and users prior to authenticating to the MDM Server.

8.7 Trusted Path/Channels

8.7.1 [SERVER] FTP_ITC.1(1):

The trusted communication channels between the MDM Server and the syslog audit server, AD/LDAP authentication server, and the SQL database server are trusted communications channels. Listed below are each of the third-party external interfaces to the MDM Server along with the protocol that is used to secure communications over the interface:

- MDM Server to syslog audit server: TLS
- MDM Server to SQL database: TLS
- MDM Server to AD/LDAP authentication server: TLS

The use of TLS to establish these trusted channels ensures that data in transit will not be subjected to unauthorized modification or disclosure.

8.7.2 [SERVER] FTP_ITC.1(2):

The TSF uses trusted channels to secure communications between the MDM Server and the MDM Agent, regardless of whether the MDM Server communicates directly with the MDM Agent or if the MDM Agent's underlying platform is used as an intermediary. The direct channel between the MDM Server and the MDM Agent uses TLS over HTTPS. When the MDM Server communicates to a mobile device via a third party push notifications server, TLS is used to secure these communications. When the MDM Agent or the underlying device platform communicates back to the MDM Server, this channel is secured using TLS over HTTPS.

8.7.3 [SERVER] FTP_ITC.1(3):

The MAS Server is logically integrated with the MDM Server. As a result of this, all remote communications that the MAS Server requires to operate are identical to those described in FTP_ITC.1(1) above.

8.7.4 [SERVER] FTP_TRP.1(1):

The MDM Server platform uses TLS/HTTPS to provide a trusted communications between the MDM Server and Administrators attempting to connect to the Admin Console for the purposes of remote administration.

8.7.5 [SERVER] FTP_TRP.1(2):

The MDM Server platform uses TLS/HTTPS to provide a trusted communications between the MDM Server and users attempting to connect to the Self-Service Portal for the purposes of remote device registration and other self-service tasks.