Assurance Activities Report for a Target of Evaluation

VMware Workspace ONE Unified Endpoint Management Version 2209

Assurance Activities Report (AAR) Version 1.0

February 10, 2023

Security Target (Version 1.0)

Evaluated by:

Booz | Allen | Hamilton

Booz Allen Hamilton Common Criteria Test Laboratory
NIAP Lab # 200423
1100 West St.
Laurel, MD 20707

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304

The Author of the Security Target:

Booz Allen Hamilton
1100 West St.
Laurel, 20707 USA

The TOE Evaluation was sponsored by:

Booz Allen Hamilton

Evaluation Personnel:

Herb Markle
Christopher Rakaczky

Applicable Common Criteria Version

Common Criteria for Information Technology Security Evaluation, April 2017 Version 3.1 Revision 5

Common Evaluation Methodology Version

Common Criteria for Information Technology Security Evaluation, Evaluation Methodology, April 2017 Version 3.1 Revision 5

Table of Contents

1 Purpose

The purpose of this document is to serve as a non-proprietary attestation that this evaluation has satisfied all the TSS, AGD, and ATE Assurance Activities required by the Protection Profiles/Extended Packages to which the TOE claims exact conformance. This will give system integrators valuable information about product configuration and testing, help to align Common Criteria evaluations with DISA Security Requirements Guides and Security Test Implementation Guides (SRGs/STIGs), and thereby streamline the process for U.S. Government procurement of validated products.

2 TOE Summary Specification Assurance Activities

The evaluation team completed the testing of the Security Target (ST) *VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target v1.0* and confirmed that the TOE Summary Specification (TSS) contains all Assurance Activities as specified by the *Protection Profile for Mobile Device Management Version 4.0 [MDMPP]* and *PP-Module for MDM Agent Version 1.0 [AGENTMOD]*. The evaluators were able to individually examine each SFR’s TSS statements and determine that they comprised sufficient information to address each SFR claimed by the TOE as well as meet the expectations of the MDMPP and AGENTMOD Assurance Activities.

Through the evaluation of ASE_TSS.1-1, described in the ETR, the evaluators were able to determine that each SFR was described in enough detail to demonstrate that the TSF addresses the SFR. However, in some cases the Assurance Activities that are specified in the claimed source material instruct the evaluator to examine the TSS for a description of specific behavior to ensure that each SFR is described to an appropriate level of detail. The following is a list of each SFR, the TSS Assurance Activities specified for the SFR, and how the TSS meets the Assurance Activities. Additionally, each SFR is accompanied by the source material MDMPP and AGENTMOD that defines where the most up-to-date TSS Assurance Activity was defined.

[MDMPP] FAU_ALT_EXT.1.1 – *“The evaluator shall examine the TSS and verify that it describes how the alert system is implemented. The evaluator shall also verify that a description of each assigned event is provided in the TSS.”*

Section 8.1.1 of the ST describes how the alert system is implemented via configurable “compliance policies.” A description of each assigned event for this SFR is also provided in this section of the ST.

[AGENTMOD] FAU_ALT_EXT.2/ANDROID – *“The evaluator shall examine the TSS and verify that it describes how the alerts are implemented.*

The evaluator shall examine the TSS and verify that it describes how the candidate policy updates are obtained and the actions that take place for successful (policy update installed) and unsuccessful (policy update not installed) cases. The software components that are performing the processing must also be identified in the TSS and verified by the evaluator.

The evaluator also ensures that the TSS describes how reachability events are implemented, and if configurable are selected in FMT_SMF_EXT.4.2. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events.

The evaluator shall ensure that the TSS describes under what circumstances, if any, the alert may not be generated (e.g., the device is powered off or disconnected from the trusted channel), how alerts are queued, and the maximum amount of storage for queued messages.”

Section 8.1.2 of the ST describes how the alert system is implemented via configurable “compliance policies.”

Section 8.5.5 of the ST describes how candidate policy updates are obtained and the actions that take place for a successful or unsuccessful policy installation. The TSS also identifies the software performing the processing. As part of the test assurance activities for FMT_POL_EXT.2 and FIA_X509_EXT.1(1) and FIA_X509_EXT.2, the software components that perform the processing was verified.

Section 8.1.2 of the ST describes how the Android Hub Agent is configured by the UEM Server to generate periodic reachability events based upon a configured ‘sample interval’ and ‘transmit interval.’ The TSS states clearly that the Android Hub Agent is responsible for initiating the reachability events. “Configure periodicity of reachability events” is selected in FMT_SMF_EXT.4.2.

Section 8.1.2 of the ST states that “if the connection between the Android Hub Agent and UEM Server is down during a 'transmit interval', the Android Hub Agent continues to queue sample intervals of collected data until a connection is available for a 'transmit interval'. The maximum amount of storage is 10 sample intervals. The actual amount of storage for alerts depends on the amount of storage space of the device and the amount the device allocates to the Android Hub Agent app.”

[AGENTMOD] FAU_ALT_EXT.2/IOS – *“The evaluator shall examine the TSS and verify that it describes how the alerts are implemented.*

The evaluator shall examine the TSS and verify that it describes how the candidate policy updates are obtained and the actions that take place for successful (policy update installed) and unsuccessful (policy update not installed) cases. The software components that are performing the processing must also be identified in the TSS and verified by the evaluator.

The evaluator also ensures that the TSS describes how reachability events are implemented, and if configurable are selected in FMT_SMF_EXT.4.2. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events.

The evaluator shall ensure that the TSS describes under what circumstances, if any, the alert may not be generated (e.g., the device is powered off or disconnected from the trusted channel), how alerts are queued, and the maximum amount of storage for queued messages.”

Section 8.1.3 of the ST describes how the alert system is implemented via configurable “compliance policies.”

Section 8.5.5 of the ST describes how candidate policy updates are obtained and the actions that take place for a successful or unsuccessful policy installation. The TSS also identifies the software performing the processing. As part of the test assurance activities for FMT_POL_EXT.2 and FIA_X509_EXT.1(1) and FIA_X509_EXT.2, the software components that perform the processing was verified.

Section 8.1.3 of the ST describes how the iOS/iPadOS MDM protocol and iOS Hub Agent are configured by the UEM Server to respond to requests from the UEM Server querying the iOS platform based upon the iOS Hub agent’s periodic reachability event configuration. “Configure periodicity of reachability events” is selected in FMT_SMF_EXT.4.2.

Section 8.1.3 of the ST states that the alerts are generated by the underlying iOS/iPad OS platform. These alerts are generated as part of the request and response relationship of an active connection between the iOS/iPadOS platform and the UEM server; thus, there are no alerts to be queued when a connection is not available. This includes the iOS/iPadOS platform sending alerts when consuming policies (profiles) assigned to it as well as in response to the UEM Server querying the iOS/iPadOS platform based on the iOS Hub Agent’s periodic reachability event configuration.

[MDMPP] FAU_GEN.1.1(1) – *“The evaluator shall check the TSS and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described in the TSS. The evaluator shall verify that for every audit event described in the TSS, the description indicates where the audit event is generated (TSF, TOE platform).*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."

Section 8.1.4 of the ST lists (in Table 16) all of the auditable events, including every audit event type mandated by the PP. It also describes for each audit event described, the component generating the record, and a note describing how it is invoked, if invoked by the platform.

[MDMPP] FAU_GEN.1.2(1) – *"The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field."*

Section 8.1.4 of the ST provides a format for audit records, a brief description of each field, and an example of an audit record. The example also depicts the required information needed in the audit record: date and time of the event (underlined text), event type (bold text), subject identity (italicized text), success or failure of the event (bold italicized text), and where the event occurred (bold underlined text). This section also refers to the AGD for more audit record examples.

[MDMPP] FAU_GEN.1.1(2) – *"The evaluator shall check the TSS and ensure that it provides a format for audit records."*

Section 8.1.5 of the ST provides a format for audit records, a brief description of each field, and an example of an audit record. The example also depicts the required information needed in the audit record: date and time of the event (underlined text), event type (bold text), and mobile device identity (italicized text). This section also refers to the AGD for more audit record examples.

[MDMPP] FAU_GEN.1.2(2) – *"The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field."*

Section 8.1.5 of the ST provides a format for audit records, a brief description of each field, and an example of an audit record. The example also depicts the required information needed in the audit record: date and time of the event (underlined text), event type (bold text), and mobile device identity (italicized text). This section also refers to the AGD for more audit record examples.

[AGENTMOD] FAU_GEN.1(2) – *"The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Agent; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."

Section 8.1.6 of the ST provides a format for audit records, a brief description of each field, and an example of an audit record. The example also depicts the required information needed in the audit record: date and time of the event (underlined text), event type (bold text), subject identity (italicized text), and success or failure of the event (bold italicized text). This section also refers to the AGD for more audit record examples.

Section 8.1.6 of the ST lists (in Table 17) all of the auditable events, including every audit event type mandated by the PP. It also describes for each audit event described, the component generating the record, and a note describing how it is invoked, if invoked by the platform.

[MDMPP] FAU_NET_EXT.1.1 – *"The evaluator ensures that the TSS describes how reachability events are implemented, for each supported mobile platform. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events."*

Section 8.1.7 of the ST states that network reachability events are implemented by the UEM Server recording each time a device connects to the server. Devices will connect to the UEM Server based upon the iOS and Android Hub Agents' periodic reachability event configuration or in response to an on-demand request by an Authorized Administrator. For iOS, the TSF uses the Apple Push Notification Service (APNS) to send the request to the device and the device will respond to the request when it has network connectivity. For Android, the TSF uses Firebase Cloud Messaging Services (FCM) to send the request to the device and the device will respond to the request when it has network connectivity.

Android device periodic reachability events are initiated by the Android Hub Agents and iOS device periodic reachability events are initiated by the iOS/iPadOS platform. It is also possible for an Authorized Administrator to query device connectivity status for iOS and Android devices by initiating the request from the UEM Server (MDM Server).

[MDMPP] FAU_SAR.1.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.1.8 of the ST states that for cryptographic behavior that is performed by the UEM Server's underlying platform, auditing is stored in the Windows event logs. These records can be sorted, filtered, and searched via the underlying platform.

Table 16 in the ST describes which events are logged by the UEM Server versus its 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.

This adequately describes how the FAU_SAR.1.1 functionality is invoked.

[MDMPP] FAU_SAR.1.2 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.1.8 of the ST states that for cryptographic behavior that is performed by the UEM Server's underlying platform, auditing is stored in the Windows event logs. These records can be sorted, filtered, and searched via the underlying platform.

Table 16 in the ST describes which events are logged by the UEM Server versus its 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.

This adequately describes how the FAU_SAR.1.2 functionality is invoked.

[AGENTMOD] FAU_SEL.1(2) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS of the ST to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Agent; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.1.9 of the ST states the Authorized Administrator is responsible for creating policies that require auditing. Once the iOS Hub Agent, iOS Platform, and/or Android Hub Agent receive and apply a policy requiring auditing, they will always generate the necessary audit records.

[MDMPP] FAU_STG_EXT.1.1 – *"The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided."*

Section 8.1.10 of the ST states that audit data managed by the UEM Server will be transmitted from the UEM Server to a remote Syslog Server over TLS v1.2 encrypted trusted channels. The actual TLS encryption is handled by the underlying Windows Server 2019 platform.

The audit data that are transferred include audit records generated by the UEM Server software as well as audit records that are received from iOS and Android Hub Agents. This does not include audit data that are generated by these TOE components' underlying platforms as this audit data are not managed by the TOE's software boundary. It is therefore the responsibility of the Operational Environment to securely transfer this audit data to a remote location for permanent storage.

The UEM Server is configured to send TOE managed audit data over a specific port to the Syslog Server and once a connection is successfully established, the TOE managed audit logs are sent to the Syslog Server in real time upon the UEM Server creating them or receiving them from Hub Agents.

[MDMPP] FCO_CPC_EXT.1.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

The SFR selection is "implement functionality", therefore, this TSS Assurance Activity is not applicable.

[MDMPP] FCO_CPC_EXT.1.2 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

The "invoke platform-provided functionality" is selected as part of FCO_CPC_EXT.1.2; however, the audit record associated with FCO_CPC_EXT.1 is a management action that is performed on the UEM Server which whitelists the mobile devices which can enroll into mobile device management. The selection for FCO_CPC_EXT.1.1 is "implement functionality" which aligns with the audit record purpose. Section 8.2.1 of the ST states that configuration of this enrollment restriction is the enablement step by the Authorized Administrator through the Admin Console. Additionally, "Table 16: Auditable Events by Enforcing Component" in Section 8.1.4 of the ST confirms that the component which performs this audit record is the UEM Server. Therefore, the evaluation team determined that "invoke platform-provided functionality" was properly selected due to FCO_CPC_EXT.1.2 discussing the handling of secure communications between a Hub Agent and the UEM Server during enrollment which is handled by the TOE components' platforms. However, the audit record associated with this SFR is implemented by the TOE's UEM Server component and would be considered "implement functionality", which is selected under FCO_CPC_EXT.1.1. Since FCO_CPC_EXT.1.1 has selected "implement functionality", the UEM Server is invoking its own audit functionality to support the audit record related to this SFR.

[MDMPP] FCO_CPC_EXT.1.3 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

The SFR selection is "implement functionality", therefore, this TSS Assurance Activity is not applicable.

[MDMPP] FCS_CKM.1.1 – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key generation functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.1 of the ST states that the UEM Server invokes the Windows Server 2019 platform provided functionality for asymmetric key generation in support of TLS communications. The Windows Server 2019 platform provides functionality to support RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and ECC schemes using "NIST curves" P-256 and P-384 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

For iOS and Android Hub Agents, the underlying OS platform is involved in support of TLS communications. Both iOS and Android platforms support RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and ECC schemes using "NIST curves" P-256 and P-384 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

This adequately describes how the FCS_CKM.1.1 functionality is invoked.

[MDMPP] FCS_CKM.2.1 – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key establishment functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.2 of the ST states that the UEM Server invokes the underlying Windows server platform in support of two key establishment schemes for the establishment of TLS communications:

- RSA key establishment conforming to "RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017" and
- Elliptic curve-based key establishment conforming to NIST Special Publication 800-56A.

For the iOS and Android Hub Agents, the software relies on the underlying mobile device platform to perform key establishment for TLS communications using the following two key establishment schemes:

- RSA key establishment conforming to "RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017" and
- Elliptic curve-based key establishment conforming to NIST Special Publication 800-56A

This adequately describes how the FCS_CKM.2.1 functionality is invoked.

[MDMPP] FCS_CKM_EXT.4.1 – *"If "invoking platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key destruction functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.3 of the ST states that the UEM Server invokes the underlying platform's FIPS cryptographic module to zeroize keys and cryptographic security parameter data when no longer needed. The invoking of key destruction occurs as a result of the UEM Server making a cryptographic function call which requires a key and/or cryptographic security parameter data to be generated by the platform. The platform will therefore perform key destruction when the generated cryptographic data is no longer needed; without requiring a separate function call for key destruction from the UEM Server. All key data maintained by the server platform exists only in volatile memory and are erased by a one-pass overwrite with zeroes followed by a read-verify.

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

This adequately describes how the FCS_CKM_EXT.4.1 functionality is invoked.

[MDMPP] FCS_CKM_EXT.4.2 – *“The evaluator shall check to ensure the TSS lists each type of plaintext key material and CSP (authentication data, authorization data, secret/private symmetric keys, data used to derive keys, etc.) and its origin and storage location.*

The evaluator shall verify that the TSS describes when each type of key material and CSP is no longer needed.

If "invoke platform-provided functionality" is selected:

The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key releasing functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”

If "implement functionality" is selected:

The evaluator shall also verify that, for each type, the type of clearing procedure that is performed is listed. If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are cleared by overwriting once with zeros, while secret keys stored on the internal persistent storage device are cleared by overwriting one time with a random pattern that is changed before each write"). For block erases, the evaluator shall also ensure that the block erase command used is listed and shall verify that the command used also addresses any copies of the plaintext key material that may be created in order to optimize the use of flash memory.

The selection “invoke platform-provided functionality” has been included for this SFR. The selection “implement functionality” has not been included for this SFR; therefore, this portion of the TSS assurance activity does not apply.

Sections 8.3.9 and 8.3.10 of the ST in tables 18 and 19 list the type of key material/CSP, their purpose, and origin for the UEM Server and the iOS and Android Hub Agents, respectively. The storage location (volatile memory for the UEM Server platform, and iOS and Android Hub Agents’ platforms; non-volatile memory in wear-leveled flash memory for iOS and Android Hub Agents’ platforms) is described in section 8.3.3 of the ST.

Section 8.3.3 of the ST states that keys are destructed when the cryptographic data are no longer needed, without requiring a separate function call to the underlying platform for key destruction from the UEM Server. Key destruction for the iOS and Android Hub Agents is performed by the underlying mobile device platform when the generated cryptographic data are no longer needed, without requiring a separate function call for key destruction from the iOS or Android Hub Agent. Key data maintained by the iOS and Android Hub Agents’ platform in volatile memory are erased by a one-pass overwrite with zeroes. Key data maintained by the iOS and Android Hub Agents’ platforms in non-volatile memory are stored in wear-leveled flash memory and are erased by a one-pass overwrite with ones (i.e. block erase).

[MDMPP] FCS_COP.1.1(1) – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the encryption/decryption functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.4 of the ST states that the UEM Server invokes the underlying Windows Server 2019 platform to perform AES encryption/decryption services for TLS communications and protection of data at rest in platform key storage. All data at rest are protected using AES-GCM-256 as defined in NIST SP 800-38D. Data in transit are protected using GCM mode and either 128-bit or 256-bit keys. The UEM Server's platform conforms to NIST SP 800-38D.

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

This adequately describes how the FCS_COP.1.1(1) functionality is invoked.

[MDMPP] FCS_COP.1(2) – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the hash functionality is invoked for each digest size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).*

The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS."

Section 8.3.5 of the ST states that the UEM Server invokes the Window server platform to provide SHA-256, SHA-384 and SHA-512 cryptographic hashing services, conformant to FIPS PUB 180-4, with digest sizes of 256, 384 and 512 bits, respectively. SHA-256 and SHA-384 are used by HMAC in message authentication for TLS, in support of ECDSA with P-256 and P-384 curves for signature services in TLS, and SHA-512 is used in the hashing of passwords and for the digital signing of policies (ECDSA with P-521 curve).

Section 8.3.5 of the ST also states that the iOS and Android Hub Agent's software invokes the underlying mobile device platform to perform cryptographic hashing in support of TLS communication. The iOS and Android Hub Agents' platforms use SHA-256 and SHA-384, conformant to FIPS PUB 180-4, in support of TLS communication.

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

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

[MDMPP] FCS_COP.1(3) – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the digital signature functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.6 of the ST states that the UEM Server invokes the Windows Server 2019 platform to provide all digital signature services in accordance with FIPS PUB 186-4. RSA with 2048-bit keys and ECDSA with P-256 and P-384 NIST curves are used for digital signature services in support of TLS communication. Additionally, ECDSA with P-521 NIST curve (using SHA-512) is used for policy signature generation.

Section 8.3.6 of the ST also states that the iOS and Android Hub Agents' software invokes the underlying mobile device platform to provide digital signature services in accordance with FIPS PUB 186-4. RSA with 2048-bit keys and ECDSA with P-256 and P-384 NIST curves are used for digital signature services in support of TLS communication.

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

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

[MDMPP] FCS_COP.1(4) – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the keyed-hash functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.7 of the ST states that the UEM Server invokes the Windows Server 2019 platform to provide keyed-hash message authentication services conformant to FIPS PUBs 180-4 and 198-1. HMAC-SHA-256 and HMAC-SHA-384 are used to perform keyed-hash message authentication, with respective digest sizes of 256 and 384 bits. The key used by HMAC is the UUID which is 160 bits. This key is stored on the Server platform.

Section 8.3.7 of the ST also states the iOS and Android Hub Agents' software invokes the underlying mobile device platform to provide keyed-hash message authentication services conformant to FIPS PUBs 180-4 and 198-1. HMAC-SHA-256 and HMAC-SHA-384 are used to perform keyed-hash message authentication, with respective digest sizes of 256 and 384 bits in support of trusted communication. The key used by HMAC is the UUID which is 160 bits.

[MDMPP] FCS_RBG_EXT.1.1 – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the RBG functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.3.8 of the ST states that the UEM Server invokes the underlying Windows Server 2019 platform to provide random bit generation services. The platform cryptographic module provides an AES counter DRBG (CTR_DRBG) that conforms to NIST SP 800-90A. In order to provide sufficient randomness to the keys generated by this DRBG (as specified in NIST SP 800-57), the DRBG is seeded with at least 256 bits of entropy which is gathered from a platform based RBG.

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

Section 8.3.8 also states: "Note: The TOE UEM Server and Hub Agent software do not directly invoke their respective platforms' deterministic random bit generator. Instead, the TOE's software indirectly invokes their platforms' deterministic random bit generator by directly invoking platform components, which in turn directly invoke the deterministic random bit generator."

This adequately describes how the FCS_RBG_EXT.1.1 functionality is invoked.

[MDMPP] FCS_RBG_EXT.1.2 – *"Documentation shall be produced-and the evaluator shall perform the activities-in accordance with Appendix D: Entropy Documentation and Assessment and the "Clarification to the Entropy Documentation and Assessment Annex."*

In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates."

A proprietary EAR was produced and approved for this evaluation.

[MDMPP] FCS_STG_EXT.1.1 – *"Regardless of whether this requirement is met by the TSF or the TOE platform, the evaluator will check the TSS to ensure that it lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored. The evaluator then performs the following actions.*

Persistent secrets and private keys manipulated by the TOE platform:

The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key storage functionality is invoked for each persistent secret and private key described in the TSS (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

Persistent secrets and private keys manipulated by the TSF:

The evaluator reviews the TSS to determine that it makes a case that, for each item listed as being manipulated by the TOE, it is not written unencrypted to persistent memory, and that the item is stored by the platform."

Section 8.3.9 of the ST lists each persistent secret in relation to the UEM Server in Table 18. For each key in the table, the purpose, the origin, and the mechanism used to invoke is described. All private keys are stored in the Windows Trust Store and all user credentials are stored in authentication repositories. The SQL database Master Key is stored encrypted using an AES-GCM 256-bit key encryption key (KEK). This KEK is encrypted and stored in the Windows Registry. The policy signing X.509v3 certificate is uploaded by the Authorized Administrator and is stored in the SQL database.

There are no persistent secrets and private keys manipulated by the TSF as all persistent secrets and private keys are handled by the platform.

[AGENTMOD] FCS_STG_EXT.1(2) – *"The evaluator will verify that the TSS lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and, for each platform listed as supported in the ST, how it is stored. The evaluator shall verify that the Agent calls a platform-provided API to store persistent secrets and private keys."*

Section 8.3.10 of the ST lists each persistent secret in relation to the iOS and Android Hub Agents in Table 19. For each key, the purpose is described, the origin, and the API used to invoke is described. The API for iOS/iPadOS devices is CoreCrypto in all instances and the APIs for Android are BoringSSL and/or SCrypto depending on function called. These are platform provided APIs as assessed under FPT_API_EXT.1. All iOS Hub Agent keys are stored in the Trust Anchor and all the Android Hub Agent keys are stored in the Android Trust Store of the device.

[MDMPP] FIA_ENR_EXT.1.1/ANDROID – *"The evaluator shall examine the TSS and verify that it describes the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall include the trusted path used for enrollment (FTP_TRP.1(2)), the method of user authentication (username/password, token, etc.), the method of authentication decision (local or remote authentication services), and the actions performed on the MDM Server upon successful authentication."*

Section 8.4.1 of the ST states that for Android, enrollment is performed by powering on the mobile device and following the standard Android Setup Assistant instructions, including language, country/region, and Wi-Fi network as well as downloading the Android Hub Agent from the Google Play Store. The device user will then enter the UEM Server's URL into the Android Hub Agent which will be used to establish the enrollment connection that is described under FTP_TRP.1(2). Once this connection is established, the user provides his or her credentials (username/password) to authenticate to the UEM Server and then the enrollment process begins. There are two methods for the authentication decision: Basic (local account defined on the UEM Server) and LDAP (account defined on a third-party identity store). Upon successful authentication, the UEM Server will then initiate the process of having a unique X.509v3 certificate being issued to the Android Hub Agent per the process described under FIA_X509_EXT.5 and will send the MDM profiles (policies) assigned to the device.

[MDMPP] FIA_ENR_EXT.1.2/ANDROID – *"The evaluator shall examine the TSS and verify that it implements a policy to limit the user's enrollment of devices."*

Section 8.4.1 of the ST describes ways the UEM Server can limit the user's enrollment of Android devices based on the device's IMEI and/or serial number, the specific model and/or manufacturer of the device, the number of devices enrolled by a user, and the installed operating system version.

[MDMPP] FIA_ENR_EXT.1.1/IOS – *"The evaluator shall examine the TSS and verify that it describes the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall include the trusted path used for enrollment (FTP_TRP.1(2)), the method of user authentication (username/password, token, etc.), the method of authentication decision (local or remote authentication services), and the actions performed on the MDM Server upon successful authentication."*

Section 8.4.2 of the ST states that for iOS devices, enrollment is performed by powering on the mobile device, and following the standard iOS Setup Assistant instructions, including language, country/region, and Wi-Fi network. The iOS Setup Assistant will continue the enrollment process to the UEM Server through Apple DEP.

As part of enrolling in Apple DEP, the iOS/iPadOS platform will receive the UEM Server's URL which will be used to establish the enrollment connection that is described under FTP_TRP.1(2). Once the HTTPS/TLS connection between an iOS/iPadOS platform and UEM Server is established, the user provides their credentials to authenticate to the UEM Server. There are two methods of configuring user authentication for device enrollment: Basic (local account defined on the UEM Server) and LDAP (account defined on a third-party identity store). Upon successful authentication, the iOS Hub Agent is then deployed as a managed app by the UEM Server to the mobile device. The UEM Server will then initiate the process of having a unique X.509v3 certificate being issued to the iOS Hub Agent per the process described under FIA_X509_EXT.5 and will send the MDM profiles (policies) assigned to the device.

[MDMPP] FIA_ENR_EXT.1.2/IOS – *"The evaluator shall examine the TSS and verify that it implements a policy to limit the user's enrollment of devices."*

Section 8.4.2 of the ST describes ways the UEM Server can limit the user's enrollment of iOS devices based on the device registration with Apple DEP. The DEP registration list of devices' DEP identifiers effectively acts as a device allow list. This is done by an Authorized Administrator specifying Registered Devices Only in the registration settings; the UEM Server will acquire the list of registered devices through periodic synchronization with Apple DEP.

[AGENTMOD] FIA_ENR_EXT.2 – *"The evaluator shall examine the TSS to verify that it describes which types of reference identifiers are acceptable and how the identifier is specified (e.g. preconfigured in the MDM Agent, by the user, by the MDM server, in a policy)."*

Section 8.4.3 of the ST states that the iOS and Android Hub Agents record the UEM Server's DNS name and full URL with hostname. This is the only reference identifier used for the UEM Server. Section 8.4.1 of the ST states that the device user manually enters the UEM Server's URL for Android Hub Agents. Section 8.4.2 of the ST states that for iOS Hub Agents will receive the UEM Server's URL as part of enrolling in Apple DEP.

[MDMPP] FIA_UAU.1.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

The SFR selection is "implement functionality", therefore, this TSS Assurance Activity is not applicable.

[MDMPP] FIA_UAU.1.2 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

The SFR selection is "implement functionality", therefore, this TSS Assurance Activity is not applicable.

[MDMPP] FIA_X509_EXT.1.1(1) – TD0641 – *"If invoke platform-provided functionality is selected:*

The evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity.)

The TSS must describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of an X.509 certificate only when it is loaded onto the device.

If implement functionality is selected:

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

The TSS must describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of an X.509 certificate only when it is loaded onto the device.

Section 8.4.5 of the ST states that the UEM Server relies on the underlying platform to provide X.509v3 certificate services for verification of the code signing of UEM Server software updates, for integrity verification of the UEM Server software, and signing profiles (policies) that are sent to iOS and Android Hub Agents.

Section 8.4.5 of the ST also states that the iOS and Android Hub Agents rely on the underlying mobile platforms' cryptographic modules to provide X.509v3 certificate services for verification of the code signing of Hub Agent software updates and in support of HTTPS/TLS connections to the Hub Agents.

Additionally, for the iOS underlying platform only, the platform's OpenSSL implements X.509v3 certificate services for the verification of signed profiles (policies) received from the UEM Server that will be applied by the iOS/iPadOS underlying platform.

The Android Hub Agent's instance of OpenSSL implements X.509v3 certificate services for the verification of signed profiles (policies) received from the UEM Server.

A description of the certificate path validation algorithm is described, outlining the checks performed in order to determine if a certificate is valid.

Finally, the TSS describes when revocation checking is performed which is the same in all implemented instances: "Revocation checking occurs each time a certificate is presented for a validation check."

This adequately describes how the FIA_X509_EXT.1.1(1) functionality is met by invoking the underlying platforms' functionality as well as implementing the functionality for the Android Hub Agent's verification of signed profiles (policies).

[MDMPP] FIA_X509_EXT.1.2(1) – *"The evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.4.5 of the ST states that the UEM Server platform's certificate validation service will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE.

The iOS and Android Hub Agents platforms' certificate validation services will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE, except for certificates related to signed profiles (policies) processed by the Android Hub Agent. The instance of OpenSSL on the Android Hub Agent will ensure that all certificate paths terminate with a CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE for certificates related to signed profiles (policies).

This adequately describes how the FIA_X509_EXT.1.2(1) functionality is invoked.

[MDMPP] FIA_X509_EXT.2.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.4.5 of the ST states that the UEM Server relies on the underlying platform to provide X.509v3 certificate services for verification of the code signing of UEM Server software updates, for integrity verification of the UEM Server software, and signing profiles (policies) that are sent to iOS and Android Hub Agents.

Section 8.4.5 of the ST also states that the iOS and Android Hub Agents rely on the underlying mobile platforms' cryptographic modules to provide X.509v3 certificate services for verification of the code signing of Hub Agent software updates and in support of HTTPS/TLS connections to the Hub Agents.

Additionally, for the iOS Hub Agent only, the platform implements X.509v3 certificate services for the verification of signed profiles (policies) received from the UEM Server that will be applied by the iOS Hub Agent or iOS/iPadOS underlying platform.

This adequately describes how the FIA_X509_EXT.2.1 functionality is invoked.

[MDMPP] FIA_X509_EXT.2.2 – *"The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

If "implement functionality" is selected, the evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described."

Section 8.4.5 of the ST states that the UEM Server relies on its platform to provide its X.509v3 certificate as part of TLS and HTTPS/TLS session establishment in all cases where the UEM is the server component in the session as well as upon the server's component's request where the UEM Server is the client component and mutual authentication has been configured. The platform knows which certificate to use based upon the certificate being bound to the ports used for these protocols. Section 6.1 of the AGD under Steps 10 & 11 provide instructions for configuring the operating environment so that the TOE can use the certificates.

Section 8.4.5 of the ST states that the UEM server uses certificates for code signing of UEM Server software updates, for integrity verification of the UEM Server software, and signing profiles (policies) that are sent to iOS and Android Hub agents. The underlying platform knows which certificate to use for policy signing based upon an authorized Administrator uploading the certificate specifically for this purpose. The underlying platform knows which certificate to use for code signing and integrity verification based upon the presented certificate's data. Section 6.1 of the AGD under Steps 8, 33, and 34 provide instructions for configuring the operating environment so that the TOE can use the certificates. Certificates used for code signing of UEM Server software updates and for integrity verification of the UEM Server software do not require administrative action as well-known certificates are used.

Section 8.4.5 of the ST states that the iOS and Android Hub Agents rely on the underlying mobile platforms' cryptographic modules to provide X.509v3 certificate services for verification of the code signing of Hub Agent software updates and in support of HTTPS/TLS connections to the Hub Agents. The underlying mobile platforms known which certificate to use for code signing based upon the presented certificate's data and which certificate to use for HTTPS/TLS based upon the certificate issued to the mobile device for this purpose as part of meeting the FIA_X509_EXT.5 requirement. Sections 6.2.6 and 6.2.7 describe the configuration the operating environment so that the TOE can use the unique X.509v3 certificates issued to Android and iOS/iPadOS devices used for HTTPS/TLS connections. Certificates used for verification of the code signing of Hub Agent software updates do not require administrative action as well-known certificates are used.

Section 8.4.5 of the ST states that the iOS Hub agent platform relies on the underlying mobile device to provide X.509v3 certificate services for verification of signed profiles (policies). Meanwhile, the Android Hub agent provides X.509v3 certificate services for verification of signed profiles (policies). In both instances, the iOS Hub agent platform and Android Hub agent know which certificate to use based upon the presented certificate's data. Section 6.1 of the AGD under Steps 8, 33, and 34 provide instructions for configuring the operating environment so that the TOE can use the certificates.

Certificate validity is verified using OCSP by the UEM Server's TOE platform. If the UEM Server's TOE platform cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.

Certificate validity is verified using OCSP TLS Status Request Extension (i.e., OCSP stapling) by the iOS Hub agent platform for policy signing and TLS. If the iOS Hub Agent platform cannot establish a connection to determine the validity of a certificate, then the certificate is accepted.

Certificate validity is verified using OCSP by the Android Hub agent platform for TLS. If the Android Hub agent platform cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.

Certificate validity is verified using OCSP by the Android Hub agent for policy signing. If the Android Hub agent cannot establish a connection to determine the validity of a certificate, then the certificate is not accepted.

This adequately describes how the FIA_X509_EXT.2.2 functionality is met by invoking the underlying platforms' functionality as well as implementing the functionality for the Android Hub Agent's verification of signed profiles (policies).

[MDMPP] FIA_X509_EXT.5.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).*

If "implement functionality" is selected then the evaluator shall examine the TSS to verify that it describes the methods to ensure that each client utilizes a unique certificate."

Section 8.4.6 of the ST states for Android devices, the UEM Server retrieves a specific SCEP challenge for that device from the CA Server. The UEM Server will then bundle the SCEP challenge and the SCEP Server's URL into a payload that is sent to the Android Hub Agent. For iOS devices, the UEM Server will request a unique device certificate over DCOM and include the certificate within the MDM profiles (policies) assigned to the device.

Android devices use SCEP to generate a unique client certificate request. After receiving the payload with the SCEP Server's URL and SCEP challenge, the Android Hub Agent will send the payload to the device. The device will then request its unique X.509 certificate from the SCEP Server. Each iOS device receives its unique X.509 certificate within the MDM profiles (policies) assigned to the device.

The ST does not select "implement functionality" for this SFR, therefore, this portion of the TSS assurance activity is not applicable.

This adequately describes how the FIA_X509_EXT.5.1 functionality is invoked.

[MDMPP] FMT_MOF.1.1(1) – *"The evaluator shall examine the TSS and user documents to ensure that they describe what security management functions are restricted to the administrator and what actions can be taken for each management function. The evaluator shall verify that the security management functions are restricted to authorized administrators and the administrator is only able to take the actions as described in the user documents."*

Section 8.5.1 of the ST lists the security management functions that an Authorized Administrator can perform. The specific actions for each management function are further described in the SFR where that function is defined.

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.

[MDMPP] FMT_MOF.1.1(2) – *"The evaluator shall examine the TSS and verify that it describes how unauthorized users are prevented from enrolling in the MDM services."*

Section 8.5.2 of the ST states that in order to enroll any device, valid user credentials are required which are verified by the UEM Server and/or the external Active Directory/LDAP Server. This can be performed by either the MD user or an Authorized Administrator. Since Authorized Administrators are responsible for the creation of MD user accounts on UEM Server, they are able to perform first-use actions requiring user authentication prior to the MD user accessing the account and changing their password.

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

[MDMPP] FMT_MOF.1.1(3) – *"The evaluator shall examine the TSS to determine that all methods of initiating an application download or update push are specified."*

Section 8.5.3 of the ST states that there are two methods for making apps available to a device or restricting apps from the device: “smart groups” and application groups. Based upon the configuration of these methods determines if an app download can be initiated or an update push.

For smart groups, the Authorized Administrator has the ability to assign one or more smart groups to the app to push it to a set of devices or make it available to be downloaded by them.

For application groups, the Authorized Administrator can assign apps to an allow list or deny list and assign these application groups to users and/or organizational groups. By doing so, this will restrict or allow an MD user from initiating the download of an internal app.

[MDMPP] FMT_POL_EXT.1.1 – *“Policies must be digitally signed by the enterprise using the algorithms in FCS_COP.1(3). The evaluator shall ensure that the TSS describes how policies are digitally signed by the TSF.”*

Section 8.5.4 of the ST states the UEM Server will digitally sign the policies with an X.509 certificate and the signature will be validated by the entity receiving the policy before to the entity applies the policy to the device. All policies are signed by the UEM Server with a trusted root certificate using ECDSA with SHA-512; which is an algorithm that has been claimed in FCS_COP.1(3).

[AGENTMOD] FMT_POL_EXT.2 – *“The evaluator ensures that the TSS describes how the candidate policies are obtained by the MDM Agent, the processing associated with verifying the digital signature of the policy updates, and the actions that take place for successful (signature was verified) and unsuccessful (signature could not be verified) cases. The software components that are performing the processing must also be identified in the TSS and verified by the evaluators.”*

Section 8.5.5 of the ST describes how candidate policies are obtained by the MDM Agent via assignment by the UEM Server. The verification (processing and actions taken for successful and unsuccessful verification) of the digital signature is described for the Android and iOS Agents. The software performing the processing (i.e. OpenSSL for Android and underlying platform for iOS/IPadOS) has been evaluated in FMT_POL_EXT.2, FIA_X509_EXT.1(1), and FIA_X509_EXT.2 test assurance activities.

[MDMPP] FMT_SMF.1.1(1)/ANDROID – TD0479 – *“The evaluator shall examine the TSS to ensure that it describes each management function claimed. The evaluator shall examine the TSS to ensure that it identifies the management functions implemented for each supported MDM Agent/platform, which are likely to be subsets of all of the management functions available to the administrator on the MDM Server. The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported MDM Agent/platform are clearly indicated.*

The evaluator shall determine if the Mobile Device has been evaluated. If so, the evaluator shall examine the TSS to verify that it clearly identifies which management functions match the selections and assignments of the evaluated Mobile Device and which management functions were not evaluated.”

Section 8.5.6 of the ST provides Table 20 which includes a description of each management function that is claimed. The table clearly identifies the management functions implemented for each supported MDM Agent/platform and any differences between Android and iOS.

The table also clearly identifies which management functions match the selections and assignments of the evaluated Mobile Device evaluations as well as which management functions were not evaluated as part of the Mobile Device evaluations.

[MDMPP] FMT_SMF.1.1(1)/IOS – TD0479 – *“The evaluator shall examine the TSS to ensure that it describes each management function claimed. The evaluator shall examine the TSS to ensure that it identifies the management functions implemented for each supported MDM Agent/platform, which are likely to be subsets of all of the management functions available to the administrator on the MDM Server.*

The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported MDM Agent/platform are clearly indicated.

The evaluator shall determine if the Mobile Device has been evaluated. If so, the evaluator shall examine the TSS to verify that it clearly identifies which management functions match the selections and assignments of the evaluated Mobile Device and which management functions were not evaluated."

Section 8.5.6 of the ST provides Table 20 which includes a description of each management function that is claimed. The table clearly identifies the management functions implemented for each supported MDM Agent/platform and any differences between Android and iOS.

The table also clearly identifies which management functions match the selections and assignments of the evaluated Mobile Device evaluations as well as which management functions were not evaluated as part of the Mobile Device evaluations.

[MDMPP] FMT_SMF.1(2)/ANDROID – *"The evaluator shall examine the TSS to ensure that it describes each management function listed. For function c.4, the evaluator shall examine the TSS to ensure that it describes the privacy-sensitive information that the TOE has the capability to collect from enrolled mobile devices."*

Section 8.5.7 of the ST describes each of the claimed management functions listed in the SFR. A description of the privacy-sensitive information that the TOE has the capability to collect from enrolled mobile devices is also provided: GPS, Telecom, Applications, Profiles, and Network data information.

[MDMPP] FMT_SMF.1(2)/IOS – *"The evaluator shall examine the TSS to ensure that it describes each management function listed. For function c.4, the evaluator shall examine the TSS to ensure that it describes the privacy-sensitive information that the TOE has the capability to collect from enrolled mobile devices."*

Section 8.5.7 of the ST describes each of the claimed management functions listed in the SFR. A description of the privacy-sensitive information that the TOE has the capability to collect from enrolled mobile devices is also provided: GPS, Telecom, Applications, Profiles, and Network data information.

[MDMPP] FMT_SMF.1.1(3) – *"The evaluator shall examine the TSS to ensure that it describes each management function listed.*

The evaluator shall examine the TSS to determine if the MAS Server creates its own groups or relies on the groups specified by the MDM Server."

Section 8.5.8 of the ST describes how the MAS Server is capable of configuring application access groups and the downloading of applications from the Apple App Store (iOS) and Google Play Store (Android).

[AGENTMOD] FMT_SMF_EXT.4 – *"The evaluator shall verify that the any assigned functions are described in the TSS and that these functions are documented as supported by the platform. The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported mobile device are listed.*

The evaluator shall verify that the TSS describes the methods in which the MDM Agent can be enrolled.

The TSS description shall make clear if the MDM Agent supports multiple interfaces for enrollment and configuration (for example, both remote configuration and local configuration)."

Section 8.5.9 of the ST describes how certificates used for authentication of MDM Agent communications are imported during enrollment into management, how administrator-provided device management functions defined in FMT_SMF.1(1)/IOS and FMT_SMF.1(1)/ANDROID are received and processed by the Agents, whether users can unenroll from management, and the configuration of the periodicity of reachability events.

Sections 8.4.1 and 8.4.2 of the ST describe how the Android and iOS Hub agents are used to enroll the mobile devices into management. These sections fully describe the enrollment process which only describes a single interface for enrollment between the MD user/Authorized Administrator as well as between the Hub Agent and UEM Server. The sections also describe options for configuring user authentication and restricting enrollment of devices.

[MDMPP] FMT_SMR.1.1(1) – This SFR does not contain any MDMPP TSS Assurance Activities.

[MDMPP] FMT_SMR.1.2(1) – *"The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role."*

Section 8.5.10 of the ST defines an Authorized Administrator as follows:

- 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.

There are four admin roles: Server primary administrator, security configuration administrator, device user group administrator, and auditor. 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.

[MDMPP] FMT_SMR.1.1(2) – This SFR does not contain any MDMPP TSS Assurance Activities.

[MDMPP] FMT_SMR.1.2(2) – *"The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role."*

Section 8.5.11 of the ST states that the MAS Server is logically integrated with the UEM Server. It is accessed by Administrators using the Apps & Books > Applications > Native tab in the Admin Console. Since this is not accessed separately from the remainder of the UEM Server capabilities, the administrative roles that can interact with the MAS Server are defined in the same manner as for FMT_SMR.1(1) above. The UEM Server also maintains the roles of enrolled mobile devices and application access groups.

[AGENTMOD] FMT_UNR_EXT.1 – *"The evaluator shall ensure that the TSS describes the mechanism used to prevent users from unenrolling or the remediation actions applied when unenrolled."*

Section 8.5.12 of the ST states that for Android, users are prevented from unenrolling due to the "Block User Unenrollment" configuration which removes the unenrollment button and prevents the removal of the Android Hub agent through the Google Play Store.

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

[MDMPP] FPT_API_EXT.1.1 – *"The evaluator shall verify that the TSS lists the platform APIs used by the MDM software. The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported."*

Section 8.6.1 of the ST lists the platform APIs used by the UEM Server and the Android and iOS Hub Agents.

The list of supported APIs and their corresponding public documentation are as follows:

UEM Server:

- .NET API - https://docs.microsoft.com/en-us/dotnet/api/
- Diagnostic API (Windows event logs) - https://docs.microsoft.com/en-us/windows/win32/diagnostics
- Networking and Internet API (Microsoft Internet Information Services (IIS)) - https://docs.microsoft.com/en-us/iis-administration/
- Security and Identity API (Windows Authentication) - https://docs.microsoft.com/en-us/windows/win32/security

Android:

- java.security.KeyStore - https://developer.android.com/reference/java/security/KeyStore
- Javax.net.HttpURLConnection - https://developer.android.com/reference/java/net/HttpURLConnection
- Javax.net.ssl - https://developer.android.com/reference/javax/net/ssl/package-summary
- KeyChain API (Android Platform Keystore) - https://developer.android.com/reference/android/security/KeyChain
- KeyMaster API - https://source.android.com/security/keystore/implementer-ref
- SCrypto - https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp3606.pdf
- BoringSSL - https://boringssl.googlesource.com/boringssl

iOS:

- Common Crypto - https://developer.apple.com/security/
- CoreCrypto - https://developer.apple.com/security/
- Security.framework - https://developer.apple.com/documentation/security

[MDMPP] FPT_ITT.1.1(2) – *"The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."

Section 8.6.2 of the ST states the UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and enrolled iOS and Android Hub Agents.

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

The protocols listed in the TSS are consistent with those specified in the requirement and are consistent with the selections of mutually authenticated TLS and HTTPS.

[MDMPP] FPT_LIB_EXT.1.1 – *"The evaluator shall verify that the TSS lists the libraries used by the MDM software. The evaluator shall verify that libraries found to be packaged with or employed by the MDM software are limited to those in the assignment."*

Section 8.6.3 of the ST specifies that the libraries used by the UEM Server are outlined in Appendix A, Section 9.1 of the ST. Libraries used by the TOE for Android are outlined in Appendix A, Section 9.2 of the ST. Libraries used by the TOE for iOS/iPadOS are outlined in Appendix A, Section 9.3 of the ST. The evaluation team reviewed the list of third-party libraries with the developer and documented the third-party libraries found during the review. The libraries documented in Appendix A match the list created by the evaluation team.

[MDMPP] FPT_TST_EXT.1.1 – This SFR does not contain any MDMPP TSS Assurance Activities.

[MDMPP] FPT_TST_EXT.1.2 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).*

If "implement functionality" is selected, the evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful e.g., hash not verified) cases."

Section 8.6.4 of the ST states the UEM Server .dll files and executable code are digitally signed using an X.509v3 certificate from a public CA certificate. During initial installation of the UEM Server and each time the server application is started, the native Windows Authenticode process is invoked to validate the integrity of the UEM Server. When successful, the TOE will start normally. If the validation fails, the native Windows Authenticode process will terminate the host process and the service will not start.

Section 8.6.4 also states that the UEM Server relies on the underlying Windows Server 2019 platform's own self-tests to verify that the platform and its underlying hardware are operating correctly. This includes the SymCrypt cryptographic module that belongs to the underlying Windows Server 2019 platform. This module performs its own power-up self-tests upon initial start-up, including cryptographic algorithm known answer tests (KATs) and an integrity verification check. The Section then references the Windows platform's Security Target for additional information on its self-test.

Section 8.6.4 also states these tests are sufficient to validate the correct operation of the TSF because they verify that the platform's cryptographic module which the UEM Server relies upon is operating correctly, the platform does an integrity check of the UEM Server's software, and that the Windows Server 2019 platform performs self-tests which confirm that the platform's own functionality as well as its underlying hardware's functionality do not have any anomalies that would cause the TOE's software to be executed in an unpredictable or inconsistent manner.

The evaluation team has determined that this information adequately describes how the FPT_TST_EXT.1 functionality is invoked in the UEM Server's underlying platform, demonstrates the integrity of the TOE's executable code, and the outcome of this integrity check. Additionally, the Section describes additional activities the underlying platform performs to provide a stable environment for the TOE to operate on and provides an argument that the evaluation team agrees with regarding why these self-tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised.

The ST does not select “implement functionality” for this SFR, therefore, this portion of the TSS assurance activity is not applicable.

[MDMPP] FPT_TUD_EXT.1.1 – This SFR does not contain any MDMPP TSS Assurance Activities.

[MDMPP] FPT_TUD_EXT.1.2 – *“If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”*

Section 8.6.5 of the ST states the UEM Server software updates are installed by the underlying platform directly onto the system; the platform does not have an automatic method of pulling down or installing the updates without Authorized Administrator (local authorized administrator of the platform) initiation via the platform. The updates are digitally signed using a Digicert X.509v3 certificate which is installed in the Windows trusted key store on the underlying platform which verifies the software updates.

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

This adequately describes how the FPT_TUD_EXT.1.2 functionality is invoked.

[MDMPP] FPT_TUD_EXT.1.3 – *“If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).*

If "implement functionality" is selected, the evaluator shall examine the TSS and verify that it describes the standards by which the updates are digitally signed and how the signature verification process is implemented.”

Section 8.6.5 of the ST states the UEM Server software updates are installed by the underlying platform directly onto the system; the platform does not have an automatic method of pulling down or installing the updates without Authorized Administrator (local authorized administrator of the platform) initiation via the platform. The updates are digitally signed using a Digicert X.509v3 certificate which is installed in the Windows trusted key store on the underlying platform which verifies the software updates.

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

This adequately describes how the FPT_TUD_EXT.1.3 functionality is invoked.

[MDMPP] FTA_TAB.1.1 – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).*

If "implement functionality" is selected, the TSS shall describe when the banner is displayed."

"Invoke platform-provided functionality" is not selected, as such, this portion of the TSS Assurance Activity is not applicable.

Section 8.7.1 of the ST states that the banner is displayed to both Administrators and users prior to authenticating to the UEM Server on their respective interfaces: Admin Console and the Self-Service Portal login pages.

[MDMPP] FTP_ITC_EXT.1.1 – *"The evaluator shall ensure that the TSS contains whether the MDM Server communication channel is internal or external to the TOE."*

Section 8.8.1 of the ST states that the iOS and Android Hub Agents are internal to the TOE, as such the communication between the MDM Server and the MDM agents is an internal communication channel.

[MDMPP] FTP_ITC.1.1(1) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.8.2 of the ST states that The UEM Server communicates with third-party systems that reside in the Operational Environment via trusted channels. In the evaluated configuration, the UEM Server connects with:

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

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

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This adequately describes how the FTP_ITC.1.1(1) functionality is invoked.

[MDMPP] FTP_ITC.1.2(1) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked* *(it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”*

Section 8.8.2 of the ST states that The UEM Server communicates with third-party systems that reside in the Operational Environment via trusted channels. In the evaluated configuration, the UEM Server connects with:

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

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

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This adequately describes how the FTP_ITC.1.2(1) functionality is invoked.

[MDMPP] FTP_ITC.1.3(1) – *“The evaluator shall examine the TSS to determine that the methods of communication with authorized IT entities are indicated, along with how those communications are protected.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”

Section 8.8.2 of the ST states that The UEM Server communicates with third-party systems that reside in the Operational Environment via trusted channels. In the evaluated configuration, the UEM Server connects with:

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

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

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This adequately describes how the FTP_ITC.1.3(1) functionality is invoked.

[MDMPP] FTP_TRP.1.1(1) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.8.3 of the ST states that the UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and Administrators attempting to connect to the Admin Console for the purposes of remote administration. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the Administrator's identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server's platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

This adequately describes how the FTP_TRP.1.1(1) functionality is invoked.

[MDMPP] FTP_TRP.1.2(1) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.8.3 of the ST states that the UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and Administrators attempting to connect to the Admin Console for the purposes of remote administration. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the Administrator's identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server's platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

This adequately describes how the FTP_TRP.1.2(1) functionality is invoked.

[MDMPP] FTP_TRP.1.3(1) – *"The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The* *evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”

Section 8.8.3 of the ST states that the UEM Server’s platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and Administrators attempting to connect to the Admin Console for the purposes of remote administration. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The UEM Server’s identity is validated through its X.509v3 certificate presented during TLS session establishment and the Administrator’s identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server’s platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

This is consistent with the protocol selections of TLS and HTTPS from the requirement.

This adequately describes how the FTP_TRP.1.3(1) functionality is invoked.

[MDMPP] FTP_TRP.1.1(2) – *“If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).”*

Section 8.8.4 of the ST states the UEM Server’s platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and users. These protocols are used to protect the data traversing the channel from disclosure and/or modification. This communication path is used to connect to the Self-Service Portal for the purposes of remote device registration and other self-service tasks as well as the enrollment of the iOS and Android Hub Agents into the TOE. The UEM Server’s identity is validated through its X.509v3 certificate presented during TLS session establishment and the user’s identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server’s platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This adequately describes how the FTP_TRP.1.1(2) functionality is invoked.

[MDMPP] FTP_TRP.1.2(2) – *"If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."*

Section 8.8.4 of the ST states the UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and users. These protocols are used to protect the data traversing the channel from disclosure and/or modification. This communication path is used to connect to the Self-Service Portal for the purposes of remote device registration and other self-service tasks as well as the enrollment of the iOS and Android Hub Agents into the TOE. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the user's identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server's platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This adequately describes how the FTP_TRP.1.2(2) functionality is invoked.

[MDMPP] FTP_TRP.1.3(2) – *"The evaluator shall examine the TSS to determine that the methods of remote enrollment are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of enrollment are consistent with those specified in the requirement, and are included in the requirements in the ST.*

If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity)."

Section 8.8.4 of the ST states the UEM Server's platform uses HTTPS/TLS (TLS v1.2) to provide a trusted communication between the UEM Server and users. These protocols are used to protect the data traversing the channel from disclosure and/or modification. This communication path is used to connect to the Self-Service Portal for the purposes of remote device registration and other self-service tasks as well as the enrollment of the iOS and Android Hub Agents into the TOE. The UEM Server's identity is validated through its X.509v3 certificate presented during TLS session establishment and the user's identity is validated through their authentication credentials presented to the UEM Server.

In the evaluated configuration, the UEM Server's platform will be configured to allow only the following cipher suites:

• TLS_RSA_WITH_AES_256_GCM_ SHA384 as defined in RFC 5288
• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

This is consistent with the protocol selections of TLS and HTTPS from the requirement.

This adequately describes how the FTP_TRP.1.3(2) functionality is invoked.

3 Operational Guidance Assurance Activities

The evaluation team completed the testing of the Operational Guidance, which includes the review of the *VMware Workspace ONE Unified Endpoint Management version 2209 Supplemental Administrative Guidance v1.0* (AGD) document and confirmed that the Operational Guidance contains all Assurance Activities as specified by the *Protection Profile for Mobile Device Management Version 4.0 [MDMPP]* and *PP-Module for MDM Agent Version 1.0 [AGENTMOD]*. The evaluators reviewed the MDMPP and AGENTMOD to identify the security functionality that must be discussed for the operational guidance. This is prescribed by the Assurance Activities for each SFR and the AGD SARs. The evaluators have listed below each of the SFRs defined in the MDMPP and AGENTMOD that have been claimed by the TOE (some SFRs are conditional or optional) as well as the AGD SAR, along with a discussion of where in the operational guidance the associated Assurance Activities material can be found. The AGD includes references to other guidance documents that must be used to properly install, configure, and operate the TOE in its evaluated configuration. The AGD and its references to other VMware Workspace ONE Unified Endpoint Management Version 2209 guidance documents were reviewed to assess the Operational Guidance Assurance Activities. The AGD contains references to these documents in Chapter 4 and these references can also be found below:

The following additional references are used in this section of the document:

[1] Installing Workspace ONE UEM for on-premises and SaaS deployments VMware Workspace ONE UEM 2209
[2] Upgrade Guide for on-premises and SaaS deployments VMware Workspace ONE UEM 2209
[3] Console Basics VMware Workspace ONE UEM 2209
[4] Directory Service Integration VMware Workspace ONE UEM 2209
[5] Certificate Authority Integrations VMware Workspace ONE UEM 2209
[6] Integration with Apple Business Manager VMware Workspace ONE UEM 2209
[7] Microsoft Windows 10 and Windows Server 2019 (version 1809) Operational and Administrative Guidance v3.0 (request from Microsoft)
[8] Apple iOS 14: iPhones and Apple iPadOS 14: iPads Common Criteria Configuration Guide Version 1.0, 2021-05-25
[9] Samsung Android 11 on Galaxy Devices Version: 7.0
[10] VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target, Version 1.0

[MDMPP] FAU_ALT_EXT.1 – *"The evaluator shall examine the guidance document and verify that it describes how the alerts can be configured, if configurable."*

Section 7.4.2 of the AGD describes how alerts are configured on the UEM Server for compliance policy violations (blacklisted apps, non-whitelisted apps, absence of required apps, last time a device communicated with the UEM Server, unapproved model/device manufacturer, and operating system) as well as for device enrollment and unenrollment.

[AGENTMOD] FAU_ALT_EXT.2/ANDROID – There are no AGD assurance activities for this SFR.

[AGENTMOD] FAU_ALT_EXT.2/IOS – There are no AGD assurance activities for this SFR.

[MDMPP] FAU_GEN.1.1(1) – *"The evaluator shall check the administrative guide and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described.*

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP including those listed in the Management section. The evaluator shall examine the administrative guide and make a determination of which administrative commands are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements."

Section 8.1.3 of the AGD lists a sample table of all the auditable events and required event types mandated by the PP. Within the sample table of auditable events, there are sample audit records for the success or failure of functions including but not limited to: configure Enterprise certificate to be used for signing policies, configure the devices specified by IMEI, serial number, specific device models, number of devices, manufacturer, and operating system allowed for enrollment, configure the TOE unlock banner, and configure the privacy-sensitive information that will and will not be collected from particular mobile devices.

The AGD was developed with the intent to provide the specific guidance for managing TOE functionality or a pointer to the necessary documentation as defined by the Intended Audience statement in Section 2:

"This document is intended for administrators responsible for installing, configuring, and/or operating VMware Workspace ONE UEM. Guidance provided in this document allows the reader to deploy the product in an environment that is consistent with the configuration that was evaluated as part of the product's Common Criteria (CC) testing process. It also provides the reader with instructions on how to exercise the security functions that were claimed as part of the CC evaluation. The reader is also expected to be familiar with the general operation of the VMware Workspace ONE UEM product. This supplemental guidance includes references to VMware's standard documentation set for the product and does not explicitly reproduce materials located there. This guidance also includes information on configuration of the behavior of the iOS Hub agent and Android Hub agent as well as the communications between these Hub agents and the UEM Server. However, these activities are still performed by administrators.

The reader is also expected to be familiar with the VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target and the general CC terminology that is referenced in it. This document references the Security Functional Requirements (SFRs) that are defined in the Security Target document and provides instructions on how to perform the security functions that are defined by these SFRs. The VMware Workspace ONE UEM product, as a whole, provides a great deal of security functionality but only those functions that were in the scope of the evaluation are discussed here. Any functionality that is not described here or in the VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target was not evaluated and should be exercised at the user's risk."

Based upon this statement, the PP author stated that the AGD was developed specifically for the scope of the Common Criteria evaluation. Through the assessment of the SFR Guidance Assurance Activities, the evaluation team confirmed that the AGD contained all management actions needed to install, configure, and operate the TOE as it relates to the SFRs. These administrative commands related to the SFRs can be found within Sections 6, 7 and 8 of the AGD. This covered all information provided in Section 7 but not

Sections 6 and 8. The other management actions in Section 6 are installation and configuration actions which occur as part of the setup of the TOE but are necessary for all TOE functions to operate. The remainder of Section 8 is not administrative commands but explaining the TOE's audit records. The remainder of the AGD document does not contain administrative actions but instead provides overall product and environment information.

[MDMPP] FAU_GEN.1.2(1) – *"The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2(1)."*

Section 8.1.2 of the AGD describes the format of the audit records generated by the TOE. A definition of each field is also provided. The description of the fields contains the information required in FAU_GEN.1.2(1): date and time of event, type of event, subject identity, (if relevant) the outcome (success or failure) of the event, and any additional required information defined in Table 13 in the ST.

[MDMPP] FAU_GEN.1.1(2) – *"The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field."*

Section 8.1.2 of the AGD describes the format of the audit records generated by the TOE. A definition of each field is also provided.

[MDMPP] FAU_GEN.1.2(2) – *"The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2(2)."*

Section 8.1.2 of the AGD describes the format of the audit records generated by the TOE. A definition of each field is also provided. The description of the fields contains the information required in FAU_GEN.1.2(2): date and time of event, type of event, and mobile device identity.

[AGENTMOD] FAU_GEN.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FAU_NET_EXT.1 – *"The evaluator shall verify that the guidance instructs administrators on the method of determining the network connectivity status of an enrolled agent."*

Section 7.3 of the AGD provides instructions for administrators on how to determine the network connectivity status of an enrolled agent via the device query function from the UEM Server or the "My Device" function from the Hub Agent.

[MDMPP] FAU_SAR.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FAU_SAR.1.2 – *"The evaluator shall check the AGD guidance and ensure that it describes how the administrator accesses the audit data and describes the format of the audit record."*

Section 8.1.2 of the AGD describes how audit data can be accessed via the Admin Console from the UEM Server or via the Syslog server. A detailed template of the audit record format is provided along with the definition of each field.

[AGENTMOD] FAU_SEL.1(2) – *"The evaluator shall examine the operational guidance to determine that it contains instructions on how to define the set of auditable events as well as explains the syntax for multi-value selection (if applicable). The evaluator shall also verify that the operational guidance shall identify those audit records that are always recorded, regardless of the selection criteria currently being enforced."*

Based on the ST the iOS Hub Agent, iOS Platform, and Android Hub Agent receive and apply a policy that identifies required auditing. Therefore, the selective auditing is based on the configuration and deployment of these policies. Section 7.4 and 7.4.1 of the AGD describes the policies and how to configure devices and policies.

[MDMPP] FAU_STG_EXT.1 – *"The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.*

The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server."

Section 8.1.1 of the AGD describes how audit data is logged locally and remotely to the audit server. Audit data is simultaneously transmitted to the Syslog Server as it is generated.

Section 6.1 (steps 27 through 29), describe how to establish the trusted channel to the audit server, including the protocol and version, as well as defining the audit record format.

[MDMPP] FCO_CPC_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FCO_CPC_EXT.1.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FCO_CPC_EXT.1.3 – *"The evaluator shall examine the guidance documentation to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component)."*

Section 6.2 of the AGD provides instructions on how to limit enrollment (enablement) to the UEM Server via device enrollment restrictions for Android and iOS. It also describes the enrollment process for Android and iOS. Finally, the section also provides instructions on how to unenroll Android and iOS devices from the UEM Server and prevent reenrollment (disablement).

[MDMPP] FCS_CKM.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_CKM.2.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_CKM_EXT.4.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_CKM_EXT.4.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_COP.1.1(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_COP.1.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_COP.1.1(3) – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_COP.1.1(4) – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_RBG_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_RBG_EXT.1.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FCS_STG_EXT.1.1 – There are no AGD assurance activities for this SFR.

[AGENTMOD] FCS_STG_EXT.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_ENR_EXT.1.1/ANDROID – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_ENR_EXT.1.2/ANDROID – *"The evaluator shall ensure that the administrative guidance describes the method(s) of restricting user enrollment and that it instructs the administrator how to configure the restrictions."*

Section 6.2.1 of the AGD describes the methods for restricting user enrollment for Android mobile devices based on the IMEI, serial number, device models, number of devices, manufacturer, and operating system.

[MDMPP] FIA_ENR_EXT.1.1/IOS – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_ENR_EXT.1.2/IOS – *"The evaluator shall ensure that the administrative guidance describes the method(s) of restricting user enrollment and that it instructs the administrator how to configure the restrictions."*

Section 6.2.2 of the AGD describes the methods for restricting user enrollment for iOS mobile devices based on DEP identifier.

[AGENTMOD] FIA_ENR_EXT.2 – *"The evaluator shall examine the operational guidance to verify that it describes how to configure reference identifier of the MDM Server's certificate and, if different than the reference identifier, the Domain Name or IP address (for connectivity) of the MDM Server."*

Android:

Section 6.2.6 of the AGD states that during enrollment, the Android Hub Agent will record the UEM Server's DNS name and full URL with hostname as the reference identifier for the UEM Server. The presented identifier in the UEM Server's certificate is the same as the reference identifier of the UEM Server.

iOS:

Section 6.2.7 of the AGD states that during enrollment, the iOS platform and iOS Hub Agent will record the UEM Server's DNS name and full URL with hostname as the reference identifier for the UEM Server. The presented identifier in the UEM Server's certificate is the same as the reference identifier of the UEM Server.

[MDMPP] FIA_UAU.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_UAU.1.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_X509_EXT.1.1(1) – TD0641 – *If "internal lookup of TOE-managed certificate status" is selected, then the evaluator shall ensure the AGD documentation describes how issued certificates are reported as invalid.*

"Internal lookup of TOE-managed certificate status" was not selected. Therefore, this AGD assurance activity does not apply.

[MDMPP] FIA_X509_EXT.1.2(1) – TD0641 – *If "internal lookup of TOE-managed certificate status" is selected, then the evaluator shall ensure the AGD documentation describes how issued certificates are reported as invalid.*

"Internal lookup of TOE-managed certificate status" was not selected. Therefore, this AGD assurance activity does not apply.

[MDMPP] FIA_X509_EXT.2.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FIA_X509_EXT.2.2 – *"If the requirement that the administrator is able to specify the default action is selected, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed."*

The requirement for UEM server and Android hub agent is that the certificate is not accepted if the connection to determine the validity of a certificate cannot be established. The requirement for the iOS hub agent is that it accepts the certificate. There is no ability provided by the TOE that these features can be configured. Therefore, no configuration is applicable for this action.

[MDMPP] FIA_X509_EXT.5.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_MOF.1.1(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_MOF.1.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_MOF.1.1(3) – *"The evaluator shall confirm that the operational guidance contains how to initiate an application download or update push."*

Section 7.5.3 of the AGD describes how to initiate an application download/push and application update download/push.

[MDMPP] FMT_POL_EXT.1.1 – *"If applicable, the evaluator shall verify that the AGD guidance instructs administrators on configuring the Enterprise certificate to be used for signing policies or signing the policies before applying them."*

Section 6.1 (steps 8 and 33 through 34) of the AGD provide steps on how to configure the Enterprise certificates to be used for signing policies sent to enrolled mobile devices.

[AGENTMOD] FMT_POL_EXT.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_SMF.1.1(1)/ANDROID – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_SMF.1.1(1)/IOS – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_SMF.1.1(2)/ANDROID – *"The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each management functional capability listed."*

Sections 6 and 7 of the AGD provide instructions on how to configure each management function defined for this SFR.

Specifically, section 6.1 (Step 10) provides steps for uploading and configuring the UEM Server's X.509v3 certificate. Configuration of the devices by IMEI, serial number, specific device models, number of devices, manufacturer, and operating system is described in section 6.2.1. The configuration of the TOE unlock banner is described in section 7.1.4. The configuration of the 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 is described in section 7.3. Configuration of the privacy-sensitive information that will and will not be collected from particular mobile devices is also described in section 7.3. Configuration of the interaction between TOE components is described in section 6.2. Configuration of the server administrator login session timeout is described in section 7.1.3. Configuration of the Enterprise certificate to be used for signing policies is described in section 6.1 (steps 8 and 33 through 34). The configuration of the MDM/Agent/platform to perform a network reachability test is described in section 7.3. The configuration of the transfer of MDM server logs to another server for storage, analysis, and reporting is described in section 6.1 (steps 27 through 29).

[MDMPP] FMT_SMF.1.1(2)/IOS – *"The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each management functional capability listed."*

Sections 6 and 7 of the AGD provide instructions on how to configure each management function defined for this SFR.

Specifically, section 6.1 (Step 10) provides steps for uploading and configuring the UEM Server's X.509v3 certificate. Configuration of the devices by DEP identifier is described in section 6.2.2. The configuration of the TOE unlock banner is described in section 7.1.4. The configuration of the 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 is described in section 7.3. Configuration of the privacy-sensitive information that will and will not be collected from particular mobile devices is also described in section 7.3. Configuration of the interaction between TOE components is described in section 6.2. Configuration of the server administrator login session timeout is described in section 7.1.3. Configuration of the Enterprise certificate to be used for signing policies is described in section 6.1 (steps 8 and 33 through 34). The configuration of the MDM/Agent/platform to perform a network reachability test is described in section 7.3. The configuration of the transfer of MDM server logs to another server for storage, analysis, and reporting is described in section 6.1 (steps 27 through 29).

[MDMPP] FMT_SMF.1.1(3) – *"The evaluator shall confirm that the operational guidance contains how to create and define user groups and how to specify which applications are accessible by which group.*

The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each management functional capability listed."

Section 7.5.1 of the AGD provides steps on how to create and define user groups in order to specify which applications are accessible by each group.

[AGENTMOD] FMT_SMF_EXT.4 – *"The evaluator shall verify the AGD guidance includes detailed instructions for configuring each function in this requirement.*

If the MDM Agent is a component of the MDM system (i.e. MDM Server is the Base-PP), the evaluator shall verify, by consulting documentation for the claimed mobile device platforms, that the configurable functions listed for this Agent are supported by the platforms.

If the MDM Agent supports multiple interfaces for configuration (for example, both remote configuration and local configuration), the AGD guidance makes clear whether some functions are restricted to certain interfaces."

Sections 6.2.6 and 6.2.7 of the AGD describe how to configure the certificates to be used for authentication of MDM Agent communications and how to enroll in management. Table 4 in section 7.4 of the AGD describes how to configure the administrator-provided device management functions. Sections 6.2.3 and 6.2.4 describe how to configure whether users can unenroll from management. Section 7.3 describes how to configure the periodicity of reachability events.

Table 4 in section 7.4 of the AGD lists all of the administrator-provided device management functions that may or may not be claimed by the underlying mobile device platform evaluation (VID11146 for iOS 14, VID11147 for iPadOS14, and VID11160 for Android). For each function listed in Table 4, it specifies under the “Claimed in VID11146/11147” or “Claimed in VID11160” column “Yes” if claimed and “No” if not claimed by the underlying mobile device platform evaluation. For functions where the value is “Yes”, the fact that it was evaluated as part of the mobile device platform evaluation means that it is supported by the platform. The evaluation team confirmed all Yes statements by reviewing the mobile devices’ Security Targets. For functions where the value is “No”, mobile device documentation which covers the function or a justification explaining why it would not be covered in the mobile device documentation is listed below:

Android:

- Unenroll from management – N/A – Not handled by the platform, logic is implemented in the Android Hub Agent (TOE).
- Install policies – N/A – Not handled by the platform, logic is implemented in the Android Hub Agent (TOE).
- Query connectivity status – https://developer.android.com/reference/android/net/ConnectivityManager
- Query the current version of the MD firmware/software – https://developer.android.com/reference/android/os/Build
- Query the current version of the hardware model of the device – https://developer.android.com/reference/android/os/Build
- Query the current version of installed mobile applications – https://developer.android.com/reference/android/content/pm/PackageManager
- Alert the user – https://developer.android.com/reference/android/app/NotificationManager#notify(int,%20android.app.Notification)
- Retrieve MD-software integrity verification values - The TOE uses the SafetyNet Attestation API which provides a cryptographically-signed attestation, assessing the device's integrity. Uses the devices custom settings to import the SafetyNet settings.
- Place applications into application process groups – N/A – This function strictly has to do with managing the availability of which mobile applications an enrolled mobile device may download from the MAS server. This configuration is outside the scope of the mobile device and strictly configured on the MAS server.
- Enable/disable policy for Wi-Fi tethering, USB tethering, and/or Bluetooth tethering https://docs.samsungknox.com/devref/knox-sdk/reference/com/samsung/android/knox/restriction/RestrictionPolicy.html#setTethering(boolean)
- Enable/disable policy for data display notification in the locked state – https://docs.samsungknox.com/devref/knox-sdk/reference/com/samsung/android/knox/container/KnoxConfigurationType.html#setKeyguardDisabledFeatures(int..html
- Enable/disable backup https://docs.samsungknox.com/devref/knox-sdk/reference/com/samsung/android/knox/restriction/RestrictionPolicy.html#setBackup(boolean)
- Enable/disable policy for user unenrollment – https://docs.samsungknox.com/devref/knox-sdk/reference/com/samsung/android/knox/EnterpriseDeviceManager.html#setAdminRemovable(boolean)
- Enable/disable automatic updates of system software – https://docs.samsungknox.com/devref/knox-sdk/reference/com/samsung/android/knox/restriction/RestrictionPolicy.html#allowOTAUpgrade(boolean)

iOS:

- Unenroll from management – N/A – Not handled by the platform, logic is implemented in the Android Hub Agent (TOE).
- Install policies – https://developer.apple.com/documentation/devicemanagement/install_a_profile

- Query connectivity status – https://developer.apple.com/documentation/devicemanagement/get_device_information
- Query the current version of the MD firmware/software – https://developer.apple.com/documentation/devicemanagement/get_device_information
- Query the current version of the hardware model of the device – https://developer.apple.com/documentation/devicemanagement/get_device_information
- Query the current version of installed mobile applications – https://developer.apple.com/documentation/devicemanagement/list_the_installed_apps
- Alert the user – https://developer.apple.com/documentation/usernotifications
- Place applications into application process groups – N/A – This function strictly has to do with managing the availability of which mobile applications an enrolled mobile device may download from the MAS server. This configuration is outside the scope of the mobile device and strictly configured on the MAS server.
- Revoke Biometric template – https://developer.apple.com/documentation/devicemanagement/install_a_profile
- Enable policy for data-at-rest protection https://developer.apple.com/documentation/devicemanagement/passcode/
- Enable/disable policy for local authentication bypass https://developer.apple.com/documentation/devicemanagement/clear_the_passcode/
- configure the Bluetooth trusted channel policy - https://developer.apple.com/documentation/devicemanagement/restrictions/
- Enable/disable USB mass storage mode and/or USB data transfer without user-authentication - https://developer.apple.com/documentation/devicemanagement/restrictions/
- Enable/disable backup https://developer.apple.com/documentation/devicemanagement/restrictions/
- Application installation policy – https://developer.apple.com/documentation/devicemanagement/install_an_enterprise_app
- iOS Hub Agent passcode authentication policy – N/A – This function strictly has to do with managing the iOS Hub Agent, which is not a function of the device platform and as such, will not be in device platform documentation.

Section 7.2 of the AGD states that all administration of VMware Workspace ONE UEM is performed through the Admin Console. As such, the MDM Agent is only configurable via the UEM Server console. There are no other interfaces over which the MDM Agent is configurable.

[MDMPP] FMT_SMR.1.1(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_SMR.1.2(1) – *"The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported."*

Sections 7.1 and 7.2 of the AGD describe and provide instructions for administering the TOE (including the various administrator roles) over the supported interfaces.

[MDMPP] FMT_SMR.1.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FMT_SMR.1.2(2) – *"The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported."*

Sections 7.1 and 7.2 of the AGD describe and provide instructions for administering the TOE (including the various administrator roles) over the supported interfaces.

[AGENTMOD] FMT_UNR_EXT.1 – *"The evaluator shall ensure that the administrative guidance instructs administrators in configuring the unenrollment prevention in each available configuration interface. If any configuration allows users to unenroll, the guidance also describes the actions that unenroll the Agent."*

Sections 6.2.3 and 6.2.4 of the AGD provide steps on how administrators can configure the unenrollment prevention by user configuration for Android and iOS mobile devices, respectively.

[MDMPP] FPT_API_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FPT_ITT.1.1(2) – *"The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method."*

Section 6.1 (steps 7, 11 and 26) of the AGD provide instructions on how to configure the mutually authenticated TLS/HTTPS between the UEM Server and the enrolled mobile devices.

[MDMPP] FPT_LIB_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FPT_TST_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FPT_TST_EXT.1.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FPT_TUD_EXT.1.1 – *"The evaluator shall ensure that the administrator guidance includes instructions for determining the current version of the TOE."*

Section 6.4.1 of the AGD provides instructions on how to determine the current version of the TOE.

[MDMPP] FPT_TUD_EXT.1.2 – There are no AGD assurance activities for this SFR.

[MDMPP] FPT_TUD_EXT.1.3 – *"The evaluator shall examine the AGD guidance to verify that it describes how to query the current version of the TSF software, how to initiate updates and how to check the integrity of updates prior to installation."*

Section 6.4.1 of the AGD provides instructions on how to determine the current version of the TOE. Sections 6.4.2 and 6.4.3 of the AGD provide instructions on how to initiate updates and check the integrity of updates prior to installation.

[MDMPP] FTA_TAB.1.1 – *"The evaluator follows the operational guidance to configure a notice and consent warning message."*

Section 7.1.4 of the AGD provides steps on how to configure the notice and consent warning message prior to authentication to the Admin Console and Self-Service Portal login pages.

[MDMPP] FTP_ITC_EXT.1.1 – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_ITC.1.1(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_ITC.1.2(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_ITC.1.3(1) – *"The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Server and authorized IT entities for each supported method."*

Section 6.1 (steps 12 through 13, 27) of the AGD provides instructions on how to configure the TOE to send audit data to the Syslog Server via the trusted channel.

Section 6.1 (step 24) of the AGD provides instructions on how to configure the TOE to send authentication requests to LDAP.

[MDMPP] FTP_TRP.1.1(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_TRP.1.2(1) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_TRP.1.3(1) – *"The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method."*

Section 7.1.2 of the AGD describes how to establish a connection to remotely administrate the TOE over the supported interface (Admin Console).

[MDMPP] FTP_TRP.1.1(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_TRP.1.2(2) – There are no AGD assurance activities for this SFR.

[MDMPP] FTP_TRP.1.3(2) – *"The evaluator shall confirm that the operational guidance contains instructions for establishing the enrollment sessions for each supported method."*

Sections 7.1.1 of the AGD describes how to establish a connection to remotely administrate the TOE over the supported interface (Self-Service Portal).

4 Test Assurance Activities (Test Report)

The following sections demonstrate that all ATE Assurance Activities for the TOE have been met. This evidence has been presented in a manner that is consistent with the "Reporting for Evaluations Against NIAP-Approved Protection Profiles" guidance that has been provided by NIAP. Specific test steps and associated detailed results are not included in this report in order for it to remain non-proprietary. The test report is a summarized version of the test activities that were performed as part of creating the Evaluation Technical Report (ETR).

The evaluation team conducted testing activities between November 2022 and February 2023. Testing was conducted at the Booz Allen CCTL in Laurel, MD on an isolated network.

4.1 Platforms Tested and Composition

Evaluator-conducted manual testing was completed in February 2023. The evaluation team set up a test environment for the independent functional testing that allowed them to perform all test assurance activities against the VMware Workspace ONE Unified Endpoint Management Version 2209 over the SFR relevant interfaces.

There was no sampling of testing, as all required test assurance activities were performed against the TOE for the three components (Workspace ONE Unified Endpoint Management 2209 (UEM Server), Android Intelligent Hub Agent 22.09 (Android Hub Agent), and iOS Intelligent Hub Agent 22.11 (iOS Hub Agent)) that were claimed in the Security Target. For testing, this evaluation used a Samsung Android 11 mobile device and an Apple iOS 14 mobile device.

The evaluation team performed testing of the TSF functionality across the claimed components as well as the Admin Console management interface. The full set of tests were developed to stimulate each applicable TSF relevant interface; which would fully test all combinations of the claimed platforms and their TSF relevant interfaces. The testing is consistent with the use of the interfaces defined within the ST. Thus, the testing of the interfaces was based upon testing SFR functionality related to user actions over each interface.

4.1.1 Test Configuration

The evaluation team conducted testing at the Booz Allen CCTL in Laurel, MD on an isolated network. The evaluation team configured the TOE for testing according to the *VMware Workspace ONE Unified*

Endpoint Management Supplemental Administrative Guidance Version 1.0 (AGD) document. The evaluation team set up a test environment for the independent functional testing that allowed them to perform the assurance activities against the TOE over the SFR relevant interfaces.
The TOE was configured to communicate with the following environment components:

- Active Directory / LDAP Server - Microsoft Windows Server 2019 (version 1809)
- Apple iOS 14 Mobile Device (VID 11146) - iPhone 12 Pro (128 GB Silver)
- Apple iPadOS 14 Mobile Device (VID 11147) – iPad Air (4th generation)
- Apple Push Notification Service (APNS) / Apple DEP
- Certification Authority (CA) Server - Microsoft Windows Server 2019 (version 1809)
- Firebase Cloud Messaging Service (FCM)
- Samsung Android 11 Mobile Device (VID 11160) - Samsung Galaxy S10, Samsung Galaxy S10e
- SQL database - Microsoft SQL Server 2019 Enterprise
- Syslog Server – Debian GNU/Linux 10 (buster) – rsyslogd 8.1901.0 (aka 2019.01)
- Windows Server 2019 - Microsoft Windows Server 2019 (version 1809)
- Workstation – Windows 10 Version 21H1

The following test tools were installed in the operational environment on multiple test workstations and servers for testing purposes:

- Fiddler Classic v.5.0.20211.51073
- Nmap version 7.80
- Wireshark version 3.4.9

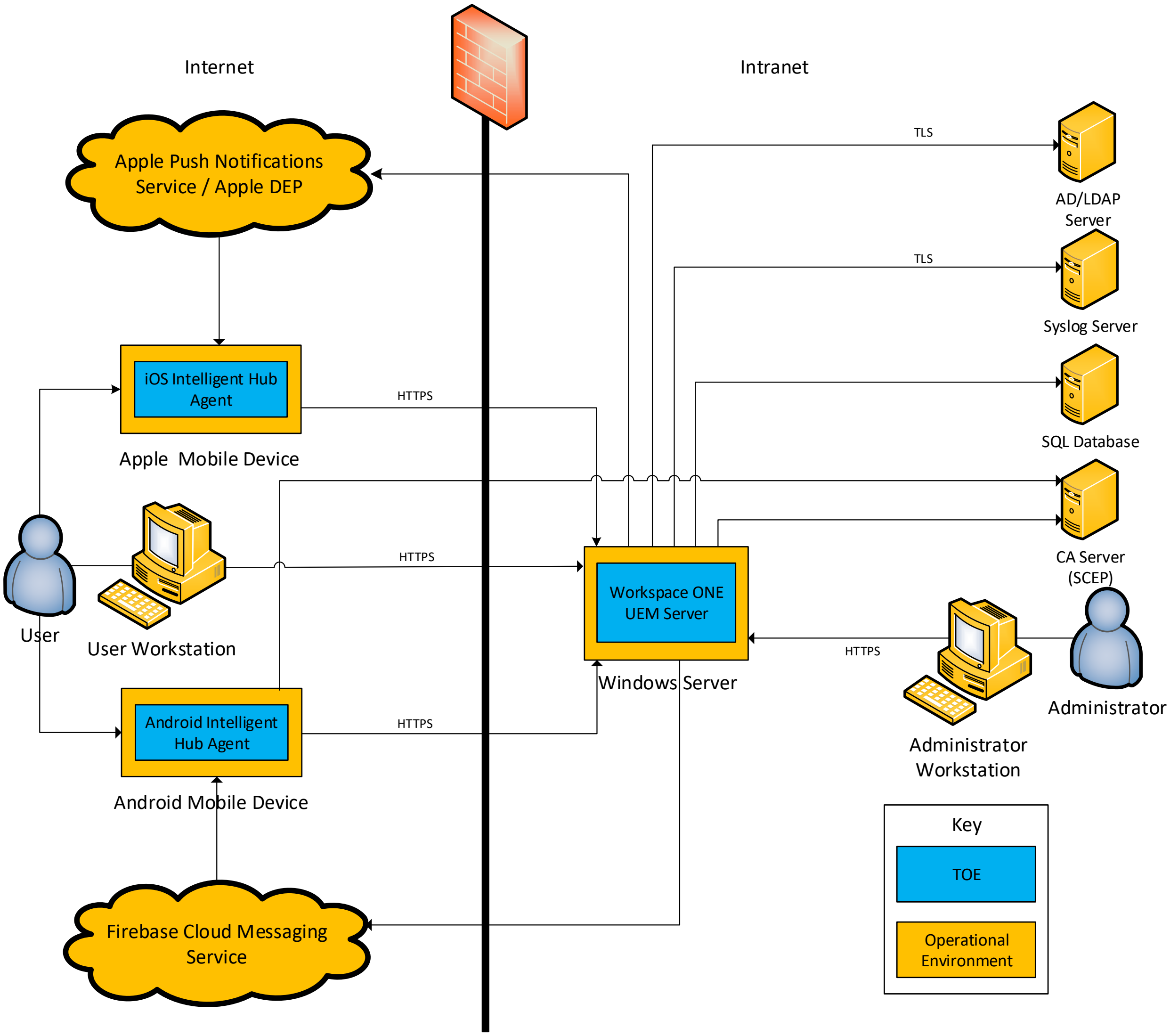

Figure 1 - Test Configuration

4.2 *Omission Justification*

There were no testing omissions because there was no sampling of testing, as all required test assurance activities were performed against the TOE as claimed in the Security Target.

4.3 *Test Cases*

The evaluation team completed the functional testing activities within the Booz Allen CCTL test environment, located at 1100 West St, Laurel, MD 20707. The evaluation team conducted a set of testing that includes all ATE Assurance Activities as specified by the *Protection Profile for Mobile Device Management Version 4.0 [MDMPP]* and *PP-Module for MDM Agent Version 1.0 [AGENTMOD]*. The evaluators reviewed the MDMPP and AGENTMOD to identify the security functionality that must be verified through functional testing. This is prescribed by the Assurance Activities for each SFR.

If an SFR is not listed, one of the following conditions applies:

- The Assurance Activity for the SFR specifically indicates that it is simultaneously satisfied by completing a test Assurance Activity for a different SFR.
- The Assurance Activity for the SFR does not specify any actions related to ATE activities (e.g. FCS_RBG_EXT.1).

Note that some SFRs do not have Assurance Activities associated with them at the element level (e.g. FMT_SMR.1.1(1)). In such cases, testing for the SFR is considered to be satisfied by completion of all Assurance Activities at the component level.

The following lists for each ATE Assurance Activity, the test objective, test instructions, test steps, and test results. Note that unless otherwise specified, the test configuration is to be in the evaluated configuration as defined by the AGD. For example, some tests require the TOE to be brought out of the evaluated configuration to temporarily disable cryptography to prove that the context of transmitted data is accurate. As part of the cleanup for each test, the TOE is returned to the evaluated configuration.

4.3.1 Security Audit

001A	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 1
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluator shall enroll a device and ensure that the MDM server alerts the administrator of the change in enrollment status. The evaluator shall unenroll (retire) a device and ensure that the MDM server alerts the administrator of the change in enrollment status.

Enrollment:

1. Enroll the mobile device to the MDM system by following the procedures in [MDMPP]FIA_ENR_EXT.1.1/ANDROID – Test Case 028.
2. Verify that the MDM server alerts the administrator of the device enrollment.

Unenrollment:

1. Unenroll the mobile device from the MDM system.
   a. Authenticate to the MDM Console as the administrator.
   b. Navigate to “Devices” > “List View”.
   c. Choose the specific Device to execute Delete Device under the “General Info” column.
   d. Choose “More Actions” from the top right-hand menu then “Delete Device” under the Admin heading.
   e. Confirm the unenrollment request.
   f. Verify that the device has been unenrolled from management.
   g. Launch the MDM Console and authenticate as the administrator.
   h. Navigate to “Devices” > “List View”.
   i. Verify that the MDM server alerts the administrator of the device unenrollment.

Test Results:	For each mobile device platform, the evaluator enrolled a device into the MDM server. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each enrollment. Then, the evaluator performed the unenrollment operation for each enrolled mobile device as part of this test. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each unenrollment. Additionally, audit records were successfully generated for each enrollment and unenrollment operation. - PASS
Execution Method:	Manual

001I	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 1
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluator shall enroll a device and ensure that the MDM server alerts the administrator of the change in enrollment status. The evaluator shall unenroll (retire) a device and ensure that the MDM server alerts the administrator of the change in enrollment status.

Enrollment:

1. Enroll the mobile device to the MDM system by following the procedures in [MDMPP]FIA_ENR_EXT.1.1/IOS – Test Case 031.
2. Verify that the MDM server alerts the administrator of the device enrollment.

Unenrollment:

1. Unenroll the mobile device from the MDM system.
    a. Authenticate to the MDM Console as the administrator.
    b. Navigate to “Devices” > “List View”.
    c. Choose the specific Device to execute Device Wipe under the “General Info” column.
    d. Choose “More Actions” from the top right-hand menu then “Device Wipe” under the Management heading.
    e. Confirm the unenrollment request.
    f. Verify that the device has been unenrolled from management.
    g. Launch the MDM Console and authenticate as the administrator.
    h. Navigate to “Devices” > “List View”.
    i. Verify that the MDM server alerts the administrator of the device unenrollment.

Test Results:	For each mobile device platform, the evaluator enrolled a device into the MDM server. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each enrollment. Then, the evaluator performed the unenrollment operation for each enrolled mobile device as part of this test. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each unenrollment. Additionally, audit records were successfully generated for each enrollment and unenrollment operation. - PASS
Execution Method:	Manual

001P	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 1
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluator shall enroll a device and ensure that the MDM server alerts the administrator of the change in enrollment status. The evaluator shall unenroll (retire) a device and ensure that the MDM server alerts the administrator of the change in enrollment status.

Enrollment (iPad OS):

1. Enroll the mobile device to the MDM system by following the procedures in [MDMPP]FIA_ENR_EXT.1.1/IOS – Test Case 031.
2. Verify that the MDM server alerts the administrator of the device enrollment.

Unenrollment:

1. Unenroll the mobile device from the MDM system.
    a. Authenticate to the MDM Console as the administrator.
    b. Navigate to “Devices” > “List View”.
    c. Choose the specific Device to execute Device Wipe under the “General Info” column.
    d. Choose “More Actions” from the top right-hand menu then “Device Wipe” under the Management heading.
    e. Confirm the unenrollment request.
    f. Verify that the device has been unenrolled from management.
    g. Launch the MDM Console and authenticate as the administrator.
    h. Navigate to “Devices” > “List View”.
    i. Verify that the MDM server alerts the administrator of the device unenrollment.

Test Results:	For each mobile device platform, the evaluator enrolled a device into the MDM server. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each enrollment. Then, the evaluator performed the unenrollment operation for each enrolled mobile device as part of this test. The evaluator confirmed that the MDM server alerted the administrator (via e-mail) of each unenrollment. Additionally, audit records were successfully generated for each enrollment and unenrollment operation. - PASS
Execution Method:	Manual

002A	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 2
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 2: The evaluator shall configure policies, which the MDM agent should not be able to apply. These policies shall include: • a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist • a valid configuration setting with an invalid parameter, which may require manual modification of the policy prior to transmission to the device The evaluator shall deploy such policies and verify that the MDM server alerts the administrator about the failed application of the policy.

A setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs:

> **This portion of this assurance activity is not applicable because no such setting exists. All Android devices claimed by this evaluation can process all policy settings claimed by this evaluation which are configured on the UEM Server.**

A valid configuration setting with an invalid parameter, which may also require manual modification of the policy prior to transmission to the device:

1. Authenticate to the MDM Server Console as the Administrator.
2. On the top right navigation bar, click “Add” > “Profile”.
3. Choose the “Android” platform.
4. Specify a name for the Profile.
5. Choose “Custom Settings” > “Add”.
6. Specify “<test>invalid</test>” in the text box.
7. Click “Next”, specify the assignment, then “Save & Publish”.
8. Verify that the profile failed to apply due to a valid configuration setting with an invalid parameter.

Test Results:	The first portion of this test (a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist) was not applicable because the evaluator determined that no such setting existed. For the second portion of this test, the evaluator created a new policy which contained an invalid parameter and assigned it to the mobile device. The evaluator confirmed that the MDM server alerted the administrator of the failed application of the policy both via observation and audit records. - PASS
Execution Method:	Manual

002I	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 2
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 2: The evaluator shall configure policies, which the MDM agent should not be able to apply. These policies shall include: • a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist • a valid configuration setting with an invalid parameter, which may require manual modification of the policy prior to transmission to the device The evaluator shall deploy such policies and verify that the MDM server alerts the administrator about the failed application of the policy.

A setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs:

1. Authenticate to the MDM Console as the Administrator.
2. On the top right navigation bar, click “Add” > “Profile”.
3. Choose the “Apple iOS” platform and “Device Profile”.
4. On the left navigation pane, choose “General”.
5. Specify a name for the Profile.
6. Specify the set of devices in the “Smart Groups”.
7. Click “VPN” then “Configure”.
8. In “Connection Name” block, give configuration organization defined name.
9. In “Connection Type” dropdown, select “LT2P”.
10. In “Server” field, type in hostname of VPN Server.
11. Check box for “Per-App VPN Rules”.
12. Click “Save & Publish”.
13. Confirm the devices/users for the policy to be assigned to, and then click “Publish”.
14. Verify that the profile failed to apply due to the target platform incompatibility.

A valid configuration setting with an invalid parameter, which may also require manual modification of the policy prior to transmission to the device:

1. Authenticate to the MDM Console as the Administrator.
2. On the top right navigation bar, click “Add” > “Profile”.
3. Choose the “Apple iOS” platform and “Device Profile”.
4. On the left navigation pane, choose “General”.
5. Specify a name for the Profile.
6. On the left navigation pane, choose “Custom Settings”.
7. Click “Configure” and then enter “invalidSetting” in the text box.
8. Click “Save & Publish”.
9. Verify that the profile failed to apply due to a valid configuration setting with an invalid parameter.

Test Results:	The evaluator performed this test by creating a new policy containing a setting configurable on the MDM server interface, but not supported by the platform (iOS 14) on which the MDM agent runs. The evaluator then assigned the policy to the specified mobile device and observed that the MDM server alerted the administrator of the failed application of the policy as well as confirmation of the failure via audit records. For the second portion of this test, the evaluator created a new policy which contained an invalid parameter and assigned it to the same mobile device. The evaluator confirmed that the MDM server alerted the administrator of the failed application of the policy both via observation and audit records. - PASS
Execution Method:	Manual

002P	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 2
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 2: The evaluator shall configure policies, which the MDM agent should not be able to apply. These policies shall include: • a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist • a valid configuration setting with an invalid parameter, which may require manual modification of the policy prior to transmission to the device The evaluator shall deploy such policies and verify that the MDM server alerts the administrator about the failed application of the policy.

A setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs:

1. Authenticate to the MDM Console as the Administrator.
2. On the top right navigation bar, click “Add” > “Profile”.
3. Choose the “Apple iOS” platform and “Device Profile”.
4. On the left navigation pane, choose “General”.
5. Specify a name for the Profile.
6. Specify the set of devices in the “Smart Groups”.
7. Click “VPN” then “Configure”.
8. In “Connection Name” block, give configuration organization defined name.
9. In “Connection Type” dropdown, select “LT2P”.
10. In “Server” field, type in hostname of VPN Server.
11. Check box for “Per-App VPN Rules”.
12. Click “Save & Publish”.
13. Confirm the devices/users for the policy to be assigned to, and then click “Publish”.
14. Verify that the profile failed to apply due to the target platform incompatibility.

A valid configuration setting with an invalid parameter, which may also require manual modification of the policy prior to transmission to the device:

1. Authenticate to the MDM Console as the Administrator.
2. On the top right navigation bar, click “Add” > “Profile”.
3. Choose the “Apple iOS” platform and “Device Profile”.
4. On the left navigation pane, choose “General”.
5. Specify a name for the Profile.

6. On the left navigation pane, choose “Custom Settings”.
7. Click “Configure” and then enter “invalidSetting” in the text box.
8. Click “Save & Publish”.
9. Verify that the profile failed to apply due to a valid configuration setting with an invalid parameter.

Test Results:	The evaluator performed this test by creating a new policy containing a setting configurable on the MDM server interface, but not supported by the platform (iPadOS 14) on which the MDM agent runs. The evaluator then assigned the policy to the specified mobile device and observed that the MDM server alerted the administrator of the failed application of the policy as well as confirmation of the failure via audit records. For the second portion of this test, the evaluator created a new policy which contained an invalid parameter and assigned it to the same mobile device. The evaluator confirmed that the MDM server alerted the administrator of the failed application of the policy both via observation and audit records. - PASS
Execution Method:	Manual

003A	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 3
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 3: (Conditional) The evaluator shall trigger each of the events listed and ensure that the MDM Server alerts the administrator.

Presence of apps on deny list:

1. Install the Bank of America Mobile Banking (com.infonow.bofa) app on the mobile device.
2. Authenticate to the MDM Server console as the administrator.
3. Configure an application denylist:
   a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups”.
   b. Click “ADD GROUP”.
   c. Choose “Denylist” > “Android, and name it “Android Denylist”.
   d. Click “Add Application”.
   e. Specify the Application Name: “Bank of America Mobile Banking”.
   f. Specify the Application ID: “com.infonow.bofa”.
   g. Click “Next”.
   h. Specify the appropriate assignment group.
   i. Click “Finish”.
4. Configure a policy that detects the presence of apps on a deny list.
   a. Navigate to “Add” > “Compliance Policy” > “Android”.
   b. Choose “Application List” > “Contains Denied App(s)”.
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
5. Verify that the MDM server alerts the administrator of the presence of apps on a deny list.

Presence of apps not on allow list:

1. Install the Bank of America Mobile Banking (com.infonow.bofa) app on the mobile device.

2. Authenticate to the MDM Server console as the administrator.
3. Configure an application group that lists the Snapchat (com.snapchat.android) application on the application Allowlist.
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Allowlist” > “Android” and specify Name: “Android Allowlist”.
    c. Click “Add Application”.
    d. Specify “Snapchat” and “com.snapchat.android”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
4. Create a compliance policy to notify the Administrator of the presence of apps not on an allow list:
    a. Navigate to “Add” > “Compliance Policy” > “Android”.
    b. Choose “Application List” > “Contains Non-Allowed App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
5. Verify that the MDM server alerts the administrator of the presence of apps not on an allow list.

Absence of required apps:

1. Ensure that the Snapchap (com.snapchat.android) application is not installed on the mobile device.
2. Authenticate to the MDM Server console as the administrator.
3. Configure a required application list:
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups”.
    b. Click “ADD GROUP”.
    c. Choose “Required” > “Android, and name it “Android Required Apps”.
    d. Click “Add Application”.
    e. Specify the Application Name: “Snapchat”.
    f. Specify the Application ID: “com.snapchat.android”.
    g. Click “Next”.
    h. Specify the appropriate assignment group.
    i. Click “Finish”.
4. Configure a policy detects the required apps missing.
    a. Navigate to “Add” > “Compliance Policy” > “Android”.
    b. Choose “Application List” > “Does Not Contain Required App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
5. Verify that the MDM server alerts the administrator of the required apps missing.

Last time a device communicated with the MDM Server:

1. Authenticate to the MDM Server console as the administrator.
2. Configure a compliance policy that alerts if the mobile device has not last communicated with the MDM Server within one minute.
   a. Navigate to “Add” > “Compliance Policy” > “Android”.
   b. Choose “Device Last Seen” > “Not Within” > “1 Hours”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
3. Place the mobile device in airplane mode and wait 1 hour.
4. Verify that the MDM server alerts the administrator of the non-compliant device.

Unapproved device manufacturer:

1. Authenticate to the MDM Server console as the administrator.
2. Configure a compliance policy that only permits HTC models.
   a. Navigate to “Add” > “Compliance Policy” > “Android”.
   b. Choose “Device Manufacturer” > “Is Not” > “HTC”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
3. Verify that the MDM server alerts the administrator of the unapproved device manufacturer.

Unapproved operating system version:

1. Authenticate to the MDM Server console as the administrator.
2. Configure a compliance policy that prohibits all devices with Android version greater than Android 8.0.0.
   a. Navigate to “Add” > “Compliance Policy” > “Android”.
   b. Choose “OS Version” > “Greater Than” > “Android – Android 8.0.0”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
3. Verify that the MDM server alerts the administrator of the unapproved operating system version.

Audit processing failure:

1. Enable the feature flag to configure Logging Server Failure notifications.
2. Authenticate to the MDM Server console as the administrator.
3. Configure the Logging Server Failure notifications:

    a. Administrator -> Manage Account Settings -> Notifications.
    b. Toggle “Console and Email” for Logging Server Failure.

4. Disconnect network path between UEM server and the remote syslog server.
5. Stimulate the generation of audit records by re-authenticating to the UEM server.
6. Verify that the administrator is notified of the audit transmission failure via:
    a. An alert generated within the UEM console bell notification.
    b. An e-mail alert to the administrator.

Test Results:	The evaluator performed this test by configuring the MDM server to alert the administrator for each of the following conditional events independently: presence of apps on deny list, presence of apps not on allow list, absence of required apps, last time a device communicated with the MDM server, unapproved device manufacturer, unapproved operating system version, and audit processing failure. In each case, the evaluator observed that the MDM server alerted the administrator of the specified event and also confirmed that corresponding audit records were generated. - PASS
Execution Method:	Manual

003I	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 3
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 3: (Conditional) The evaluator shall trigger each of the events listed and ensure that the MDM Server alerts the administrator.

Presence of apps on deny list:

1. Configure a policy that lists the UFC (com.ufc.UFCTV) application on the application denylist.
    a. Navigate “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Denylist” > “Apple iOS” and specify Name: “Apple iOS Denylist”.
    c. Click “Add Application”.
    d. Specify “UFC” and “com.ufl.UFCTV”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
2. Create a compliance policy to notify the Administrator of the presence of apps on a deny list:
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “Application List” > “Contains Denied App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
3. Install the UFC (com.ufc.UFCTV) app on the mobile device.
4. Verify that the MDM server alerts the administrator of the presence of the application on a deny list.

Presence of apps not on allow list:

1. Configure a policy that lists the UFC (com.ufc.UFCTV) application on the application Allowlist.
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Allowlist” > “Apple iOS” and specify Name: “Apple iOS Allowlist”.
    c. Click “Add Application”.
    d. Specify “Weather” and “com.weather.TWC”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
2. Create a compliance policy to notify the Administrator of the presence of apps not on an allow list:
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “Application List” > “Contains Non-Allowed App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
3. Install the UFC (com.ufc.UFCTV) app on the mobile device.
4. Verify that the MDM server alerts the administrator of the presence of the application not on an allow list.

Absence of required apps:

1. Ensure that the Weather (com.weather.TWC) application is not installed on the mobile device.
2. Authenticate to the MDM Server console as the administrator.
3. Configure a policy that lists the Weather (com.weather.TWC) as a required installed application.
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Required” > “Apple iOS” and specify Name: “Apple iOS Required Apps”.
    c. Click “Add Application”.
    d. Specify “Weather” and “com.weather.TWC”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
4. Create a compliance policy to notify the Administrator of the absence of required apps:
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “Application List” > “Does Not Contain Required App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
5. Verify that the MDM server alerts the administrator of the missing application.

Last time a device communicated with the MDM Server:

1. Configure a compliance policy that alerts if the mobile device has not last communicated with the MDM

Server within one minute.

   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
   b. Choose “Device Last Seen” > “Not Within” > “61 Minutes”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.

2. Place the mobile device in airplane mode and wait 1 minute.
3. Verify that the MDM server alerts the administrator of the non-compliant device.

Unapproved model:

1. Configure a compliance policy that only permits Apple iOS – iPad models.
   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
   b. Choose “Model” > “Is Not” > “Apple iOS - iPad”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
2. Verify that the MDM server alerts the administrator of the unapproved model.

Unapproved operating system version:

1. Configure a compliance policy that prohibits all devices with Apple iOS version greater than iOS 11.0.0.
   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
   b. Choose “OS Version” > “Greater Than” > “Apple iOS – iOS 11.0.0”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
2. Verify that the MDM server alerts the administrator of the unapproved operating system version.

Audit processing failure:

Refer to [MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 3 (Test Case 003A)

Test Results:	The evaluator performed this test by configuring the MDM server to alert the administrator for each of the following conditional events independently: presence of apps on deny list, presence of apps not on allow list, absence of required apps, last time a device communicated with the MDM server, unapproved model, unapproved operating system version, and audit processing failure. In each case, the evaluator observed that the MDM server alerted the administrator of the specified event and also confirmed that corresponding audit records were generated. - PASS
Execution Method:	Manual

003P	[MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 3
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 3: (Conditional) The evaluator shall trigger each of the events listed and ensure that the MDM Server alerts the administrator.

Presence of apps on deny list:

5. Configure a policy that lists the UFC (com.ufc.UFCTV) application on the application denylist.
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Denylist” > “Apple iOS” and specify Name: “Apple iOS Denylist”.
    c. Click “Add Application”.
    d. Specify “UFC” and “com.ufl.UFCTV”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
6. Create a compliance policy to notify the Administrator of the presence of apps on a deny list:
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “Application List” > “Contains Denied App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
7. Install the UFC (com.ufc.UFCTV) app on the mobile device.
8. Verify that the MDM server alerts the administrator of the presence of the application on a deny list.

Presence of apps not on allow list:

5. Configure a policy that lists the Weather (com.weather.TWC) application on the application Allowlist.
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
    b. Choose “Allowlist” > “Apple iOS” and specify Name: “Apple iOS Allowlist”.
    c. Click “Add Application”.
    d. Specify “Weather” and “com.weather.TWC”.
    e. Click “NEXT”.
    f. Specify the appropriate assignment group.
    g. Click “Finish”.
6. Create a compliance policy to notify the Administrator of the presence of apps not on an allow list:
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “Application List” > “Contains Non-Allowed App(s)”.
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.

g. Assign the policy to the appropriate assignment group.
h. Click “Next” and then “Finish & Activate”.

7. Install the UFC (com.ufc.UFCTV) app on the mobile device.
8. Verify that the MDM server alerts the administrator of the presence of the application not on an allow list.

Absence of required apps:

6. Ensure that the Weather (com.weather.TWC) application is not installed on the mobile device.
7. Authenticate to the MDM Server console as the administrator.
8. Configure a policy that lists the Weather (com.weather.TWC) as a required installed application.
   a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups” > “ADD GROUP”.
   b. Choose “Required” > “Apple iOS” and specify Name: “Apple iOS Required Apps”.
   c. Click “Add Application”.
   d. Specify “Weather” and “com.weather.TWC”.
   e. Click “NEXT”.
   f. Specify the appropriate assignment group.
   g. Click “Finish”.
9. Create a compliance policy to notify the Administrator of the absence of required apps:
   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
   b. Choose “Application List” > “Does Not Contain Required App(s)”.
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
10. Verify that the MDM server alerts the administrator of the missing application.

Last time a device communicated with the MDM Server:

1. Configure a compliance policy that alerts if the mobile device has not last communicated with the MDM Server within one minute.
   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
   b. Choose “Device Last Seen” > “Not Within” > “61 Minutes”
   c. Click “Next”.
   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.
2. Place the mobile device in airplane mode and wait 1 minute.
3. Verify that the MDM server alerts the administrator of the non-compliant device.

Unapproved model:

3. Configure a compliance policy that only permits Apple iOS – iPad models.
   a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.

    b. Choose “Model” > “Is Not” > “Apple iOS - iPhone”
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.

4. Verify that the MDM server alerts the administrator of the unapproved model.

Unapproved operating system version:

3. Configure a compliance policy that prohibits all devices with Apple iOS version greater than iOS 11.0.0.
    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose “OS Version” > “Greater Than” > “Apple iOS – iOS 11.0.0”
    c. Click “Next”.
    d. Choose “Notify” > “Send Email to Administrator”
    e. Specify the e-mail address of the administrator.
    f. Click “Next”.
    g. Assign the policy to the appropriate assignment group.
    h. Click “Next” and then “Finish & Activate”.
4. Verify that the MDM server alerts the administrator of the unapproved operating system version.

Audit processing failure:

Refer to [MDMPP]FAU_ALT_EXT.1.1 – Server Alerts – Test 3 (Test Case 003A)

Test Results:	The evaluator performed this test by configuring the MDM server to alert the administrator for each of the following conditional events independently: presence of apps on deny list, presence of apps not on allow list, absence of required apps, last time a device communicated with the MDM server, unapproved model, unapproved operating system version, and audit processing failure. In each case, the evaluator observed that the MDM server alerted the administrator of the specified event and also confirmed that corresponding audit records were generated. - PASS
Execution Method:	Manual

004	[AGENTMOD]FAU_ALT_EXT.2/ANDROID – Agent Alerts – Test 1
Test Objective:	Test 1: The evaluator shall perform a policy update from the test environment MDM server. The evaluator shall verify the MDM Agent accepts the update, makes the configured changes, and reports the success of the policy update back to the MDM Server.

1. Authenticate to the MDM Server console.
2. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
3. Specify the name of the profile.
4. Click the “Restrictions” tab and “Add”.
5. Uncheck “Allow Bluetooth” and “Allow Camera”.
6. Click “Next”, specify the smart group.
7. Click “Save & Publish”.
8. Verify on the mobile device that the profile is received and enforced.
9. Navigate to “Devices” > “Profiles & Resources” > “Profiles”.
10. Locate a profile that was already assigned to a device and click on the “Edit” (pencil icon).
11. Click “Add Version”.

12. Click on the “Restrictions” tab.
13. Check “Allow Camera”.
14. Click “Next”.
15. Click on “Save And Publish”.
16. Verify on the mobile device that Camera can be enabled.
17. Verify via audit records that the mobile device reports that the policy update is successfully received.

Test Results:	The evaluator performed this test by first configuring a base policy (i.e. initial policy) on the MDM server and then assigned it to the mobile device. The evaluator confirmed the mobile device received this policy. Next, the evaluator configured a policy update to the base policy on the MDM server and then assigned it to the same mobile device. Using audit records and functional observation, the evaluator confirmed that the mobile device received this policy, observed that the configuration defined in the policy update was accepted by the mobile device, and that the mobile device successfully reported the success of the policy update to the MDM server. - PASS
Execution Method:	Manual

005	[AGENTMOD]FAU_ALT_EXT.2/ANDROID – Agent Alerts – Test 2
Test Objective:	Test 2: The evaluator shall perform each of the actions listed in FAU_ALT_EXT.2.1 and verify that the alert does in fact reach the MDM Server.

Successful application of policies to a mobile device:

This is tested in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004.

Generating periodic reachability events:

This is tested in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 006.

Change in enrollment state:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 001A.

Failure to install an application from the MAS Server:

1. Upload an invalid test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click “Save and Assign”.
    f. Click “Add Assignment”, click in “Select Assignment Groups” box, and choose applicable Assignment Group of users/devices to assign application.
    g. Choose “AUTO” for “App Delivery Method”.
    h. Click “Add”.
    i. Click “Save and Publish”.
    j. Click “Publish”.
2. Verify that the MDM Agent notifies the MDM Server of the failure to install the application from the

MAS Server.

Failure to update an application from the MAS Server:

1. Upload a valid initial version of the test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click “Save and Assign”.
    f. Click “Add Assignment”, click in “Select Assignment Groups” box, and choose applicable Assignment Group of users/devices to assign application.
    g. Choose “AUTO” for “App Delivery Method”.
    h. Click “Add”.
    i. Click “Save and Publish”.
    j. Click “Publish”.
2. Verify that the initial version of the test application installs successfully.
3. Upload the application update version of the test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click “Save and Assign”.
    f. Verify the application assignment is correct.
    g. Click “Publish”.
4. Verify that the MDM Agent notifies the MDM Server of the failure to install the application update from the MAS Server.

Detection of apps on a deny list:

1. Install the Bank of America Mobile Banking (com.infonow.bofa) app on the mobile device.
2. Authenticate to the MDM Server console as the Administrator.
3. Configure an application denylist:
    a. Navigate to “Apps & Books” > “Applications” > “Application Settings” > “App Groups”.
    b. Click “ADD GROUP”.
    c. Choose “Denylist” > “Android, and name it “Android Denylist”.
    d. Click “Add Application”.
    e. Specify the Application Name: “Bank of America Mobile Banking”.
    f. Specify the Application ID: “com.infonow.bofa”.
    g. Click “Next”.
    h. Specify the appropriate assignment group.
    i. Click “Finish”.
4. Configure a policy that detects the presence of apps on a deny list.
    a. Navigate to “Add” > “Compliance Policy” > “Android”.
    b. Choose “Application List” > “Contains Denied App(s)”.
    c. Click “Next”.

   d. Choose “Notify” > “Send Email to Administrator”
   e. Specify the e-mail address of the administrator.
   f. Click “Next”.
   g. Assign the policy to the appropriate assignment group.
   h. Click “Next” and then “Finish & Activate”.

5. Create a policy to prevent the installation (disable if already installed) of apps on a deny list:
   a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
   b. Give the Profile a name.
   c. Click the “Application Control” tab and “Add”.
   d. Check “Disable Access to Denied Apps”.
   e. Click “Next”.
   f. Specify the smart group assignment.
   g. Click “Save & Publish”.
6. Verify that the MDM server alerts the administrator of the detection of the application on a deny list and that it is disabled on the mobile device.

Detection of apps not on an allow list:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003A.

Required apps missing:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003A.

Unapproved device manufacturer:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003A.

Unapproved operating system version:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003A.

Test Results:	The evaluator confirmed that all notifications were successfully received at the MDM server for each of the scenarios. – PASS
Execution Method:	Manual

006	[AGENTMOD]FAU_ALT_EXT.2/ANDROID – Agent Alerts – Test 3
Test Objective:	Test 3: The evaluator shall configure the MDM Agent to perform a network reachability test, both with and without such connectivity and ensure that results reflect each.

With Connectivity:

1. Launch the MDM Agent on the mobile device.
2. Tap “This Device” > “Sync Device” to perform a network reachability test to the MDM Server.
3. Authenticate to the MDM Server console.
4. Navigate to “Devices” > “List View”.
5. Under the “Last Seen” column for the mobile device, ensure that the time value is near zero (or near the elapsed time since the device sample was sent from the mobile device).

Without Connectivity: 1. Place the mobile device into Airplane Mode. 2. Launch the MDM Agent on the mobile device. 3. Tap “This Device” > “Sync Device” to perform a network reachability test to the MDM Server. 4. Authenticate to the MDM Server console. 5. Navigate to “Devices” > “List View”. 6. Under the “Last Seen” column for the mobile device, ensure that the time value is greater than the time value from when connectivity was established in the previous test. 7. Disable Airplane Mode on the mobile device. 8. Verify that audit records are generated for the failure of the MDM agent sending the reachability alert to the MDM server.	
Test Results:	The evaluator performed this test by first ensuring there was connectivity between the MDM server and the MDM agent and then initiated a device sync from the MDM agent. The evaluator observed that the "Last Seen" value was updated to a near zero value which was consistent with the expectation that the device had just communicated with the MDM server. The evaluator also verified that corresponding audit records were generated for this event. For the "without connectivity" portion of this test, the evaluator placed the mobile device into airplane mode, initated a device sync from the MDM agent and confirmed that the "Last Seen" value increased to a value greater than the previous value. Next, the evaluator disabled airplane mode on the mobile device and observed that audit records were generated for the failure of the MDM agent sending the reachability alert to the MDM server. – PASS
Execution Method:	Manual

007	[AGENTMOD]FAU_ALT_EXT.2/ANDROID – Agent Alerts
Test Objective:	Test 4: The evaluator shall remove network connectivity from the MDM Agent and generate an alert/event as defined in FAU_ALT_EXT.2.1. The evaluator shall restore network connectivity to the MDM Agent and verify that the alert generated while the TOE was disconnected is sent by the MDM Agent upon re-establishment of the connectivity.
1. Place the mobile device in airplane mode. 2. Attempt to cause the mobile device to become non-compliant by installing an application that is on the application deny list. 3. Turn off airplane mode on the mobile device. 4. Verify that an alert detecting a non-compliant device due to the presence of an application that is on the application deny list is transmitted to the MDM Server from the MDM Agent.	
Test Results:	The evaluator found that after the network connection was restored the alert was sent to the MDM server. – PASS
Execution Method:	Manual

008	[AGENTMOD]FAU_ALT_EXT.2/IOS – Agent Alerts – Test 1
Test Objective:	Test 1: The evaluator shall perform a policy update from the test environment MDM server. The evaluator shall verify the MDM Agent accepts the update, makes the configured changes, and reports the success of the policy update back to the MDM Server.
	1. Authenticate to the MDM Server console. 2. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile” 3. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted. 4. Click in the “Assigned Groups” field. 5. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices. 6. Click the “Restrictions” tab. 7. Uncheck “Allow use of camera”. 8. Click “Save & Publish”. 9. Confirm devices/users for the policy assignment, then click “Publish”. 10. Verify on the mobile device that the profile is received and enforced. 11. Navigate to “Devices” > “Profiles & Resources” > “Profiles”. 12. Locate a profile that was already assigned to a device and click on the “Edit” (pencil icon). 13. Click “Add Version”. 14. Click on the “Restrictions” tab. 15. Uncheck “Allow screen capture”. 16. Click on “Save”. 17. Click on “Save And Publish”. 18. Verify on the mobile device that the use of screen capture is disabled. 19. Verify via audit records that the mobile device reports that the policy update is successfully received.
Test Results:	The evaluator performed this test by first configuring a base policy (i.e. initial policy) on the MDM server and then assigned it to the mobile device. The evaluator confirmed the mobile device received this policy. Next, the evaluator configured a policy update to the base policy on the MDM server and then assigned it to the same mobile device. The evaluator confirmed that the mobile device received this policy, observed that the configuration defined in the policy update was accepted by the mobile device, and that the mobile device successfully reported the success of the policy update to the MDM server via audit records. – PASS
Execution Method:	Manual

009	[AGENTMOD]FAU_ALT_EXT.2/IOS – Agent Alerts – Test 2
Test Objective:	Test 2: The evaluator shall perform each of the actions listed in FAU_ALT_EXT.2.1 and verify that the alert does in fact reach the MDM Server.
	Successful application of policies to a mobile device: This is tested in [AGENTMOD]FAU_ALT_EXT.2/IOS – Test Case 008. ***Generating periodic reachability events:*** This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003I. This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003P.

Change in enrollment state:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 001I.
This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 001P.

Failure to install an application from the MAS Server:

1. Configure the MAS server to prevent the installation of an application.
2. Upload a test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click “Save and Assign”.
    f. Click “Add Assignment”, click in “Select Assignment Groups” box, and choose applicable Assignment Group of users/devices to assign application.
    g. Choose “On Demand” for “App Delivery Method”.
    h. Click “Add”.
    i. Click “Save and Publish”.
    j. Click “Publish”.
3. Verify that the MDM Agent notifies the MDM Server of the failure to install the application from the MAS Server.

Failure to update an application from the MAS Server:

1. Upload a valid initial version of the test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click “Save and Assign”.
    f. Click “Add Assignment”, click in “Select Assignment Groups” box, and choose applicable Assignment Group of users/devices to assign application.
    g. Choose “On Demand” for “App Delivery Method”.
    h. Click “Add”.
    i. Click “Save and Publish”.
    j. Click “Publish”.
2. Verify that the initial version of the test application installs successfully.
3. Configure the MAS server to prevent the installation of an application.
4. Upload the application update version of the test application to the MDM/MAS Server.
    a. Click “Add” from the top menu > “Internal Application”.
    b. Click “Upload” and specify the path to the application.
    c. Once the application is uploaded, click “Save”.
    d. Click “Continue”.
    e. On Application Summary screen verify information regarding chosen Application and click

“Save and Assign”.

   f. Verify the application assignment is correct.
   g. Click “Publish”.

5. Verify that the MDM Agent notifies the MDM Server of the failure to install the application update from the MAS Server.

Detection of apps on deny list, detection of apps not on allow list, required apps missing, unapproved model, unapproved operating system version:

This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003I.
This is tested in [MDMPP]FAU_ALT_EXT.1.1 – Test Case 003P.

Test Results:	The evaluator confirmed that all notifications were successfully received at the MDM server for each of the scenarios. – PASS
Execution Method:	Manual

010	[AGENTMOD]FAU_ALT_EXT.2/IOS – Agent Alerts – Test 3
Test Objective:	Test 3: The evaluator shall configure the MDM Agent to perform a network reachability test, both with and without such connectivity and ensure that results reflect each.

With Connectivity:

1. Launch the Hub MDM Agent on the mobile device.
2. Tap “This Device”.
3. Tap “Send Data” to perform a network reachability test to the MDM Server.
4. Navigate to “Devices” > “List View”.
5. Under the “Last Seen” column for the mobile device, ensure that the time value is near zero (or near the elapsed time since the device sample was sent from the mobile device).

Without Connectivity:

1. Join the mobile device to a Wi-Fi network that cannot reach the MDM Server.
2. Launch the Hub MDM Agent on the mobile device.
3. Tap “Send Data” to perform a network reachability test to the MDM Server.
4. Navigate to “Devices” > “List View”.
5. Under the “Last Seen” column for the mobile device, ensure that the time value is greater than the time value from when connectivity was established in the “with connectivity” subtest.
6. Restore connectivity between the mobile device and the MDM Server.
7. Verify that audit records are generated for the failure of the MDM agent sending the reachability alert to the MDM server.

Test Results:	The evaluator performed this test by first ensuring there was connectivity between the MDM server and the MDM agent and then initiated a device sync from the MDM agent. The evaluator observed that the "Last Seen" value was updated to a near zero value which was consistent with the expectation that the device had just communicated with the MDM server. The evaluator also verified that corresponding audit records were generated for this event. For the "without connectivity" portion of this test, the evaluator joined the mobile device to a wireless network that could not reach the MDM server, initated a device sync from the MDM agent and confirmed that the "Last Seen" value increased to a value greater than the previous value. Next, the evaluator restored connectivity to the MDM server on the mobile device and observed that audit records were generated for the failure of the MDM agent sending the reachability alert to the MDM server. – PASS
Execution Method:	Manual

011	[AGENTMOD]FAU_ALT_EXT.2/IOS – Agent Alerts
Test Objective:	Test 4: The evaluator shall remove network connectivity from the MDM Agent and generate an alert/event as defined in FAU_ALT_EXT.2.1. The evaluator shall restore network connectivity to the MDM Agent and verify that the alert generated while the TOE was disconnected is sent by the MDM Agent upon re-establishment of the connectivity.
1. Join the mobile device to a Wi-Fi network that cannot reach the MDM Server. 2. Attempt to cause the mobile device to become non-compliant by installing an application that is on the application deny list. 3. Restore connectivity between the mobile device and the MDM Server. 4. Verify that an alert detecting a non-compliant device due to the presence of an application that is on the application deny list is transmitted to the MDM Server from the MDM Agent.	
Test Results:	The evaluator found that after the network connection was restored the alert was sent to the MDM server. - PASS
Execution Method:	Manual

012	[MDMPP]FAU_GEN.1.1(1) – Audit Data Generation
Test Objective:	The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the provided table and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.
The TOE's ability to generate audit records for each of the events listed in Table 13 from the Security Target, including all administrative actions is tested in conjunction with the testing of the security mechanism directly.	

Test Results:	The TOE’s ability to generate audit records for each of the events listed in Table 13 from the Security Target, including all administrative actions is tested in conjunction with the testing of the security mechanism directly. – PASS
Execution Method:	Manual

013	[MDMPP]FAU_GEN.1.2(1) – Audit Data Generation
Test Objective:	When verifying the test results from FAU_GEN.1.1(1), the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected. The Auditable Events table includes optional, selection-based and objective requirements. The auditing of these requirements are only required if the requirement is included in the ST. (Refer to Table 13 in the Security Target for the Server Auditable Events table)
The TOE’s ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed in Table 13 from the Security Target, including all administrative actions is tested in conjunction with the testing of the security mechanism directly.	
Test Results:	The TOE’s ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed in Table 13 from the Security Target, including all administrative actions is tested in conjunction with the testing of the security mechanism directly. - PASS
Execution Method:	Manual

014	[MDMPP]FAU_GEN.1.1(2) – Audit Generation (MAS Server)
Test Objective:	The evaluator shall verify that when an application or update push fails, that the audit records generated match the format specified in the guidance and that the fields in each audit record have the proper entries.
Refer to [AGENTMOD]FAU_ALT_EXT.2/ANDROID - Test Case 005 **Failure to install an application from the MAS Server** **Failure to update an application from the MAS Server** **and** **[AGENTMOD]FAU_ALT_EXT.2/IOS - Test Case 009** **Failure to install an application from the MAS Server** **Failure to update an application from the MAS Server**	

Test Results:	Refer to [AGENTMOD]FAU_ALT_EXT.2/ANDROID - Test Case 005 Failure to install an application from the MAS Server Failure to update an application from the MAS Server and [AGENTMOD]FAU_ALT_EXT.2/IOS - Test Case 009 Failure to install an application from the MAS Server Failure to update an application from the MAS Server – PASS
Execution Method:	Manual

015	[MDMPP]FAU_GEN.1.2(2) – Audit Generation (MAS Server)
Test Objective:	When verifying the test results from FAU_GEN.1.1(2), the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.
The TOE's ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed from [MDMPP] FAU_GEN.1.1(2), is tested in conjunction with the testing of the security mechanism directly.	
Test Results:	The TOE's ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed from [MDMPP] FAU_GEN.1.1(2), is tested in conjunction with the testing of the security mechanism directly. - PASS
Execution Method:	Manual

016	[AGENTMOD]FAU_GEN.1(2) – Audit Data Generation
Test Objective:	The evaluator shall use the TOE to perform the auditable events defined in the Auditable Events table in FAU_GEN.1.1(2) and observe that accurate audit records are generated with contents and formatting consistent with those described in the TSS. Note that this testing can be accomplished in conjunction with the testing of the security mechanisms directly.
The TOE's ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed from [AGENTMOD] FAU_GEN.1.1(2), is tested in conjunction with the testing of the security mechanism directly.	
Test Results:	The TOE's ability to generate audit records, in the format specified per the AGD (with each field having the proper entries), for each of the events listed from [AGENTMOD] FAU_GEN.1.1(2), is tested in conjunction with the testing of the security mechanism directly. - PASS
Execution Method:	Manual

017	[MDMPP]FAU_NET_EXT.1.1 – Network Reachability Review
Test Objective:	For each MDM Agent/platform listed as supported in the ST: The evaluator shall configure the MDM Agent/platform to perform a network reachability test, both with and without such connectivity and shall ensure that by following the guidance, the evaluator can determine results that reflect both.
1. Ensure the mobile device is powered on and connected to the Internet/test network. 2. Authenticate to the MDM Console as the Administrator. 3. Navigate to “Devices” > “List View”. 4. Choose the specific device to query the connectivity status under the “General Info” column. 5. Choose “Query” from the top toolbar. 6. Confirm the “Query” on the confirmation page. 7. Navigate to “Devices > “List View”. 8. Refresh the List View section. 9. Verify the time value under the “Last Seen” column. 10. Power off the mobile device. 11. Repeat Steps 3-8. 12. Verify that the time value under the “Last Seen” column has increased.	
Test Results:	For each mobile device platform, the evaluator performed a device network reachability query from the MDM server with connectivity between the MDM server and MDM agent. The evaluator verified that the "Last Seen" value was updated on the MDM server for each mobile device. Next, the evaluator powered off each mobile device (in order to terminate connectivity between the MDM server and MDM agent). Once each device was powered off, the evaluator then performed another device query from the MDM server. The evaluator verified that the "Last Seen" value increased from the previous value. - PASS
Execution Method:	Manual

018	[MDMPP]FAU_SAR.1.2 – Audit Review
Test Objective:	The evaluator shall attempt to view the audit record as the authorized administrator and verify that the action succeeds. The evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide.
1. Navigate to the MDM Server Web Console. 2. Authenticate as the administrator. 3. Navigate to “Monitor” > “Reports & Analytics” > “Events” > “Device Events”. 4. Attempt to view a sample of audit records. 5. Navigate to “Monitor” > “Reports & Analytics” > “Events” > “Console Events”. 6. Attempt to view a sample of audit records. 7. Verify that the audit records match the format specified in the administrative guide. Note that auditable events are also able to be transmitted to a remote syslog server. The format for the syslog events is administrator configurable and can be viewed any number of ways, depending on the remote syslog configuration.	

Test Results:	The evaluator was able to successfully view audit records after authenticating to the MDM server web console as an authorized administrator. The evaluator confirmed that the audit records generated throughout testing conformed to the format specified in the administrative guidance documentation. NOTE: Auditable events are also able to be transmitted to a remote syslog server. The format for the syslog events is administrator configurable and can be viewed any number of ways, depending on the remote syslog configuration. – PASS
Execution Method:	Manual

019	[AGENTMOD]FAU_SEL.1(2) – Security Audit Event Selection
Test Objective:	Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.

Android:

1. Perform Steps 1 through 3 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003A to demonstrate that no audit/alert is generated (“failure of auditable security events”) for the presence of apps not on an allow list.
2. Perform Steps 4 through 5 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003A to demonstrate that an audit/alert is generated (“success of auditable security events”) for the presence of apps not on an allow list by enabling that audit/alert functionality.

iOS:

1. Perform Step 1 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003I to demonstrate that no audit/alert is generated (“failure of auditable security events”) for the presence of apps not on an allow list.
2. Perform Steps 2 through 4 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003I to demonstrate that an audit/alert is generated (“success of auditable security events”) for the presence of apps not on an allow list by enabling that audit/alert functionality.

iPadOS:

1. Perform Step 1 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003P to demonstrate that no audit/alert is generated (“failure of auditable security events”) for the presence of apps not on an allow list.
2. Perform Steps 2 through 4 in “Presence of apps not on an allow list” from FAU_ALT_EXT.1.1 – Test Case 003P to demonstrate that an audit/alert is generated (“success of auditable security events”) for the presence of apps not on an allow list by enabling that audit/alert functionality.

Test Results:	Android: The evaluator performed this test by executing steps 1 through 3 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003A, which demonstrated that no audit/alert was generated ("failure of auditable security events") for the presence of apps not on an allow list. The evaluator then executed steps 4 through 5 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003A, which demonstrated that an audit/alert was generated ("success of auditable security events") for the presence of apps not on an allow list by enabling that audit/alert functionality. iOS: The evaluator performed this test by executing steps 1 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003I, which demonstrated that no audit/alert was generated ("failure of auditable security events") for the presence of apps not on an allow list. The evaluator then executed steps 2 through 4 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003I, which demonstrated that an audit/alert was generated ("success of auditable security events") for the presence of apps not on an allow list by enabling that audit/alert functionality. iPadOS: The evaluator performed this test by executing steps 1 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003I, which demonstrated that no audit/alert was generated ("failure of auditable security events") for the presence of apps not on an allow list. The evaluator then executed steps 2 through 4 in "Presence of apps not on an allow list" from FAU_ALT_EXT.1.1 – Test Case 003I, which demonstrated that an audit/alert was generated ("success of auditable security events") for the presence of apps not on an allow list by enabling that audit/alert functionality. - PASS
Execution Method:	Manual

020	[AGENTMOD]FAU_SEL.1(2) – Security Audit Event Selection
Test Objective:	Test 2: [conditional]: If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.
N/A – The ST does not claim other attributes for this SFR.	
Test Results:	N/A
Execution Method:	N/A

021	[MDMPP]FAU_STG_EXT.1.1 – External Trail Storage
Test Objective:	Testing of the trusted channel mechanism will be performed as specified in the associated evaluation activities for the particular trusted channel mechanism. The evaluator shall perform the following test for this requirement: Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.
1. Begin capturing packets with Wireshark between the MDM Server and the remote audit server. 2. Perform some activity on the TOE that causes audit records to be transmitted to the remote audit server. 3. Stop capturing packets with Wireshark. 4. Examine the packet capture and verify that the communications are not transmitted in plaintext. 5. On the remote audit server, verify the records generated during Step 2 are received successfully. 6. Record the name and version of the software used on the audit server.	
Test Results:	The evaluator performed this test by first initiating a packet capture between the TOE (MDM server) and the remote audit (syslog) server. The evaluator then stimulated the TOE by performing several activities that generated audit records. The evaluator terminaed the packet capture. The evaluator reviewed the packet capture using Wireshark and confirmed that the data was unale to be viewed in the clear as it was encrypted using TLS. Additionally, the evaluator examined the audit records transferred from the TOE to the audit server and confirmed they were received successfully. Finally, the evaluator obtained and recorded the software name and version of the software running on the audit server: rsyslogd 8.1901.0 (aka 2019.01). - PASS
Execution Method:	Manual

4.3.2 Communication

022	[MDMPP]FCO_CPC_EXT.1.3 – Component Registration Channel Definition
Test Objective:	Test 1: The evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by an administrator for each of the non-equivalent TOE components that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE)
Refer to FIA_ENR_EXT.1.2/ANDROID – Test Case 029: “Limit enrollment to specific devices by IMEI” and FIA_ENR_EXT.1.2/IOS – Test Case 032: “Limit Enrollment to Specific Devices based on DEP identifier”. Test Case 029 verifies that an Android Hub Agent that has not been enabled cannot communicate with the UEM Server. Test Case 032 verifies that an iOS/iPadOS Hub Agent that has not been enabled cannot communicate with the UEM Server.	

Test Results:	Refer to FIA_ENR_EXT.1.2/ANDROID – Test Case 029: “Limit enrollment to specific devices by IMEI” and FIA_ENR_EXT.1.2/IOS – Test Case 032: “Limit Enrollment to Specific Devices based on DEP identifier”. Test Case 029 verifies that an Android Hub Agent that has not been enabled cannot communicate with the UEM Server. Test Case 032 verifies that an iOS/iPadOS Hub Agent that has not been enabled cannot communicate with the UEM Server. - PASS
Execution Method:	Manual

023	[MDMPP]FCO_CPC_EXT.1.3 – Component Registration Channel Definition – TD0594
Test Objective:	The evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication is possible but has not been explicitly enabled. Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.
Refer to FIA_ENR_EXT.1.1/ANDROID – Test Case 028 and FIA_ENR_EXT.1.1/IOS – Test Case 031. Test Case 028 verifies that once enabled the UEM Server and the Android Hub Agent can communicate. Test Case 031 verified that once enabled the UEM Server and the iOS/iPadOS Hub Agent can communicate.	
Test Results:	Refer to FIA_ENR_EXT.1.1/ANDROID – Test Case 028 and FIA_ENR_EXT.1.1/IOS – Test Case 031. Test Case 028 verifies that once enabled the UEM Server and the Android Hub Agent can communicate. Test Case 031 verified that once enabled the UEM Server and the iOS/iPadOS Hub Agent can communicate. – PASS
Execution Method:	Manual

024	[MDMPP]FCO_CPC_EXT.1.3 – Component Registration Channel Definition – TD0594
Test Objective:	The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component. In situations where one component acts as the "Gatekeeper" for all other components, the test would involve disabling the components in turn on the Gatekeeper and ensuring that the TOE no longer communicates with disabled components.
1. Unenroll and prevent the mobile device from reenrolling (disable re-enrollment) into MDM. **Android:** a. Authenticate to the MDM Server Console as the Administrator. b. Command the specified mobile device to unenroll from management by: i. Navigate to “Devices” > “List View”. ii. Choose the specific Device to execute Delete Device under the “General Info” column. iii. Choose “More Actions” from the top right-hand menu then “Delete Device” under the Admin heading. c. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Authentication”. d. Ensure “Devices Enrollment Mode” is set to “Registered Devices Only”. e. Navigate to “Devices” > “Lifecycle” > “Enrollment Status”.	

f. Remove the permitted IMEI corresponding to the device unenrolled in Step 1. **iOS:** a. Authenticate to the MDM Server Console as the Administrator. b. Command the specified mobile device to unenroll from management by: i. Navigate to “Devices” > “List View”. ii. Choose the specific Device to execute Device Wipe under the “General Info” column. iii. Choose “More Actions” from the top right-hand menu then “Enterprise Wipe” under the Management heading. c. Remove the device from the Apple Device Enrollment Portal (DEP) corresponding to the device unenrolled in Step 1. d. Attempt to re-enroll the device that was unenrolled during Step 1 and verify that enrollment is prevented.	
Test Results:	The evaluator separately disabled each TOE component in turn and ensured that the other TOE components didn't communicate with the disabled component, whether attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component. – PASS
Execution Method:	Manual

4.3.3 Cryptographic Support

All cryptographic services for the VMware Workspace ONE Unified Endpoint Management Version 2209 are provided by the underlying platforms for the TOE components, except for policy signing digital signature verification on Android devices which is validated by the Android Hub Agent’s implementation of OpenSSL.

- The UEM Server is installed on the Windows Server 2019 platform which uses Microsoft’s SymCrypt cryptographic implementation. These cryptographic modules’ certificates have been validated under Microsoft Windows Common Criteria Evaluation Microsoft Windows 10 version 1803 (April 2018 Update) Microsoft Windows Server version 1803 (April 2018 Update).

- The iOS Hub Agent uses Apple iOS/iPad platform’s CoreCrypto Module. This cryptographic module’s certificates have been validated under VID11146/VID11147.

- The Android Hub Agent uses the SCrypto and BoringSSL cryptographic modules to perform the cryptographic services to support all cryptographic functionality but the digital signature validation of policies requirements. These cryptographic modules’ certificates have been validated under VID11160.

Test cases for FCS_CKM.1, FCS_CKM.2, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), and FCS_RBG_EXT.1 are not included within this section. This is because the Test Assurance Activities have been satisfied through CAVS testing perform on the platforms’ cryptographic modules and the Android Hub Agent’s implementation of OpenSSL. Refer to the platforms’ Security Targets for their CAVP algorithm certificates.

The following table contains the CAVP algorithm certificates corresponding to the Android Hub Agent’s digital signature verification cryptographic functionality which is implemented by its OpenSSL version 2.0.16 cryptographic module.

	Algorithm	CAVP Cert. # (Android 9)
FCS_COP.1(2) – Hashing Algorithms	SHA-512 Digest sizes 512	A3270
FCS_COP.1(3) – Signature Algorithms	Elliptic Curve Digital Sig Algorithm (256 bits, NIST curve P-521)	A3270

<table>
<tr><td>025</td><td colspan="2">[MDMPP]FCS_CKM_EXT.4.2 – Cryptographic Key Destruction</td></tr>
<tr><td colspan="2"><b>Test Objective:</b></td><td>For each software and firmware key clearing situation the evaluator shall repeat the following tests. Note that at this time hardware-bound keys are explicitly excluded from testing.

Test 1: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate copies of keys that are persisted encrypted by the TOE:

1. Load the instrumented TOE build in a debugger.
2. Record the value of the key in the TOE subject to clearing.
3. Cause the TOE to perform a normal cryptographic processing with the key from #1.
4. Cause the TOE to clear the key.
5. Cause the TOE to stop the execution but not exit.
6. Cause the TOE to dump the entire memory footprint of the TOE into a binary file.
7. Search the content of the binary file created in #4 for instances of the known key value from #1.

The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise. The evaluator shall perform this test on all keys, including those persisted in encrypted form, to ensure intermediate copies are cleared.</td></tr>
<tr><td colspan="3"><b>N/A – No ATE testing activities as “invoke platform-provided functionality” is selected in the Security Target.</b></td></tr>
<tr><td colspan="2"><b>Test Results:</b></td><td>N/A</td></tr>
<tr><td colspan="2"><b>Execution Method:</b></td><td>N/A</td></tr>
</table>

026	[MDMPP]FCS_CKM_EXT.4.2 – Cryptographic Key Destruction
Test Objective:	For each software and firmware key clearing situation the evaluator shall repeat the following tests. Note that at this time hardware-bound keys are explicitly excluded from testing. Test 2: In cases where the TOE is implemented in firmware and operates in a limited operating environment that does not allow the use of debuggers, the evaluator shall utilize a simulator for the TOE on a general purpose operating system. The evaluator shall provide a rationale explaining the instrumentation of the simulated test environment and justifying the obtained test results.
N/A – No ATE testing activities as "invoke platform-provided functionality" is selected in the Security Target.	
Test Results:	N/A
Execution Method:	N/A

4.3.4 Identification and Authentication

027	[MDMPP]FIA_ENR_EXT.1.1/ANDROID – Enrollment of Mobile Device into Management – Test 1
Test Objective:	Test 1: The evaluator shall attempt to enroll a device without providing correct credentials. The evaluator shall verify that the device is not enrolled and that the described enrollment actions are not taken.
1. Power on the Android mobile device. 2. Join the mobile device to the test network. 3. Attempt to enroll the device using invalid credentials. 4. Enter the Username created in the Setup and an invalid password. 5. Navigate to the MDM Console Server > "Devices" > "List View". 6. Verify that the device is not listed. 7. Repeat Steps 1-6, except use invalid LDAP based credentials.	
Test Results:	The evaluator performed this test by attempting to enroll the mobile device to MDM using invalid local credentials. The evaluator observed that the enrollment was unsuccessful. The evaluator repeated the enrollment attempt using invalid AD/LDAP based credentials and observed that the enrollment attempt was unsuccessful. The evaluator also verified that audit records were generated for each failed enrollment attempt. – PASS
Execution Method:	Manual

028	[MDMPP]FIA_ENR_EXT.1.1/ANDROID – Enrollment of Mobile Device into Management – Test 2
Test Objective:	Test 2: The evaluator shall attempt to enroll the device providing correct credentials. The evaluator shall verify that the device is enrolled and that the described enrollment actions are taken.
1. Power on the Android mobile device. 2. Join the mobile device to the test network. 3. Attempt to enroll the device into MDM using valid credentials: a. Install the Android Hub Agent onto the mobile device via QR code enrollment method.	

   b. Specify the MDM server reference identifier: https://uem.cctl.company.com
   c. Specify the Group ID: “cc”

4. Enter the Username and Password created in the Setup.
5. After the profile is completely installed and configured, verify enrollment in MDM Console Server.
6. Navigate to the MDM Console Server > “Devices” > “List View”.
7. Verify that the enrolled device is listed.
8. Repeat Steps 1-7, except use LDAP based credentials.

Test Results:	The evaluator performed this test by attempting to enroll the mobile device to MDM using valid local credentials. The evaluator observed that the enrollment was successful. The evaluator then unenrolled the device from MDM. The evaluator repeated the enrollment attempt using valid AD/LDAP based credentials and observed that the enrollment attempt was successful. – PASS
Execution Method:	Manual

029	[MDMPP]FIA_ENR_EXT.1.2/ANDROID – Enrollment of Mobile Device into Management – Test 1
Test Objective:	For each type of policy selected, the evaluator shall perform the following: Test 1: The evaluator shall attempt to configure the MDM Server according to the administrative guidance in order to prevent enrollment. The evaluator shall verify that the user cannot enroll a device outside of the configured limitation. (For example, the evaluator may try to enroll a disallowed device, or may try to enroll additional devices beyond the number allowed.)

Limit enrollment to specific devices by IMEI:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Authentication”.
3. Ensure “Devices Enrollment Mode” is set to “Registered Devices Only”.
4. Navigate to “Devices” > “Lifecycle” > “Enrollment Status” > “ADD” > “Allow Devices”.
5. Specify the permitted IMEIs.
6. Specify “IMEI” for device attribute.
7. Attempt to enroll a mobile device with a non-permitted IMEI.
8. Verify that enrollment of the prohibited device is unsuccessful.
9. Remove enrollment restrictions that were created specifically for each subtest of this test.

Limit enrollment of devices by specific device models:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Restrictions” > “ADD POLICY”.
3. Specify the Enrollment Policy Name, Organization Group, Policy Type.
4. Check “Allowed Device Types – Limit enrollment to specific platforms, models or operating systems”.
5. Specify “Only allow listed device types (Allowlist).”
6. Click “Add Device Restriction”.
7. Specify the Platform to Android, Manufacturer is Samsung, Model is Galaxy S10e-SM-G970.
8. Specify the Device Limit per User to 10 and the operating system to Any.
9. Attempt to enroll a Samsung Galaxy S10 SM-G973 Android device into MDM.
10. Verify that enrollment of the prohibited device is unsuccessful.

11. Remove enrollment restrictions that were created specifically for each subtest of this test.

Limit enrollment of devices by the number of devices:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Restrictions” > “ADD POLICY”.
3. Specify the Enrollment Policy Name, Organization Group, Policy Type.
4. Check “Device Limit per User”.
5. Specify “Maximum Devices Per User” to “1”.
6. Enroll one device into MDM.
7. Attempt to enroll a second device into MDM using the same user account used in Step 6.
8. Verify that enrollment of the second device fails.
9. Remove enrollment restrictions that were created specifically for each subtest of this test.

Limit enrollment of devices by serial number:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Authentication”.
3. Ensure “Devices Enrollment Mode” is set to “Registered Devices Only”.
4. Navigate to “Devices” > “Lifecycle” > “Enrollment Status” > “ADD” > “Allow Devices”.
5. Specify the permitted serial numbers.
6. Specify “Serial Number” for device attribute.
7. Attempt to enroll a mobile device with a non-permitted serial number.
8. Verify that enrollment of the prohibited device is unsuccessful.
9. Remove enrollment restrictions that were created specifically for each subtest of this test.

Limit enrollment of devices by manufacturer:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Restrictions” > “ADD POLICY”.
3. Specify the Enrollment Policy Name, Organization Group, Policy Type.
4. Check “Allowed Device Types – Limit enrollment to specific platforms, models or operating systems”.
5. Specify “Only allow listed device types (Allowlist).”
6. Click “Add Device Restriction”.
7. Specify the Platform to Android, Manufacturer is HTC, Model is Any.
8. Specify the Device Limit per User to 10 and the operating system to Any.
9. Attempt to enroll a Samsung Galaxy Android device into MDM.
10. Verify that enrollment of the prohibited device is unsuccessful.
11. Remove enrollment restrictions that were created specifically for each subtest of this test.

Limit enrollment of devices by operating system:

1. Authenticate to the MDM Server Console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Restrictions” > “ADD POLICY”.

3. Specify the Enrollment Policy Name, Organization Group, Policy Type.
4. Check “Allowed Device Types – Limit enrollment to specific platforms, models or operating systems”.
5. Specify “Only allow listed device types (Allowlist).”
6. Click “Add Device Restriction”.
7. Specify the Platform to Android, Manufacturer is Samsung, Model is Any.
8. Specify the Device Limit per User to 10 and the operating system to Android 8.1.0.
9. Attempt to enroll a Samsung Galaxy Android 9 device into MDM.
10. Verify that enrollment of the prohibited device is unsuccessful.
11. Remove enrollment restrictions that were created specifically for each subtest of this test.

Test Results:	The evaluator performed this test by configuring the TOE (MDM server), for each type of policy selected in the ST for this SFR, to prevent enrollment. Enrollment was first restricted to only permitted IMEIs. The evaluator attempted to enroll a device with a non-permitted IMEI and confirmed that it failed to enroll. The evaluator then removed the enrollment restriction. Second, enrollment was restricted to only permitted device models. The evaluator attempted to enroll a device model that was not permitted and confirmed that it failed to enroll. The evaluator then removed the enrollment restriction. Third, enrollment was limited to a maximum number of devices per user with a value of "1". The evaluator successfully enrolled one device and then attempted to enroll an additional device using the same user. That enrollment failed. The evaluator then removed the enrollment restriction. Fourth, enrollment was restricted to only permitted serial numbers. The evaluator attempted to enroll a device with a non-permitted serial number. That enrollment also failed. The evaluator then removed the enrollment restriction. Fifth, enrollment was restricted to only permitted device manufacturers. The evaluator attempted to enroll a non-permitted device manufacturer and confirmed that the enrollment failed. The evaluator then removed the enrollment restriction. Finally, enrollment was restricted to only permitted operation system versions. The evaluator attempted to enroll a device with a non-permitted operating system version and confirmed that it failed to enroll. – PASS
Execution Method:	Manual

030	[MDMPP]FIA_ENR_EXT.1.1/IOS – Enrollment of Mobile Device into Management – Test 1
Test Objective:	Test 1: The evaluator shall attempt to enroll a device without providing correct credentials. The evaluator shall verify that the device is not enrolled and that the described enrollment actions are not taken.

DEP Based Enrollment:

1. Power on the Apple mobile device.
2. Join the mobile device to the Internet.
3. Attempt to enroll the device using invalid credentials.
4. Enter the Username created in the Setup and an invalid password.
5. Navigate to the MDM Console Server > “Devices” > “List View”.
6. Verify that the device is not listed.
7. Repeat Steps 1-6, except use LDAP based credentials.

Test Results:	The evaluator performed this test by attempting to enroll the mobile device to MDM using invalid local credentials. The evaluator observed that the enrollment was unsuccessful. The evaluator repeated the enrollment attempt using invalid AD/LDAP based credentials and observed that the enrollment attempt was unsuccessful. The evaluator also verified that audit records were generated for each failed enrollment attempt. – PASS
Execution Method:	Manual

031	[MDMPP]FIA_ENR_EXT.1.1/IOS – Enrollment of Mobile Device into Management – Test 2
Test Objective:	Test 2: The evaluator shall attempt to enroll the device providing correct credentials. The evaluator shall verify that the device is enrolled and that the described enrollment actions are taken.

DEP Based Enrollment:

1. Power on the Apple mobile device.
2. Join the mobile device to the Internet.
3. Attempt to enroll the device using valid credentials.
4. Enter the Username and Password created in the Setup.
5. After the profile is completely installed and configured, verify enrollment in MDM Console Server.
6. Navigate to the MDM Console Server > “Devices” > “List View”.
7. Verify that the enrolled device is listed.
8. Repeat Steps 1-7, except use LDAP based credentials.

Test Results:	The evaluator performed this test by attempting to enroll the mobile device to MDM using valid local credentials. The evaluator observed that the enrollment was successful. The evaluator then unenrolled the device from MDM. The evaluator repeated the enrollment attempt using valid AD/LDAP based credentials and observed that the enrollment attempt was successful. - PASS
Execution Method:	Manual

032	[MDMPP]FIA_ENR_EXT.1.2/IOS – Enrollment of Mobile Device into Management – Test 1
Test Objective:	For each type of policy selected, the evaluator shall perform the following: Test 1: The evaluator shall attempt to configure the MDM Server according to the administrative guidance in order to prevent enrollment. The evaluator shall verify that the user cannot enroll a device outside of the configured limitation. (For example, the evaluator may try to enroll a disallowed device, or may try to enroll additional devices beyond the number allowed.)

Limit Enrollment to Specific Devices based on DEP identifier:

1. Authenticate to the MDM Server console as the Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment”.
3. Specify “Registered Devices Only” for “Devices Enrollment Mode”.
4. Follow the Test Steps in “[MDMPP]FIA_ENR_EXT.1.1/IOS – Enrollment of Mobile Device into Management – Test 2 (Test Case 031)” using an Apple device not registered for enrollment.

<table>
<tr><td><b>Test Results:</b></td><td>The evaluator performed this test by configuring the TOE (MDM server), for each type of policy selected in the ST for this SFR (i.e. DEP identifier), to prevent enrollment. The evaluator attempted to enroll a device with a non-permitted DEP identifier and confirmed that the enrollment failed. – PASS</td></tr>
<tr><td><b>Execution Method:</b></td><td>Manual</td></tr>
</table>

<table>
<tr><th>033A</th><th>[AGENTMOD]FIA_ENR_EXT.2 – Agent Enrollment of Mobile Device into Management</th></tr>
<tr><td><b>Test Objective:</b></td><td>The evaluator shall follow the operational guidance to establish the reference identifier of the MDM server on the MDM Agent and in conjunction with other evaluation activities verify that the MDM Agent can connect to the MDM Server and validate the MDM Server’s certificate.</td></tr>
<tr><td colspan="2">1. Enroll an Android mobile device into MDM.
2. Notate the fully qualified domain name (FQDN) of the MDM Server.
3. On the Android device, launch the MDM Agent.
4. Tap “This Device”.
5. Tap “Enrollment”.
6. Observe that the “Enrolled Server” is the reference identifier.</td></tr>
<tr><td><b>Test Results:</b></td><td>The evaluator queried the mobile device to establish the reference identifier of the MDM server on the MDM agent. In conjunction with other evaluation activities (FIA_X509_EXT.1 and FPT_ITT.1(2)), the evaluator confirmed that the MDM agent successfully connected to the MDM server and validated the MDM server's certificate. The evaluator validated that the reference identified matched that of the MDM server and the MDM server certificate. – PASS</td></tr>
<tr><td><b>Execution Method:</b></td><td>Manual</td></tr>
</table>

<table>
<tr><th>033I</th><th>[AGENTMOD]FIA_ENR_EXT.2 – Agent Enrollment of Mobile Device into Management</th></tr>
<tr><td><b>Test Objective:</b></td><td>The evaluator shall follow the operational guidance to establish the reference identifier of the MDM server on the MDM Agent and in conjunction with other evaluation activities verify that the MDM Agent can connect to the MDM Server and validate the MDM Server’s certificate.</td></tr>
<tr><td colspan="2"><b>iOS Platform MDM Agent:</b>

1. Follow the enrollment steps in “[MDMPP]FIA_ENR_EXT.1.1 – Enrollment of Mobile Device into Management – Test 2” to enroll a mobile device into MDM.
2. Notate the fully qualified domain name (FQDN) of the MDM Server.
3. On the iOS device, tap “Settings” > “General” > “Device Management” > “Device Manager”.
4. Tap “More Details”.
5. Tap “MDM Settings”.
6. Observe that the “Server URL” is the reference identifier.

<b>TOE (Hub) MDM Agent:</b>

7. Launch the Hub MDM Agent on the mobile device.
8. Tap “This Device” > “Enrollment”.
9. Observe that the “Server” is the reference identifier URL.</td></tr>
</table>

Test Results:	The evaluator queried the mobile device to establish the reference identifier of the MDM server on the MDM agent. In conjunction with other evaluation activities (FIA_X509_EXT.1 and FPT_ITT.1(2)), the evaluator confirmed that the MDM agent successfully connected to the MDM server and validated the MDM server's certificate. The evaluator validated that the reference identified matched that of the MDM server and the MDM server certificate. – PASS
Execution Method:	Manual

033P	[AGENTMOD]FIA_ENR_EXT.2 – Agent Enrollment of Mobile Device into Management
Test Objective:	The evaluator shall follow the operational guidance to establish the reference identifier of the MDM server on the MDM Agent and in conjunction with other evaluation activities verify that the MDM Agent can connect to the MDM Server and validate the MDM Server's certificate.

iPadOS Platform MDM Agent:

1. Follow the enrollment steps in "[MDMPP]FIA_ENR_EXT.1.1 – Enrollment of Mobile Device into Management – Test 2" to enroll a mobile device into MDM.
2. Notate the fully qualified domain name (FQDN) of the MDM Server.
3. On the iOS device, tap "Settings" > "General" > "Device Management" > "Device Manager".
4. Tap "More Details".
5. Tap "MDM Settings".
6. Observe that the "Server URL" is the reference identifier.

TOE (Hub) MDM Agent:

7. Launch the Hub MDM Agent on the mobile device.
8. Tap "This Device" > "Enrollment".
9. Observe that the "Server" is the reference identifier URL.

Test Results:	The evaluator queried the mobile device to establish the reference identifier of the MDM server on the MDM agent. In conjunction with other evaluation activities (FIA_X509_EXT.1 and FPT_ITT.1(2)), the evaluator confirmed that the MDM agent successfully connected to the MDM server and validated the MDM server's certificate. The evaluator validated that the reference identified matched that of the MDM server and the MDM server certificate. – PASS
Execution Method:	Manual

034	[MDMPP]FIA_UAU.1.2 – Timing of Authentication – Test 1
Test Objective:	The evaluator shall perform the following tests: Test 1: The evaluator shall attempt to perform the prohibited actions before authentication. The evaluator shall verify the actions cannot be performed.

MDM Server Console:

1. Attempt to navigate to a protected web resource within the MDM Server console prior to authentication:

   Navigate to:
   https://uem.cctl.company.com/AirWatch/#/AirWatch/Settings/AndroidServiceApplications

2. Verify that access to the protected resource is denied.

MDM Self-Service Portal:

1. Attempt to navigate to a protected web resource within the MDM Self-Service Portal prior to authentication:

   Navigate to:
   https://uem.cctl.company.com/MyDevice/Landing/#/MyDevice/Device/Actions/Advanced/149

2. Verify that access to the protected resource is denied.

Test Results:	The evaluator performed this test by attempting to access resources on the TOE (MDM server web console and MDM server self-service portal) that require authentication by an authorized Security Administrator, prior to authentication. Access to the protected resources was denied. – PASS
Execution Method:	Manual

035	[MDMPP]FIA_UAU.1.2 – Timing of Authentication – Test 2
Test Objective:	The evaluator shall perform the following tests: Test 2: The evaluator shall attempt to perform the prohibited actions after authentication. The evaluator shall verify the actions can be performed.

MDM Server Console:

1. Authenticate to the AirWatch Console Server.
2. Attempt to navigate to a protected web resource within the AirWatch Console Server:

   https://uem.cctl.company.com/AirWatch/#/AirWatch/Settings/AndroidServiceApplications

3. Verify that access to the protected resource is granted.

MDM Self-Service Portal:

1. Authenticate to the MDM Self-Service Portal.
2. Attempt to navigate to a protected web resource within the MDM Self-Service Portal:

   https://uem.cctl.company.com/MyDevice/Landing/#/MyDevice/Device/Actions/Advanced/149

3. Verify that access to the protected resource is granted.

Test Results:	The evaluator performed this test by attempting to access resources on the TOE (MDM server web console and MDM server self-service portal) that require authentication by an authorized Security Administrator, after authentication. Access to the protected resources was granted. – PASS
Execution Method:	Manual

036	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 1 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn: • by establishing a certificate path in which one of the issuing certificates is not a CA certificate, • by omitting the basicConstraints field in one of the issuing certificates, • by setting the basicConstraints field in an issuing certificate to have CA=False, • by omitting the CA signing bit of the key usage field in an issuing certificate, and • by setting the path length field of a valid CA field to a value strictly less than the certificate path. The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

Valid / invalid certificate path

1. Ensure the root CA certificate is installed in the mobile device trust store.
2. From the MDM Server, configure a policy and transmit it to the mobile device.
3. Verify the mobile device received the configured policy and is accepted.
4. Remove the policy from the mobile device.
5. Remove the root CA certificate from the mobile device trust store.
6. Repeat Step 2.
7. Verify the mobile device rejected the configured policy.

Not a CA certificate

Refer to FIA_X509_EXT.1.2(1) – Test 1 and 2 (Test Case 044, 045)

Omitting basicConstraints

Refer to FIA_X509_EXT.1.2(1) – Test 1 (Test Case 044)

Setting CA=FALSE in basicConstraints

Refer to FIA_X509_EXT.1.2(1) – Test 2 (Test Case 045)

Omitting the CA signing bit

1. Load the Root CA that validates the MDM Server's node certificate signed by a CA without the CA signing bit into the MDM Agent's trusted CA database.
2. Send a signed policy from the MDM Server to the MDM Agent.
3. Verify that the MDM Agent failed to successfully validate the certificate that signed the policy from the MDM Server and that the policy is not applied.

CA path length value strictly less than certificate path

1. Load the Root CA that validates the MDM Server's node certificate signed by an intermediate CA whose issuer intermediate CA path length value is strictly less than the certificate path into the MDM Agent's trusted CA database.
2. Send a signed policy from the MDM Server to the MDM Agent.
3. Verify that the MDM Agent failed to successfully validate the certificate that signed the policy from the MDM Server and that the policy is not applied.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. **Valid / invalid certificate path** For the MDM Agent (Android), the evaluator performed this test by first ensuring that the root CA certificate that completes the certification path was present in the mobile device CA trust store. The evaluator then configured a policy on the MDM server and transmitted it to the mobile device. The evaluator confirmed that the MDM agent accepted the policy. **Not a CA certificate** The evaluator then removed the policy from the mobile device and removed the root CA certificate from the mobile device CA trust store. The evaluator then configured a new policy and sent it to the mobile device. The evaluator confirmed that the MDM agent rejected that policy and that it generated audit records for the failure to validate the policy signing certificate due to it being an untrusted certificate. **Omitting basicConstraints** The evaluator then removed the policy from the mobile device and installed a certificate chain where the signing certificate had the basicConstraints missing. The evaluator confirmed that the MDM agent rejected that policy and that it generated audit records for the failure to validate the policy signing certificate due to it being an untrusted certificate. **Setting CA=FALSE in basicConstraints** The evaluator then removed the policy from the mobile device and installed a certificate chain where the signing certificate had the CA=FALSE in basicConstraints. The evaluator confirmed that the MDM agent rejected that policy and that it generated audit records for the failure to validate the policy signing certificate due to it being an untrusted certificate. **Omitting the CA signing bit** The evaluator then removed the policy from the mobile device and installed a certificate chain where the signing certificate had the CA signing bit omitted. The evaluator confirmed that the MDM agent rejected that policy and that it generated audit records for the failure to validate the policy signing certificate due to it being an untrusted certificate. **CA path length value strictly less than certificate path** The evaluator then removed the policy from the mobile device and installed a certificate chain where the CA path length value was set to less than the certificate path. The evaluator confirmed that the MDM agent rejected that policy and that it generated audit records for the failure to validate the policy signing certificate due to it being an untrusted certificate. - PASS
Execution Method:	Manual

037	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 2 – TD0641

Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the test machine, launch the debugger and configure it to set a breakpoint (https://uem.cctl.company.com/deviceservices/policysigningcertificate?requestType=x5c).
2. From the MDM Server, configure a policy and transmit it to the mobile device.
3. When the breakpoint is hit, replace the certificate data with the expired certificate data.
4. Verify the mobile device rejected the configured policy.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by configuring a policy on the MDM server and transmitted it to the mobile device. The evaluator then replaced the policy signing certificate with an expired one while in transit to the MDM Agent. The evaluator confirmed that the mobile device rejected the policy and that it generated audit records for the failure to validate the X.509 certificate due to it being expired. - PASS
Execution Method:	Manual

038	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 3 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates--conditional on whether CRL, OCSP, OCSP stapling, or certificate status lookup is selected; if multiple methods are selected, then a test shall be performed for each method. The evaluator shall test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). If OCSP stapling per RFC 6066 is the only supported revocation method, testing revocation of the intermediate CA certificate is omitted. For FIA_X509_EXT.1.1(2) if included, if "no revocation method" is selected, this test is omitted. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails, if "internal lookup of TOE-managed certificate status" is selected, then the evaluator shall follow AGD guidance to report the certificate as invalid.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as “invoke platform-provided functionality” is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the MDM Server, configure a policy and transmit it to the mobile device.
2. Verify the mobile device received the configured policy and is accepted.
3. Revoke the Policy Signing Certificate.
4. Begin capturing packets between the MDM Agent and the OCSP responder server.
5. From the MDM Server, configure a policy and transmit it to the mobile device.
6. Stop capturing packets between the MDM Agent and the OCSP responder server.
7. Verify the mobile device rejected the configured policy.
8. Un-revoke the Policy Signing Certificate and revoke the Intermediate01 CA certificate.
9. From the MDM Server, configure a policy and transmit it to the mobile device.
10. Verify the mobile device rejected the configured policy.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by first ensuring that a valid, non-revoked policy signing certificate was loaded on the MDM server. Then, the evaluator configured a policy on the MDM server and transmitted it to the mobile device. The evaluator confirmed that an OCSP check was performed, and that the certificate and its chain were successfully validated by the TOE, and the policy was successfully received on the mobile device. The evaluator then revoked the node certificate (policy signing certificate), loaded this certificate and its complete issuer chain on the MDM server, then configured a policy, transmitted it to the mobile device, confirmed an OCSP check was performed, that the node/leaf/policy signing certificate was deemed as revoked, and the policy was not successfully received on the mobile device. Finally, the evaluator unrevoked all certificates, revoked the intermediate 01 CA certificate (the certificate that is issued by the root CA certificate and the issuer of the intermediate 02 CA certificate, of which is the issuer of the node/leaf/policy signing certificate), loaded this certificate and its complete chain on the MDM server, configured a policy, transmitted it to the mobile device, confirmed an OCSP check was performed, that the intermediate 01 CA certificate was deemed as revoked, and the policy was not successfully received on the mobile device. The evaluator observed that both the MDM server and MDM agent reported no indication of a successfully installed policy in all instances (except for the case where none of the certificates were revoked). Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

039	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 4 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 4: [conditional] If OCSP option is selected, the evaluator shall send the TOE an OCSP response signed by a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall cause a CA to sign a CRL with a certificate that has a key usage extension but does not have the cRLsign key usage bit set, and verify that validation of the CRL fails. If certificate status lookup is selected, this test is omitted. For FIA_X509_EXT.1.1(2) if included, if "no revocation method" is selected, this test is omitted.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. Begin capturing packets between the MDM Agent and the OCSP responder server.
2. From the MDM Server, configure a policy and transmit it to the mobile device.

3. Verify that the OCSP response certificate does not contain the OCSP signing purpose.
4. Stop capturing packets between the MDM Agent and the OCSP responder server.
5. Verify the mobile device rejected the configured policy.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by first ensuring that the OCSP responder was configured to sign OCSP responses using a certificate that lacked the OCSP signing purpose. Then, the evaluator began capturing packets between the MDM agent and the OCSP responder. Next, the evaluator configured a policy on the MDM server and transmitted it to the mobile device. The evaluator terminated the packet capture. The evaluator confirmed that the mobile device rejected the policy and generated audit records for the failure to validate the X.509 certificate due to it being signed by a certificate that lacked the OCSP signing purpose. Finally, the evaluator examined the packet capture to confirm that the transmitted OCSP response lacked the OCSP signing purpose. - PASS
Execution Method:	Manual

040	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 5 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the test machine, launch the debugger and configure it to set a breakpoint (https://uem.cctl.company.com/deviceservices/policysigningcertificate?requestType=x5c) so that the policy signing certificate is modified such that the eighth byte of the certificate is modified from "0xD4" to "0x41".
2. From the MDM Server, configure a policy and transmit it to the mobile device.
3. Launch the MDM Agent on the mobile device.
    a. Press the "Sync device" button.
4. Verify the mobile device rejected the configured policy due to a certificate parse error.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by taking a legitimate policy signing certificate and modifying its binary structure so that the eighth byte of the certificate was changed from 0xD4 to 0x41. Then, the evaluator configured a policy on the MDM server, configured the mobile device to route its traffic between itself and the MDM server via a HTTP proxy. A breakpoint was set on "deviceservices/policysigningcertificate?requestType=x5c" so that when the policy was transmitted to the mobile device, the man-in-the-middle HTTP proxy halted the transmission of traffic from the MDM server to the MDM agent, which was then intercepted by the evaluator. The evaluator inspected this HTTP response for the base64 encoded certificate data and replaced the policy signing certificate portion with the base64 encoded modified certificate data. The evaluator then resumed the transmission of the intercepted traffic by releasing the breakpoint with the modification, observed that the traffic was transmitted and received by the mobile device, and that both the MDM server and MDM agent reported no indication of a successfully installed policy. Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

041	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 6 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the test machine, launch the debugger and configure it to set a breakpoint (https://uem.cctl.company.com/deviceservices/policysigningcertificate?requestType=x5c) so that the policy signing certificate is modified such that the last byte of the certificate is modified from "0xF8" to "0x41".
2. From the MDM Server, configure a policy and transmit it to the mobile device.
3. Launch the MDM Agent on the mobile device.
    a. Press the "Sync device" button.
4. Verify the mobile device rejected the configured policy due to a certificate parse error.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by taking a legitimate policy signing certificate and modifying its binary structure so that the last byte of the certificate was changed from 0xF8 to 0x41. Then, the evaluator configured a policy on the MDM server, configured the mobile device to route its traffic between itself and the MDM server via a HTTP proxy. A breakpoint was set on "deviceservices/policysigningcertificate?requestType=x5c" so that when the policy was transmitted to the mobile device, the man-in-the-middle HTTP proxy halted the transmission of traffic from the MDM server to the MDM agent, which was then intercepted by the evaluator. The evaluator inspected this HTTP response for the base64 encoded certificate data and replaced the policy signing certificate portion with the base64 encoded modified certificate data. The evaluator then resumed the transmission of the intercepted traffic by releasing the breakpoint with the modification, observed that the traffic was transmitted and received by the mobile device, and that both the MDM server and MDM agent reported no indication of a successfully installed policy. Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

042	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 7 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

MDM Server/iOS:

This test assurance activity is met by the underlying platform as “invoke platform-provided functionality” is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the test machine, launch the debugger and configure it to set a breakpoint (https://uem.cctl.company.com/deviceservices/policysigningcertificate?requestType=x5c) so that the policy signing certificate is modified such that the public key ((offset 332 bytes) from 0x9F to 0x41) of the certificate is modified.
2. From the MDM Server, configure a policy and transmit it to the mobile device.
3. Launch the MDM Agent on the mobile device.
    a. Press the “Sync device” button.
4. Verify the mobile device rejected the configured policy due to a certificate parse error.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by taking a legitimate policy signing certificate and modifying its binary structure so that the byte in position 32 of the certificate's public key was changed from 0x9F to 0x41. Then, the evaluator configured a policy on the MDM server, configured the mobile device to route its traffic between itself and the MDM server via a HTTP proxy. A breakpoint was set on "deviceservices/policysigningcertificate?requestType=x5c" so that when the policy was transmitted to the mobile device, the man-in-the-middle HTTP proxy halted the transmission of traffic from the MDM server to the MDM agent, which was then intercepted by the evaluator. The evaluator inspected this HTTP response for the base64 encoded certificate data and replaced the policy signing certificate portion with the base64 encoded modified certificate data. The evaluator then resumed the transmission of the intercepted traffic by releasing the breakpoint with the modification, observed that the traffic was transmitted and received by the mobile device, and that both the MDM server and MDM agent reported no indication of a successfully installed policy. Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

043	[MDMPP]FIA_X509_EXT.1.1(1) – X.509 Certificate Validation – Test 8 – TD0641
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. Test 8a: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. Test 8b: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as “invoke platform-provided functionality” is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

8a

1. Create an EC leaf certificate (“leaf”), two EC intermediate CA certificates (“int CA 02” and “int CA 01”), and an EC root CA certificate (“root CA”), such that they are all chained up to the EC root CA certificate: leaf → int CA 02 → int CA 01 → root CA.

2. Install the “root CA” certificate created in Step 1 into the MDM Agent’s platform trust store such that it is designated as a trust anchor.

3. Install the “leaf”, “int CA 02”, and “int CA 01” onto the MDM Server such that the leaf certificate is used to sign a policy configured on the MDM Server and that it and the intermediate certificates are presented to the MDM Agent when a policy is transmitted from the MDM Server to the MDM Agent.

4. Configure and transmit a policy from the MDM Server to the MDM Agent.

5. Verify that the MDM Agent accepts the signed policy from the MDM Server.

8b

1. Regenerate “int CA 01” with a modified public key information where the EC parameters use an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate, hereafter referred to as: “int CA 01 explicit”. Ensure that “int CA 01 explicit” is signed by “root CA” that was created in Step 1, with no other changes. Generate a new leaf certificate: (leaf → int CA 02 → int CA 01 explicit → root CA)

    a. Execute the following command to generate the explicit parameter version of the key generated from using a named curve:

    ```
    openssl ec -in <namedCurve.key> -param_enc explicit -out <explicit.key>
    ```

2. Install the “leaf → int CA 02 → int CA 01 explicit” chain onto the MDM Server such that the leaf certificate is used to sign a policy configured on the MDM Server and that it and the intermediate certificates are presented to the MDM Agent when a policy is transmitted from the MDM Server to the MDM Agent.

3. Configure and transmit a policy from the MDM Server to the MDM Agent.

4. Verify that the MDM Agent rejects the signed policy from the MDM Server.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by first creating an EC chain of certificates such that they are all chained up to the EC root CA certificate: leaf -> int CA 02 -> int CA 01 -> root CA. The evaluator then ensured that this certificate chain was presented to the MDM agent (Hub agent/TOE) on the mobile device as part of a policy transmission from the MDM server to the MDM agent. The evaluator confirmed that the MDM agent successfully received this policy. This was confirmed also by the MDM server reporting a successful policy installation. For the second part of this test, the evaluator regenerated an intermediate 01 CA certificate with a modified public key information where the EC parameters use an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate. This certificate was signed by the same common root CA certificate that was used in the first part of this test, with no other changes. The intermediate 02 CA certificate was generated and issued by this explicit format version of the intermediate 01 CA certificate, and the leaf/node/policy signing certificate was generated and issued by the intermediate 02 CA certificate. This certificate chain was presented to teh MDM agent (Hub agent/TOE) on the mobile device as part of a policy transmission from the MDM server to the MDM agent. The evaluator confirmed that the MDM agent did not successfully received this policy and that the certificate validation failed. This was confirmed also by the MDM server not reporting a successful policy installation. Additionally, audit records were generated that confirmed the failed validation of the certificate with the reason. - PASS
Execution Method:	Manual

044	[MDMPP]FIA_X509_EXT.1.2(1) – X.509 Certificate Validation – Test 1
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. Test 1: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate does not contain the basicConstraints extension. The validation of the certificate path fails.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the MDM Server, configure a policy and transmit it to the mobile device.
2. Verify the mobile device rejected the configured policy due to invalid CA certificate (basicConstraints missing) error.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by creating a policy signing certificate that was signed by an intermediate CA certificate whose issuer lacked the basicConstraints extension. The evaluator loaded this complete certificate chain, including the root CA, onto the MDM server. The evaluator then configured a policy on the MDM server, transmitted it to the mobile device, and observed that the mobile device did not install the policy, and that the MDM server did not report a successful policy installation. Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

045	[MDMPP]FIA_X509_EXT.1.2(1) – X.509 Certificate Validation – Test 2
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. Test 2: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the MDM Server, configure a policy and transmit it to the mobile device.
2. Verify the mobile device rejected the configured policy due to invalid CA certificate (cA flag not set) error.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by creating a policy signing certificate that was signed by an intermediate CA certificate whose issuer contained a CA flag with its value set to FALSE in the basicConstraints extension. The evaluator loaded this complete certificate chain, including the root CA, onto the MDM server. The evaluator then configured a policy on the MDM server, transmitted it to the mobile device, and observed that the mobile device did not install the policy, and that the MDM server did not report a successful policy installation. Audit records also confirm the intended outcome of this test, including the reason for the certificate validation failure. - PASS
Execution Method:	Manual

046	[MDMPP]FIA_X509_EXT.1.2(1) – X.509 Certificate Validation – Test 3
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. Test 3: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.

MDM Server/iOS:

This test assurance activity is met by the underlying platform as "invoke platform-provided functionality" is specified for the MDM Server and iOS TOE components in the Security Target. Therefore, this test assurance activity does not apply for these TOE components.

MDM Agent (Android):

1. From the MDM Server, configure a policy and transmit it to the mobile device.
2. Verify the mobile device accepted the configured policy.

Test Results:	For the MDM Server/iOS platforms, the evaluator determined that the test assurance activity was met by the underlying platform as "invoke platform-provided functionality" is specified for these TOE components in the Security Target. For the MDM Agent (Android), the evaluator performed this test by creating a policy signing certificate that was signed by an intermediate CA certificate whose issuer contained a CA flag with its value set to TRUE in the basicConstraints extension. The evaluator loaded this complete certificate chain, including the root CA, onto the MDM server. The evaluator then configured a policy on the MDM server, transmitted it to the mobile device, and observed that the mobile device installed the policy, and that the MDM server reported a successful policy installation. Audit records also confirm the intended outcome of this test. - PASS
Execution Method:	Manual

047	[MDMPP]FIA_X509_EXT.2.2 – X.509 Certificate Validation – Test 1
Test Objective:	The evaluator shall perform the following test for each trusted channel: Test 1: The evaluator shall demonstrate use of a valid certificate requiring certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

MDM Server to Environmental Entity (e.g. syslog, AD/LDAP) (TLS):

1. Begin capturing packets from the MDM server.
2. Perform an action that causes the MDM server to initiate a TLS connection to the environmental entity.
3. Verify the connection from the MDM server to the environmental entity is successful.
4. Stop capturing packets from the OCSP responder server.
5. Stop capturing packets from the MDM server.

6. Disconnect the connection between the MDM Server and the OCSP responder.

7. Begin capturing packets from the MDM server.
8. Perform an action that causes the MDM server to initiate a TLS connection to the environmental entity.
9. Verify the connection from the MDM server to the environmental entity is unsuccessful.
10. Stop capturing packets.

MDM Agent to MDM Server (TLS):

1. Begin capturing packets between the mobile device, OCSP responder, and UEM server hosts.
2. Perform an action that causes the MDM Agent to initiate a TLS connection to the MDM Server.
3. Verify the connection from the MDM Agent to the MDM Server is successful.
4. Stop capturing packets.

5. Disconnect the connection between the MDM Agent and the OCSP responder.
6. Begin capturing packets from the OCSP responder server.
7. Begin capturing packets from the MDM server.
8. Perform an action that causes the MDM Agent to initiate a TLS connection to the MDM Server.
9. *Android:* Verify the connection from the MDM server to the environmental entity is unsuccessful
or
iOS/IPadOS: Verify the connection from the MDM server to the environmental entity is successful.
10. Stop capturing packets.

MDM Android Agent to MDM Server (Policy Signing):

1. Begin capturing packets between the mobile device, OCSP responder, and UEM server hosts.
2. From the MDM Server, configure a policy and transmit it to the Android mobile device
3. Verify the mobile device accepted the configured policy.
4. Stop capturing packets.

5. Disconnect the connection between the MDM Android Agent and the OCSP responder. 6. Begin capturing packets from the OCSP responder server. 7. From the MDM Server, configure a policy and transmit it to the mobile device. 8. Verify the mobile device rejected the configured policy. 9. Stop capturing packets from the OCSP responder server	
Test Results:	The evaluator successfully demonstrated use of a valid certificate requiring certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator manipulated the environment so that the TOE is unable to verify the validity of the certificate, and observed that the action selected in FIA_X509_EXT.2.2 was performed. - PASS
Execution Method:	Manual

048	[MDMPP]FIA_X509_EXT.5.1 – X.509 Unique Certificate – Test 1
Test Objective:	For each MDM Agent/platform listed as supported in the ST: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds as needed to perform this test. One of the following tests must be performed depending on whether the MDM agent allows for the loading of certificates. Test 1: [conditional] If the MDM agent allows for the loading of certificates: The evaluator shall initiate communications between the MDM Server and a client device over a trusted channel established using the device's unique certificate, verifying that a successful communication channel was established. The evaluator shall then attempt to initiate communications between the MDM Server and a second client device over a trusted channel established using the unique certificate from the first device, verifying that the MDM Server rejects this attempt at communication.
N/A – The MDM agent does not allow for the loading of certificates; therefore, this test assurance activity does not apply.	
Test Results:	N/A
Execution Method:	N/A

049	[MDMPP]FIA_X509_EXT.5.1 – X.509 Unique Certificate – Test 2

Test Objective:	For each MDM Agent/platform listed as supported in the ST: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds as needed to perform this test. One of the following tests must be performed depending on whether the MDM agent allows for the loading of certificates. Test 2: [conditional] If the MDM agent does not allow for the loading of certificates: The evaluator shall concurrently enroll 10 devices and ensure that the client certificate for each is unique, per the methods described in the TSS.
1. Begin capturing packets on the MDM Server. 2. Enroll a mobile device into Mobile Device Management (MDM). 3. Repeat Step 2 nine times to ensure that 10 mobile devices are concurrently enrolled into MDM. 4. Stop capturing packets on the MDM Server. 5. Inspect the packet capture and verify that each enrolled mobile device presents a unique client certificate to the MDM Server.	
Test Results:	The evaluator performed this test by first capturing packets on the MDM server. The evaluator then enrolled 10 mobile devices into the MDM server. The evaluator terminated the packet capture. The evaluator reviewed the packet capture and confirmed that, upon successful enrollment, each of the ten enrolled mobile devices presented a unique client certificate to the MDM server. - PASS
Execution Method:	Manual

4.3.5 Security Management

050	[MDMPP]FMT_MOF.1.1(1) – Management of Functions Behavior – Test 1
Test Objective:	Test 1: The evaluator shall attempt to access the functions and policies in FMT_SMF.1(1) as an unauthorized user and verify that the attempt fails.
1. Authenticate to the MDM Server console as the user with the Limited Admin role. 2. From within the Mobile Device Management Console: a. Attempt to lock the device: See “Command the mobile device to transition to the locked state” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058. b. Attempt to perform a full wipe of protected data: See “Command the mobile device to perform a full wipe of protected data” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058. c. Attempt to unenroll the device from management: See “Command the device to unenroll from management” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS	

– Test Case 058. d. Attempt to install new policies: See “Command the MDM Server to install new policies” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058. e. Attempt to query information from a device: See “Query the connectivity status of a device” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058. f. Attempt to wipe enterprise data: See “Wipe Enterprise data” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058. g. Attempt to define/create/install a device profile: See “Define a password length/complexity/lifetime” in [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058.

Test Results:	The evaluator performed this test by authenticating to the TOE (MDM server) as a non-security administrator. The evalautor then attempted to access the functions and policies defined in FMT_SMF.1(1) and was unsuccessful. – PASS
Execution Method:	Manual

051	[MDMPP]FMT_MOF.1.1(1) – Management of Functions Behavior – Test 2
Test Objective:	Test 2: [conditional] The evaluator shall attempt to access the functions and policies in FMT_SMF.1(3) as an unauthorized user and verify that the attempt fails.

1. Authenticate to the MDM Server console as the user with the Limited Admin role.
2. From within the MDM Server console:

   a. Configure application access groups:

      Attempt to access the following URLs:

      - https://uem.cctl.company.com/AirWatch/#/SmartGroup
      - https://uem.cctl.company.com/AirWatch/#/ApplicationGroup/ViewGrid
      - https://uem.cctl.company.com/AirWatch/#/AppManagement/ViewCategories

   b. Download applications:

      Attempt to access the following URLs:

      - https://uem.cctl.company.com/AirWatch/#/Apps/List/Internal?provisioningEnabled=False

	• https://uem.cctl.company.com/AirWatch/#/Apps/Edit/AddInternalApp?provisioningEnabled=False • https://uem.cctl.company.com/AirWatch/#/AirWatch/Apps/List/Public • https://uem.cctl.company.com/AirWatch/#/AppManagement/AddPublicApplication?productType=App&provisioningEnabled=False • https://uem.cctl.company.com/AirWatch/#/AirWatch/Apps/List/Purchased • https://uem.cctl.company.com/AirWatch/#/AppManagement/New?productType=Unknown • https://uem.cctl.company.com/AirWatch/#/apps/list/web • https://uem.cctl.company.com/AirWatch/#/Apps/Edit/AddWebApp
Test Results:	The evaluator performed this test by authenticating to the TOE (MDM server) as a non-security administrator. The evalautor then attempted to access the functions and policies defined in FMT_SMF.1(3) and was unsuccessful. – PASS
Execution Method:	Manual

052	[MDMPP]FMT_MOF.1.1(2) – Management of Functions Behavior (Enrollment)
Test Objective:	The test of this function is performed in conjunction with FIA_ENR_EXT.1.
Refer to FIA_ENR_EXT.1.1/ANDROID - Test Cases 027, 028, 029 and FIA_ENR_EXT.1.1/IOS - Test Cases 030, 031 032.	
Test Results:	Refer to FIA_ENR_EXT.1.1/ANDROID - Test Cases 027, 028, 029 and FIA_ENR_EXT.1.1/IOS - Test Cases 030, 031, 032. – PASS
Execution Method:	Manual

053	[MDMPP]FMT_MOF.1.1(3) – Management of Functions in (MAS Server Downloads)
Test Objective:	The evaluator shall ensure that the MAS Server verifies that the mobile device is enrolled in the MDM Server and is in a compliant state. The evaluator shall verify that an application cannot be downloaded from the MAS Server prior to enrolling the device with the MDM. The evaluator shall partially enroll the mobile device, so the device is connected to the MDM Server, but is not compliant and verify that applications cannot be downloaded.
According to FMT_SMR.1(2) from the Security Target TSS, "The MAS Server is logically integrated with the UEM Server." Therefore, when a device is not enrolled into the MDM Server, it is not possible to access or download applications from the MAS Server. **Partially enrolled mobile device, not belonging to appropriate application access group** 1. Authenticate to the MDM Server Console as the administrator. 2. Create a new user group by navigating to "Groups & Settings" > "Groups" > "User Groups". 3. Click "ADD" > "Add User Group" 4. Specify "Custom" for the type.	

5. Specify the Group Name, Description, and Managed By.
6. Click “Save”.
7. Create a new Smart Group by navigating to “Groups & Settings” > “Groups” > “Assignment Groups”.
8. Click “ADD SMART GROUP”.
9. Under “User Group” enter the User Group name that was created previously.
10. Specify a name for the Smart Group.
11. Click “Save”.
12. Assign an application to the newly created Smart Group by navigating to “Apps & Books” > “Applications” > “Native” > “ADD” > “Application File”.
13. Specify the Organization Group ID and Application File then click “Continue” and “Save & Assign”.
14. Click “Add Assignment” and specify the Smart Group created in Step 11.
15. Specify the App Delivery Method to “ON DEMAND” and then click “Create”.
16. Click “Save”.
17. Verify on the “Preview Assigned Devices” screen that none of the mobile devices should have access to the published application.
18. Click “PUBLISH”.
19. On the mobile device, launch the MDM Agent > “App Catalog”.
20. Verify that the App Catalog page does not return the presence of the app assigned in Step 12.

Mobile device belongs to appropriate application access group

21. Add a user to the SMART Group by navigating to “Groups and Settings” > “Groups” > “Assignment Groups”.
22. Locate the create SMART Group and click the “edit” button.
23. Click “Select Devices or Users” and click in the “Users” box and select the user that is being assigned the application.
24. Click “Add” then “Next”
25. Click “Publish”
26. Verify that the mobile devices assigned to that user have access to the published application via the MDM Agent > “App Catalog”.

Partially enrolled mobile device, non-compliant state

27. From the MDM Server console:

    Android:

    a. Navigate to “Add” > “Compliance Policy” > “Android”.
    b. Choose rule “Device Manufacturer is Samsung”, then click “NEXT”.

    iOS:

    a. Navigate to “Add” > “Compliance Policy” > “Apple iOS”.
    b. Choose rule “OS Version Is Not Apple iOS – iOS 13.0.0”, then click “NEXT”.

28. Choose action “Application”, “Block/Remove All Managed Apps”, then click “NEXT”.
29. Specify the appropriate Smart Group to apply the compliance policy, then click “NEXT”.
30. Click “FINISH & ACTIVATE”.

31. On the mobile device, launch the MDM Agent > "App Catalog". 32. Tap "Install" on to attempt to install the application published in Step 18. 33. Verify that the application published in Step 18 is not downloaded.	
Test Results:	The evaluator confirmed that it was not possible to download an application from the MAS Server prior to enrolling the device with the MDM. The evaluator partially enrolled the mobile device (and tested it in the following states: not belonging to appropriate application access group, not compliant) and verified that applications cannot be downloaded. Only when the mobile device was in a compliant state and a member to an appropriate application access group, it was possible to access an application from the MAS Server. – PASS
Execution Method:	Manual

054A	[MDMPP]FMT_POL_EXT.1.1 – Trusted Policy Update
Test Objective:	The evaluator shall perform a policy update in accordance with FMT_SMF.1(1). The evaluator shall examine the policy either at the MDM Server, in transmission, or at the MDM agent, and verify the TSF signs the update and provides it to the MDM Agent.
1. Perform Steps 1 - 8 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004. 2. Connect the mobile device to the debugger. 3. Perform Steps 11 - 17 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004. 4. After the transmission of the policy update completes and is received by the mobile device, use the debugger to inspect and verify that the police update is signed.	
Test Results:	The evaluator conducted this test by performing a policy update, examined it in transmission, and verified that it was signed by the TOE and that it was provided to the MDM Agent. - PASS
Execution Method:	Manual

054I	[MDMPP]FMT_POL_EXT.1.1 – Trusted Policy Update
Test Objective:	The evaluator shall perform a policy update in accordance with FMT_SMF.1(1). The evaluator shall examine the policy either at the MDM Server, in transmission, or at the MDM agent, and verify the TSF signs the update and provides it to the MDM Agent.
iOS Platform MDM Agent (iOS): 1. Perform Steps 1 - 10 in [AGENTMOD]FAU_ALT_EXT.2/IOS – Test Case 008. 2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine. 3. Perform Steps 11 - 19 in [AGENTMOD]FAU_ALT_EXT.2/IOS – Test Case 008. 4. Intercept the HTTP response containing the policy and its signature. 5. After the transmission of the policy update completes, confirm it is received by the mobile device. 6. Inspect the data captured in Step 4 and verify that the policy is signed. **TOE MDM Agent (iOS):** 1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent. 2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine. 3. Intercept the HTTP response containing the policy and its signature. 4. After the transmission of the policy update completes, confirm it is received by the mobile device. 5. Inspect the data captured in Step 4 and verify that the policy is signed.	

Test Results:	The evaluator conducted this test by performing a policy update, examined it in transmission, and verified that it was signed by the TOE and that it was provided to the MDM Agent. – PASS
Execution Method:	Manual

054P	[MDMPP]FMT_POL_EXT.1.1 – Trusted Policy Update
Test Objective:	The evaluator shall perform a policy update in accordance with FMT_SMF.1(1). The evaluator shall examine the policy either at the MDM Server, in transmission, or at the MDM agent, and verify the TSF signs the update and provides it to the MDM Agent.

iOS Platform MDM Agent (iOS):

1. Perform Steps 1 - 10 in [AGENTMOD]FAU_ALT_EXT.2/IOS – Test Case 008.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Perform Steps 11 - 19 in [AGENTMOD]FAU_ALT_EXT.2/IOS – Test Case 008.
4. Intercept the HTTP response containing the policy and its signature.
5. After the transmission of the policy update completes, confirm it is received by the mobile device.
6. Inspect the data captured in Step 4 and verify that the policy is signed.

TOE MDM Agent (iOS):

1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Intercept the HTTP response containing the policy and its signature.
4. After the transmission of the policy update completes, confirm it is received by the mobile device.
5. Inspect the data captured in Step 4 and verify that the policy is signed.

Test Results:	The evaluator conducted this test by performing a policy update, examined it in transmission, and verified that it was signed by the TOE and that it was provided to the MDM Agent. – PASS
Execution Method:	Manual

055A	[AGENTMOD]FMT_POL_EXT.2 – Agent Trusted Policy Update – Test 1
Test Objective:	This evaluation activity is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the Base-PPs. Test 1: The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall verify the update is signed and is provided to the MDM Agent. The evaluator shall verify the MDM Agent accepts the digitally signed policy.
This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054A.	
Test Results:	This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054A. – PASS
Execution Method:	Manual

055I	[AGENTMOD]FMT_POL_EXT.2 – Agent Trusted Policy Update – Test 1
Test Objective:	This evaluation activity is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the Base-PPs. Test 1: The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall verify the update is signed and is provided to the MDM Agent. The evaluator shall verify the MDM Agent accepts the digitally signed policy.
This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054I.	
Test Results:	This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054I. – PASS
Execution Method:	Manual

055P	[AGENTMOD]FMT_POL_EXT.2 – Agent Trusted Policy Update – Test 1
Test Objective:	This evaluation activity is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the Base-PPs. Test 1: The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall verify the update is signed and is provided to the MDM Agent. The evaluator shall verify the MDM Agent accepts the digitally signed policy.
This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054P.	
Test Results:	This is tested in conjunction with the testing activities in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054P. – PASS
Execution Method:	Manual

056	[AGENTMOD]FMT_POL_EXT.2 – Agent Trusted Policy Update – Test 2
Test Objective:	This evaluation activity is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the Base-PPs. Test 2: The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall provide an unsigned and an incorrectly signed policy to the MDM Agent. The evaluator shall verify the MDM Agent does not accept the digitally signed policy.
MDM Agent (Android): **Unsigned Policy** 1. Perform Steps 1 - 10 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004. 2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine. 3. Perform Steps 11 - 17 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004. 4. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is missing. 5. Verify on the mobile device that Camera is still disabled. 6. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied. **Incorrectly Signed Policy**	

1. Perform Steps 1 - 10 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Perform Steps 11 - 17 in [AGENTMOD]FAU_ALT_EXT.2/ANDROID – Test Case 004.
4. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is incorrectly signed.
5. Verify on the mobile device that Camera is still disabled.
6. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied.

TOE MDM Agent (iOS):

Unsigned Policy

1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is missing:

   a. Set a breakpoint (deviceservices/awmdmsdk/v3/processor AND /deviceservices/securechannel.aws/v2?deviceID<unique-device-ID>&bundledid=com.air.watch.agent) and remove the “x-aw-policy-request-path-signature” header from the HTTP response.

4. Verify on the mobile device that the iOS Hub Agent authentication function complexity is unchanged.
5. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied.

Incorrectly Signed Policy

1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is incorrectly signed:
   a. Set a breakpoint (deviceservices/awmdmsdk/v3/processor AND /deviceservices/securechannel.aws/v2?deviceID<unique-device-ID>&bundledid=com.air.watch.agent) and modify the signature in the “x-aw-policy-request-path-signature” header from the HTTP response.

4. Verify on the mobile device that the iOS Hub Agent authentication function complexity is unchanged.
5. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied.

TOE MDM Agent (iPadOS):

Unsigned Policy

1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is missing:

   a. Set a breakpoint (deviceservices/awmdmsdk/v3/processor AND /deviceservices/securechannel.aws/v2?deviceID<unique-device-ID>&bundledid=com.air.watch.agent) and remove the “x-aw-policy-request-path-signature” header from the HTTP response.

4. Verify on the mobile device that the iOS Hub Agent authentication function complexity is unchanged.
5. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied.

Incorrectly Signed Policy

1. Configure a policy (i.e. increase minimum length of the iOS Hub Agent authentication complexity requirements) that is consumed by the (TOE) iOS Hub Agent.
2. Configure the mobile device Wi-Fi settings to connect via the proxy server setup on the test machine.
3. Intercept the HTTP response containing the policy signature and then modify the HTTP header such that the policy signature is incorrectly signed:

   a. Set a breakpoint (deviceservices/awmdmsdk/v3/processor AND /deviceservices/securechannel.aws/v2?deviceID<unique-device-ID>&bundledid=com.air.watch.agent) and modify the signature in the “x-aw-policy-request-path-signature” header from the HTTP response.

4. Verify on the mobile device that the iOS Hub Agent authentication function complexity is unchanged.
5. Verify via audit records that the mobile device reports that the policy update failed validation and was not applied.

Test Results:	The evaluator performed this test by sending a policy update from the MDM Server to the MDM Agent. The evaluator then removed (unsigned) the signature of the policy in transit and verified the MDM Agent did not accept the policy. Finally, the evaluator sent another policy update and modified (incorrectly signed) the signature of the policy in transit and verified the MDM Agent did not accept the policy. - PASS
Execution Method:	Manual

057	[MDMPP]FMT_SMF.1.1(1)/ANDROID – Specification of Management Functions (Server configuration of Agent)

Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluator shall verify the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed above.

1. ***Command the mobile device to transition to the locked state:***

    a. Ensure that the device is currently in an unlocked state.
    b. Navigate to “Devices” > “List View”.
    c. Choose the specific Device to transition to the locked state under the “General Info” column.
    d. On the top toolbar, choose “Lock”.
    e. Confirm the device lock request.
    f. Verify that the device transitioned to the locked state.

2. ***Command the mobile device to perform a full wipe of protected data:***

    a. Navigate to “Devices” > “List View”.
    b. Choose the specific Device to execute Device Wipe under the “General Info” column.
    c. Choose “More Actions” from the top right-hand menu (from the Menu reading “Query”, “Send”, “Lock, and “More Actions”).
    d. Choose “Device Wipe” under the Management heading.
    e. Specify the Target Data to Wipe: “Both Device & Removable Storage” and a reason for the wipe in the Notes section.
    f. Review the information, then enter the Administrator PIN that was created at the initial Administrator logon.
    g. The Device Wipe will execute and provide confirmation.
    h. Verify that the device has been wiped of all protected data.

3. ***Command the device to unenroll from management:***

    a. Navigate to “Devices” > “List View”.
    b. Choose the specific Device to execute Device Wipe under the “General Info” column.
    c. Choose “More Actions” from the top right-hand menu (from the Menu reading “Query”, “Send”, “Lock, and “More Actions”).
    d. Choose “Device Wipe” under the Management heading.
    e. Specify the Target Data to Wipe: “Both Device & Removable Storage” and a reason for the wipe in the Notes section.
    f. Review the information, then enter the Administrator PIN that was created at the initial Administrator logon.
    g. The Device Wipe will execute and provide confirmation.
    h. Verify that the device has been unenrolled from management.

4. ***Command the MDM Server to install new policies:***

NOTE: In reference to “Policies” provided to the Mobile Device platform to manage and monitor security, configurations, and applications, two main methods are used: “Device Profiles” and “Compliance Engine Policies”. The guide specifies many of these exact configurations for each function-specific SFRID as warranted, but the below is a high level of how both “Device Profiles” and “Compliance Policies” are created in the UEM Mobile Device Management Console:

Compliance Policies:

a. Choose “Add”.
b. Select “Compliance Policy”.
c. Choose applicable Platform (Android) as covered by this Security Target and managed by the organization.
d. In drop-down, select to match “All” of the chosen policies, or “Any” of the chosen policies.
e. In first drop-down menu, select Rule to enforce, and in second drop-down, select particular caveats of chosen Rule as appropriate.
f. Click “Next”.
g. In first drop-down, select particular action type, and in second drop-down, select particular caveats of Action to execute (note: this action will automatically take place when device is found to be out of compliance with Chosen “Rule” in step “e”).
h. Click “Next”.
i. Click in “Assigned Groups” box, and select which Organization, Smart Group, or User Group to apply Policy to (note: will be enforced on these users and/or devices).
j. Click “Next”.
k. Review Summary and click “Finish and Activate”.

Device Profiles:

a. Choose “Add” > “Profile”.
b. Choose applicable Platform (Android) as covered by this Security Target and managed by the organization.
c. In General Payload tab, give the Profile an assigned name to denote its function, fill out or choose any applicable functions (recommended to leave as default for Common Criteria configuration), and click in “Assigned Groups” box and choose which Organization, Smart Group, or User Group to apply Profile to (note: will be enforced on these devices and/or users).
d. Select particular Profile Payload to include and enter caveats for the function (See particular sections of this SFR for specific Payload configurations to be deployed in the Common Criteria validated configuration. It is recommended that only one Payload be applied to each individual created Device Profile).
e. Click “Save and Publish”.
f. Review Devices and User to be applied to and click “Publish”.

5. *Query the connectivity status of a device:*

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to query the connectivity status under the “General Info” column.
c. Choose “Query” from the top toolbar.
d. Click “Ok” on the browser initiated pop-up confirmation.
e. Navigate to “Devices” > “List View”.
f. Refresh the List View section.
g. Verify the time value under the “Last Seen” column.

6. ***Query the current version of the device firmware/software:***
    a. Navigate to “Devices” > “List View”.
    b. Choose the specific device to query the current version of the device firmware/software under the “General Info” column.
    c. Choose “Query” from the top toolbar.
    d. Click “Ok” on the browser initiated pop-up confirmation.
    e. Navigate to “Devices” > “List View”.
    f. Refresh the List View section.
    g. Verify the device software version (i.e. operating system and version) for the specific device under in the 1st and 3rd lines under the “Platform” column.

7. ***Query the current version of the device hardware model:***
    a. Navigate to “Devices” > “List View”.
    b. Choose the specific device to query the current version of the hardware model under the “General Info” column.
    c. Choose “Query” from the top toolbar.
    d. Click “Ok” on the browser initiated pop-up confirmation.
    e. Navigate to “Devices” > “List View”.
    f. Refresh the List View section.
    g. Verify the hardware model version for the specific device in the 2nd line under the “Platform Column”

8. ***Query the current version of the installed applications:***
    a. Navigate to “Devices” > “List View”.
    b. Choose the specific device to query the current version of installed applications under the “General Info” column.
    c. Choose “Query” from the top toolbar.
    d. Click “Ok” on the browser initiated pop-up confirmation.
    e. Navigate to “Apps” tab.
    f. Refresh the Apps section.
    g. Verify the list of applications and the current version under the “Name” and “App Status” (version) headings, respectively.

9. ***Import X.509v3 certificates into the Trust Anchor Database:***
    a. Choose “Add” > “Profile”.
    b. Choose applicable Platform (Android) as covered by this Security Target and managed by the organization.
    c. Give the profile an assigned name to denote its function.
    d. Select “Credentials” > “Upload” > Specify a Credential Name and then Upload a certificate.
    e. Click “Next” and specify the device assignment.
    f. Click “Save & Publish”.

10. ***Install applications:***

a. Navigate to “Devices” > “List View.

b. Choose the specific device to install an application on under the “General Info” column.

c. Click the “Apps” tab.

d. Install an application to the device by choosing the assigned app and then clicking “INSTALL”.

e. Confirm the application install.

f. Verify that the application has been successfully installed on the device.

11. ***Update system software:***

1. Configure an API key to allow Samsung E-FOTA to establish a connection to the MDM Server.
2. Navigate to “Groups & Settings” > “All Settings” > “System” > “Advanced” > “API” > “REST API”.
3. Click “ADD” and then specify the Service Name, Account Type (Admin), then click “SAVE”.
4. Navigate to “Apps & Books” > “Applications” > “Native” and add “Knox Service Plugin” and “Knox E-FOTA” to the internal MAS server.
5. Configure the Knox Service Plugin application configuration to enable the following: Device-wide policies, Firmware update (FOTA) policy, Enable firmware controls, Enable Allow firmware update over-the-air, Enable Allow firmware update in recovery mode, Enable E-FOTA client installation & launch.
6. Assign the “Knox Service Plugin” and “Knox E-FOTA” apps to the Android device to be updated.
7. Supply the MDM Server URL, API key, and credentials to Samsung E-FOTA so that it can connect to the MDM Server.
8. From the Samsung E-FOTA portal, perform a sync to obtain the set of devices.
9. Create a firmware update campaign on the Samsung E-FOTA portal.
10. Assign the campaign to the device to be updated.
11. Initialize the firmware update to the device from the MDM Server by commanding the installation of the “Knox E-FOTA” and “Knox Service Plugin” applications to the mobile device.

12. ***Remove applications:***

a. Navigate to “Devices” > “List View”.

b. Choose the specific device to remove an application from under the “General Info” column.

c. Ensure that a managed application is already installed (See “Install applications”).

d. Click on the “Apps” tab.

e. Remove an application from the device by selecting the application and choosing “REMOVE”.

f. Click “Ok” to confirm the application removal.

g. Verify the application has been removed from the mobile device.

13. ***Remove Enterprise applications:***

a. Navigate to “Devices” > “List View”.

b. Choose the specific device to remove an application from under the “General Info” column.

c. Ensure that a managed application is already installed (See “Install applications”).

d. Click on the “Apps” tab.

e. Remove an application from the device by selecting the application and choosing "REMOVE".
f. Click "Ok" to confirm the application removal.
g. Verify the application has been removed from the mobile device.

14. ***Wipe Enterprise data:***
    a. Navigate to "Devices" > "List View".
    b. Choose the specific Device to execute Device Wipe under the "General Info" column.
    c. Choose "More Actions" from the top right-hand menu (from the Menu reading "Query", "Send", "Lock, and "More Actions").
    d. Choose "Device Wipe" under the Management heading.
    e. Specify the Target Data to Wipe: "Both Device & Removable Storage" and a reason for the wipe in the Notes section.
    f. Review the information, then enter the Administrator PIN that was created at the initial Administrator logon.
    g. The Device Wipe will execute and provide confirmation.
    h. Verify that the device has been wiped of all managed enterprise data.

15. ***Remove imported X.509v3 certificates from the Trust Anchor Database:***

a. Navigate to "Devices" > "List View".
b. Choose the specific mobile device.
c. Choose "Profiles".
d. Choose the profile that pushed the certificate.
e. Click "Remove".
f. Confirm the profile removal.

16. ***Alert the user:***

a. Navigate to "Devices" > "List View".
b. Choose the specific device to send the user a message to under the "General Info" column.
c. Click "Send" from the top toolbar.
d. Select the message type as "Push Notification".
e. Enter the message text in the dialog box then click "Send".

17. ***N/A – OMIT***
18. ***N/A – OMIT***
19. ***N/A – OMIT***

20. ***Retrieve MD-software integrity verification values:***

a. Navigate to "Groups & Settings" > "All Settings" > "Apps" > "Settings & Policies" > "Settings" > "Custom Settings".
b. Specify the following in the "Custom Settings" field:

```
{ "SafetyNetEnabled":true }
```

c. Click “Save”.
d. Verify SafetyNet Attestation on the Summary tab within “Device details” view of the mobile device.

21. ***N/A – OMIT***

22. ***N/A – OMIT***

23. ***N/A – OMIT***
24. ***N/A – OMIT***

Mobile Device Configuration Policies:

25. ***Define a password length/complexity/lifetime*:**
    a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”.
        i. Specify the required values for the naming of the profile.
        ii. On the Passcode tab, enable Device Passcode Policy.
        iii. On the Passcode tab, enable and choose a minimum passcode length of 8.
        iv. On the Passcode tab, enable and choose a minimum number of numbers value of 3.
        v. On the Passcode tab, enable and choose a maximum passcode age of 1 day.
    b. Choose “Next”, specify the device assignment.
    c. Click “Save & Publish”.

26. ***Define the session locking policy (screen-lock enabled/disabled, screen lock timeout, number of authentication failures):***

    a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”.
        i. Specify the required values for the naming of the profile.
        ii. Choose the “Passcode” tab.
        iii. On the Passcode tab, enable Device Passcode Policy.
        iv. Enable and choose a minimum passcode length of 4.
        v. In the “Device Lock Timeout Range (in Minutes)” field, enter maximum available amount of idle time of 1 minute after which the device transitions to the screen lock.
        vi. In the “Maximum Number of Failed Attempts” drop down, select 4 failed passcode authentication attempts before a device wipe is initiated.
        vii. Choose “Next”, specify the device assignment.
        viii. Click “Save & Publish”.

27. ***Define the wireless networks (SSIDs) to which the MD may connect:***

    a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”

b. Specify the required values for the naming of the profile.
c. On the “Restrictions” tab uncheck “Allow Managed Wi-Fi Profiles Changes” and uncheck “Allow Wi-Fi Changes”.
d. On the “Wi-Fi” tab specify the SSID value.
e. Specify the other Wi-Fi fields with their corresponding values as applicable to the specified Wi-Fi SSID.
f. Choose “Next”, specify the device assignment.
g. Click “Save & Publish”.

28. ***Define the security policy for each wireless network, including the CA(s) from which the MD will accept WLAN authentication server certificate(s), the security type, the authentication protocol, and the client credentials to be used for authentication:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. On the “Credentials” tab upload (Certificate #1) the client authentication certificate and (Certificate #2) the CA certificate.
d. On the “Wi-Fi” tab specify the SSID value.
e. Specify the other Wi-Fi fields with their corresponding values.
    i. Check “Hidden Network”.
    ii. Security Type: “WPA/WPA2 Enterprise”
    iii. Authentication Protocol (SFA Type): “TLS”
    iv. Authentication Credentials
        1. Identity: “Administrator”
        2. Identity Certificate: “Certificate #1”
        3. Root Certificate: “Certificate #2”
f. Choose “Next”, specify the device assignment.
g. Click “Save & Publish”.

29. ***Define the application installation policy by specifying authorized application repository(s) and denying application installation:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow Google Play” and “Allow Non-Market App Installation”.
e. Click the “Application Control” tab.
f. Check “Disable Access to Denied Apps”.
g. Choose “Next”, specify the device assignment.
h. Click “Save & Publish”.

Specifying a set of denied applications (an application deny list):

Refer to “FAU_ALT_EXT.1.1 – Test Case 003A – Presence of apps on a deny list”

30. ***Enable/disable policy for camera and screen capture across device:***

   a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
   b. Specify the required values for the naming of the profile.
   c. Click the “Restrictions” tab.
   d. Uncheck “Allow Camera” and “Allow screen capture”.
   e. Choose “Next”, specify the device assignment.
   f. Click “Save & Publish”.

31. ***Enable/disable policy for the VPN protection across mobile device:***

This is tested in FMT_SMF.1.1(1)/ANDROID – Test Case 057 (Subtest 054).

On a per-app basis:

   a. Deploy the Knox Service Plugin and Knox VPN Services as Internal apps in the MAS.
   b. Configure and assign the Knox Service Plugin with the following settings:

      Debug Mode: Enable

      Device-wide policies (Selectively applicable to Fully Manage Device (DO) or Work Profile-on company owned devices (WP-C) mode as noted): Configure and Enable device policy controls

      VPN policy (Premium) Enable VPN controls: Enable
      VPN type: Selected Apps(Per-app)
      Enable on-demand VPN: Disable

      VPN profiles (Premium) Profile name: VPN profile 1
      Vendor: Knox built-in
      Host: vpn.test.tld
      VPN connection type: IPSEC

      Parameters for Knox built-in VPN (for Strong Swan):
      ipsec_ike2_psk
      Identifier: vpn
      Pre-shared key: abc1234567890
      User certificate alias: clientcert
      CA certificate alias: CAcert
      Server certificate alias: vpnserver

   c. Click Create, Save, and Publish.
   d. Command the MSM server to install the Knox VPN Services and Knox Service Plugin app to the mobile device.

32. ***Enable/disable policy for Bluetooth:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow Bluetooth”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

33. ***N/A - OMIT***

34. ***Enable/disable policy for Wi-Fi tethering, USB tethering, and Bluetooth tethering:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow All Tethering”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

35. ***N/A – OMIT***

36. ***N/A – OMIT***

37. ***N/A – OMIT***

38. ***Enable/disable policy for local authentication bypass:***

a. Configure a Passcode policy in a new profile and set the initial passcode:
    i. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
    ii. Specify the required values for the naming of the profile.
    iii. Click the “Passcode” tab.
    iv. Check “Enable Device Passcode Policy”.
    v. Enable “Set initial passcode”.
    vi. Specify the initial device passcode.
    vii. Click “Next”.

viii. Click “Save & Publish”.

b. Navigate to the “Device” > “List View” page.
c. Click on the specified device to view device details.
d. Navigate to “More Actions” > “Change Device Passcode”.
e. Specify the desired device passcode.

39. ***N/A – OMIT***

40. ***Enable/disable policy for display notification in the locked state of email notifications, calendar appointments, contact associated with phone call notification, text message notification, other application-based notifications:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow Keyguard Notifications”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

41. ***N/A – OMIT***
42. ***N/A – OMIT***
43. ***N/A – OMIT***
44. ***N/A – OMIT***
45. ***N/A – OMIT***
46. ***N/A – OMIT***

47. ***Define the unlock banner policy:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Custom Messages” tab.
d. Choose “Enable” for “Set a lockscreen message”.
e. Specify the lockscreen message.
f. Choose “Next”, specify the device assignment.
g. Click “Save & Publish”.

48. ***N/A – OMIT***

49. ***Enable/disable USB mass storage mode:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”

b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow USB File Transfer” and “Allow Mounting Physical Storage Media”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

50. ***Enable/disable backup of the following items (all applications, selected applications, selected groups of applications, configuration data) to locally connected system and remote system:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow Backup Service”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

Backup to locally connected system is tested as part of (49) - enable/disable USB mass storage mode.

51. ***Enable/disable hotspot functionality authenticated by no authentication, USB tethering authenticated by no authentication:***

This is tested as part of (34) - Enable/disable policy for Wi-Fi tethering, USB tethering, and Bluetooth tethering as the “Allow All Tethering” option is configured in that subtest, which affects the functionality claimed for subtest 51.

52. ***Enable/disable location services across device:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Set “Allow Location Service Configuration (Managed devices only)” to “Allow No Location Access” and “Allow User to Modify Location Settings”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

53. ***Enable/disable policy for user unenrollment:***

a. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Android” > “Intelligent Hub Settings”.
b. Choose “ENABLED” for “Block User Unenrollment”.
c. Click “Save”.

54. ***Enable/disable policy for the Always-On VPN protection across device:***

a. Deploy the Knox Service Plugin and Knox VPN Services as Internal apps in the MAS.
b. Configure and assign the Knox Service Plugin with the following settings:

Debug Mode: Enable

Device-wide policies (Selectively applicable to Fully Manage Device (DO) or Work Profile-on company owned devices (WP-C) mode as noted): Configure and Enable device policy controls

VPN policy (Premium) Enable VPN controls: Enable
VPN type: Device-wide
Enable on-demand VPN: Disable

VPN profiles (Premium) Profile name: VPN profile 1
Vendor: Knox built-in
Host: vpn.test.tld
VPN connection type: IPSEC

Parameters for Knox built-in VPN (for Strong Swan):
ipsec_ike2_psk
Identifier: vpn
Pre-shared key: abc1234567890
User certificate alias: clientcert
CA certificate alias: CAcert
Server certificate alias: vpnserver

c. Click Create, Save, and Publish.
d. Command the MSM server to install the Knox VPN Services and Knox Service Plugin app to the mobile device.

55. ***Enable/disable policy for use of a Biometric Authentication Factor:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Android”
b. Specify the required values for the naming of the profile.
c. Click the “Restrictions” tab.
d. Uncheck “Allow Keyguard Fingerprint Sensor”.
e. Choose “Next”, specify the device assignment.
f. Click “Save & Publish”.

56. ***N/A – OMIT***

57. ***N/A – OMIT***

58. ***Enable/disable automatic updates of system software:***

a. Deploy the Knox Service Plugin as an Internal app in the MAS.
b. Configure and assign the Knox Service Plugin with the following settings:

Debug Mode: Enable

Device-wide policies (Selectively applicable to Fully Manage Device (DO) or Work

Profile-on company owned devices (WP-C) mode as noted): Configure and Enable device policy controls

Firmware update (FOTA) policy: Enable firmware controls: Enable
Allow firmware update over-the-air: Disable

c. Click Create, Save, and Publish.
d. Command the MSM server to install the Knox Service Plugin app to the mobile device.

59. ***N/A – OMIT***

60. ***Application installation policy by specifying a set of allowed applications (an application allow list):***

Refer to [MDMPP]FAU_ALT_EXT.1.1 – Test 3 (Test Case 003A). The configuration of "an application allow list" is conducted in conjunction with the testing performed in: Presence of apps on deny list, presence of apps not on allow list, absence of required apps.

Test Results:	The evaluator verified the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed in the ST. – PASS
Execution Method:	Manual

058	[MDMPP]FMT_SMF.1.1(1)/IOS – Specification of Management Functions (Server configuration of Agent)
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluator shall verify the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed above.

1. ***Command the mobile device to transition to the locked state:***
   a. Ensure that the device is currently in an unlocked state.
   b. Navigate to "Devices" > "List View".
   c. Choose the specific Device to transition to the locked state under the "General Info" column.
   d. On the top toolbar, choose "Lock".
   e. Confirm the device lock request.
   f. Verify that the device transitioned to the locked state.

2. ***Command the mobile device to perform a full wipe of protected data:***
   g. Navigate to "Devices" > "List View".
   h. Choose the specific Device to execute Device Wipe under the "General Info" column.
   i. Choose "More Actions" from the top right-hand menu (from the Menu reading "Query", "Send", "Lock, and "More Actions").
   j. Choose "Device Wipe" under the Management heading.
   k. Review the information, specify a reason for the wipe in the Notes, then enter the Administrator PIN that was created at the initial Administrator logon.
   l. The Device Wipe will execute and provide confirmation.

m. Verify that the device has been wiped of all protected data.

3. *Command the device to unenroll from management:*

n. Navigate to “Devices” > “List View”.
o. Choose the specific Device to execute Device Wipe under the “General Info” column.
p. Choose “More Actions” from the top right-hand menu (from the Menu reading “Query”, “Send”, “Lock, and “More Actions”).
q. Choose “Device Wipe” under the Management heading.
r. Review the information, specify a reason for the wipe in the Notes, then enter the Administrator PIN that was created at the initial Administrator logon.
s. The Device Wipe will execute and provide confirmation.
t. Verify that the device has been unenrolled from management.

4. *Command the MDM Server to install new policies:*

NOTE: In reference to “Policies” provided to the Mobile Device platform to manage and monitor security, configurations, and applications, two main methods are used: “Device Profiles” and “Compliance Engine Policies”. Below is a high level of how both “Device Profiles” and “Compliance Policies” are created in the UEM Mobile Device Management Console:

<u>Compliance Policies:</u>

a. From the top navigation bar, choose “Add” -> “Compliance Policy”.
b. Choose applicable Platform (iOS) as covered by this Security Target and managed by the organization.
c. In drop-down, select to match “All” of the chosen policies, or “Any” of the chosen policies.
d. In first drop-down menu, select Rule to enforce, and in second drop-down, select particular caveats of chosen Rule as appropriate.
e. Click “Next”.
f. In first drop-down, select particular action type, and in second drop-down, select particular caveats of Action to execute (note: this action will automatically take place when device is found to be out of compliance with Chosen “Rule” in step “e”).
g. Click “Next”.
h. Click in “Smart Groups” box, and select which Organization, Smart Group, or User Group to apply Policy to (note: will be enforced on these users and/or devices).
i. Click “Next”.
j. Review Summary and click “Finish and Activate”.

<u>Device Profiles:</u>

a. From the top navigation bar, choose “Add” -> “Profile”.
b. Choose applicable Platform (iOS) as covered by this Security Target and managed by the organization.
c. In General Payload tab, give the Profile an assigned name to denote its function, fill out or choose any applicable functions (recommended to leave as default for Common Criteria configuration), and click in “Smart Groups” box and choose which Organization, Smart

Group, or User Group to apply Profile to (note: will be enforced on these devices and/or users).

d. Select particular Profile Payload to include and enter caveats for the function (See particular sections of this SFR for specific Payload configurations to be deployed in the Common Criteria validated configuration. It is recommended that only one Payload be applied to each individual created Device Profile).
e. Click “Save and Publish”.
f. Review Devices and User to be applied to and click “Publish”.

5. ***Query the connectivity status of a device:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to query the connectivity status under the “General Info” column.
c. Choose “Query” from the top toolbar.
d. Click “Ok” on the browser initiated pop-up confirmation.
e. Navigate to “Devices” > “List View”.
f. Refresh the List View section.
g. Verify the time value under the “Last Seen” column.

6. ***Query the current version of the device firmware/software:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to query the current version of the device firmware/software under the “General Info” column.
c. Choose “Query” from the top toolbar.
d. Click “Ok” on the browser initiated pop-up confirmation.
e. Navigate to “Devices” > “List View”.
f. Refresh the List View section.
g. Verify the device software version (i.e. operating system and version) for the specific device under in the 1st and 3rd lines under the “Platform” column.

7. ***Query the current version of the device hardware model:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to query the current version of the hardware model under the “General Info” column.
c. Choose “Query” from the top toolbar.
d. Click “Ok” on the browser initiated pop-up confirmation.
e. Navigate to “Devices” > “List View”.
f. Refresh the List View section.
g. Verify the hardware model version for the specific device in the 2nd line under the “Platform Column”

8. ***Query the current version of the installed applications:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to query the current version of installed applications under the “General Info” column.

c. Choose “Query” from the top toolbar.
d. Click “Ok” on the browser initiated pop-up confirmation.
e. Navigate to “Apps” tab.
f. Refresh the Apps section.
g. Verify the list of applications and the current version under the “Name” and “Installed Version” headings, respectively.

9. ***Import X.509v3 certificates into the Trust Anchor Database:***
a. Choose “Add” > “Profile”.
b. Choose applicable Platform (iOS) as covered by this Security Target and managed by the organization.
c. In General Payload tab, give the Profile an assigned name to denote its function, fill out or choose any applicable functions (recommended to leave as default for Common Criteria configuration), and click in “Assigned Groups” box and choose which Organization, Smart Group, or User Group to apply Profile to (note: will be enforced on these devices and/or users).
d. Select “Credentials” > “Upload” > Specify a Credential Name and then Upload a certificate.
e. Click “Save and Publish”.
f. Review Devices and User to be applied to and click “Publish”.

10. ***Install applications:***
a. Navigate to “Devices” > “List View.
b. Choose the specific device to install an application on under the “General Info” column.
c. Click the “Apps” tab.
d. Install an application to the device by clicking “Install”.
e. Confirm the application install.
f. Verify that the application has been successfully installed on the device.

11. ***Update system software:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to install the system software update from under the “General Info” column.
c. Choose “Updates”.
d. Select the latest update form the list of available updates and then click “PUBLISH”.
e. Select “Download/Install the software update” for the Device Installation Method, then click “SEND”.
f. Once the update is downloaded, repeat Steps (a) through (c).
g. For “Device Installation Method” choose “Install an already downloaded software update” and then click “SEND”.

12. ***Remove applications:***

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to remove an application from under the “General Info” column.
c. Ensure that a managed application is already installed (See “Install applications”).
d. Click on the “Apps” tab.
e. Remove an application from the device by clicking “Remove”.
f. Click “Ok” to confirm the application removal.
g. Verify the application has been removed from the mobile device.

13. *Remove Enterprise applications:*

See “12. Remove applications”

14. *Wipe Enterprise data:*

a. Navigate to “Devices” > “List View”.
b. Choose the specific Device to execute Device Wipe under the “General Info” column.
c. Choose “More Actions” from the top right-hand menu (from the Menu reading “Query”, “Send”, “Lock, and “More Actions”).
d. Choose “Device Wipe” under the Management heading.
e. Review the information, specify a reason for the wipe in the Notes, then enter the Administrator PIN that was created at the initial Administrator logon.
f. The Device Wipe will execute and provide confirmation.
g. Verify that the device has been unenrolled from management.

15. *Remove imported X.509v3 certificates from the Trust Anchor Database:*

a. Navigate to “Devices” > “List View”.
b. Choose the specific device under the “General Info” column.
c. Choose “Profiles”.
d. Choose the profile that pushed the certificate.
e. Click “Devices” > “Remove Profile”.
f. Confirm the profile removal.

16. *Alert the user:*

a. Navigate to “Devices” > “List View”.
b. Choose the specific device to send the user a message to under the “General Info” column.
c. Click “Send” from the top toolbar.
d. Select the message type as “Push Notification”.
e. Select the Application Name as “Intelligent Hub”.
f. Enter the message text in the dialog box then click “Send”.

17. *N/A – OMIT*

18. ***N/A – OMIT***
19. ***N/A – OMIT***
20. ***N/A – OMIT***
21. ***N/A – OMIT***

22. ***N/A – OMIT***

23. ***Revoke Biometric template:***
    u. Navigate to "Devices" > "List View".
    v. Choose the specific device under the "General Info" column.
    w. Select the "More Actions" dropdown > "Clear Passcode" > "Device".
    x. Verify that the device can be accessed without authentication credentials.

24. ***N/A – OMIT***

Mobile Device Configuration Policies:

25. ***Define a password length/complexity/lifetime*:**
    a. Create a new Profile by selecting from the top navigation pane, choose "Add" > "Profile" > "Apple iOS" > "Device Profile".
        vi. Specify the required values for the General tab, including the device assigned groups.
        vii. On the Passcode tab, check "Require passcode on device".
        viii. On the Passcode tab, enable and choose a minimum passcode length of 8.
        ix. On the Passcode tab, enable and choose a minimum number of complex characters value of 3.
        x. On the Passcode tab, enable and choose a maximum passcode age of 1 day.
    b. Choose "Save & Publish" from the Profile creation page.
    c. Confirm the devices/users for the policy assignment, then click "Publish".

26. ***Define the session locking policy for screen-lock enabled/disabled, screen lock timeout, and number of authentication failures:***

    a. Create a new Profile by selecting from the top navigation pane, choose "Add" > "Profile" > "Apple iOS" > "Device Profile".
        i. Specify the required values for the General tab, including the device assigned groups.
        ii. Choose the "Passcode" tab.
        iii. Check "Require passcode on device".
        iv. In the "Auto-Lock (min)" drop down, select the amount of idle time (e.g. 1 minute) after which the device transitions to the screen lock.
        v. In the "Maximum Number of Failed Attempts" drop down, select a number (e.g. 4 attempts) of failed passcode authentication attempts before a device wipe is initiated.
    b. Click "Save and Publish".
    c. Confirm the devices/users for the policy assignment, then click "Publish".

27. ***Define the wireless networks (SSIDs) to which the MD may connect:***

   a. Create a new Profile by selecting from the top navigation pane, choose "Add" > "Profile" > "Apple iOS" > "Device Profile"
   b. Specify the required values for the General tab, including the device assigned groups.
   c. On the "Restrictions" tab, click "Configure", then check the "Require Managed Wi-Fi" box.
   d. On the "Wi-Fi" tab, click "Configure", then specify the SSID value and Password.
   e. Submit the Profile by choosing "Save & Publish".
   f. Confirm the devices/users for the profile assignment, then click "Publish".

28. ***Define the wireless SSID security policy for each wireless network:***

   Specify the CA(s) from which the MD will accept WLAN authentication server certificate(s), the FQDN(s) of acceptable WLAN authentication server certificate(s), security type, authentication protocol, and client credentials to be used for authentication:

   a. Create a new Profile by selecting from the top navigation pane, choose "Add" > "Profile" > "Apple iOS" > "Device Profile"
   b. Specify the required fields in "General".
   c. On the "Wi-Fi" tab, click "Configure", then specify the SSID value.
   d. Specify the other Wi-Fi fields with their corresponding values.
      i. Permitted CAs and FQDN of server authentication certificate
      ii. Security Type
      iii. Authentication Protocol
      iv. Authentication Credentials
         1. For Certificate Authentication, click "Credentials" in the left toolbar, upload the applicable certificate, then specify the certificates on the Wi-Fi tab.
   e. Submit the Profile by choosing "Save & Publish".

Confirm the devices/users for the policy assignment, then click "Publish".

29. ***Define the application installation policy by:***

   Specifying authorized application repository(s):

   a. Create a new Profile by selecting from the top navigation pane, choose "Add" > "Profile" > "Apple iOS" > "Device Profile"
   b. In the "General" tab, give the Profile an assigned organizational name in the "Name" field, fill out any other optional information as warranted.
   c. Click in the "Assigned Groups" field.
   d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
   e. Click the "Restrictions" tab.
   f. Uncheck "Allow installing public apps".
   g. Click "Save & Publish".
   h. Confirm devices/users for the policy assignment, then click "Publish".
   i. Verify the restriction is enforced by the mobile device.

30. *Enable/disable policy for camera and screen capture across device:*

j. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”

k. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.

l. Click in the “Assigned Groups” field.

m. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.

n. Click the “Restrictions” tab.

o. Uncheck “Allow use of camera” and “Allow screen capture”.

p. Click “Save & Publish”.

q. Confirm devices/users for the policy assignment, then click “Publish”.

31. *Enable/disable policy for VPN protection across the mobile device and on a per-app basis:*

Full Device VPN:

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”

b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.

c. Click in the “Assigned Groups” field.

d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.

e. Click “VPN” then “Configure”.

f. In “Connection Name” block, give configuration organization defined name.

g. In “Connection Type” dropdown, select “IPSec (Cisco)”.

h. In “Server” field, type in hostname of VPN Server.

i. Check box for “Send all Traffic”.

j. Set the “Authentication” or “Proxy” information as necessary.

k. Click “Save & Publish”.

l. Confirm the devices/users for the policy to be assigned to, and then click “Publish”.

On a per-app basis (App-on-Demand VPN):

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”

b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.

c. Click in the “Assigned Groups” field.

d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.

e. Click “VPN” then “Configure”.

f. In “Connection Name” block, give configuration organization defined name.

g. In “Connection Type” dropdown, select “IPSec (Cisco)”.

h. In “Server” field, type in hostname of VPN Server.

i. Check box for “Per-App VPN Rules”.

j. Set the “Authentication” or “Proxy” information as necessary.

k. Click “Save & Publish”.

l. Confirm the devices/users for the policy to be assigned to, and then click “Publish”.

m. Navigate to “Apps & Books” > “Applications” > “Native” > “Public” > Choose a Public application (UFC).

n. Click the “Pencil” icon to Edit the application.
o. Click “Save & Assign”
p. Edit the assignment.
q. Click the “Tunnel & Other Attributes” tab.
r. Choose the VPN Profile that was created from the “Per-App VPN Profile” drop-down selection.
s. Click “Save”.
t. Click “Save and Publish”.

Verify the devices/users to be assigned the modified Application configuration, then click “PUBLISH”.

32. ***N/A - OMIT***

33. ***N/A – OMIT***

34. ***N/A – OMIT***

35. ***N/A – OMIT***

36. ***Enable policy for data-at-rest protection:***

According to VID11146 and VID11147: “…[D]ata is always encrypted for protection which requires the use of a passcode
on the device. Administrators must ensure that TOE users set a passcode…Users can check that data protection is enabled on their device in the Settings at Touch ID and
Passcode on devices without Face ID functionality (models with a Home button) and Face ID and Passcode on devices with Face ID functionality (models without a Home button). A passcode is required to access the device and this enables data protection on the device. No further configuration is required.”

Refer to FMT_SMF.1.1(1)/IOS – Test Case 58 (Subtest 25) where password policies are enforced.

37. ***N/A – OMIT***

38. ***Enable/disable policy for local authentication bypass:***

y. Navigate to “Devices” > “List View”.
z. Choose the specific device under the “General Info” column.
aa. Select the “More Actions” dropdown > “Clear Passcode” > “Device”.
bb. Verify that the device can be accessed without authentication credentials.

39. ***The Bluetooth trusted channel policy: enable/disable the Discoverable mode, change the Bluetooth device name:***

cc. Navigate to “Groups & Settings” > “Devices & Users” > “Apple” > “Apple iOS” > “Managed Settings”.
dd. Check “Corporate – Dedicated” for “Apply Default Settings To”.
ee. Check “Bluetooth” to enable Bluetooth (also enables Bluetooth discoverable mode).
ff. Click Save.
gg. Navigate to “Groups & Settings” > “Devices & Users” > “General” > “Friendly Name”.
hh. Enable “Set Device Name to Friendly Name” to change the Bluetooth device name.
ii. Click Save.

40. ***Enable/disable policy for display notification in the locked state of email notifications, calendar appointments, contact associated with phone call notification, text message notification, and other application-based notifications:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”
b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.
c. Click in the “Assigned Groups” field.
d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
e. Click the “Restrictions” tab.
f. Uncheck “Allow Wallet notifications in Lock screen” and “Show Notifications Center in Lock screen”.
g. Click the “Notifications” tab.
h. Click “Select App” and specify the App name.
i. Uncheck “Show in Lock Screen”, then click “SAVE”.
j. Click “Save & Publish”.
k. Confirm devices/users for the policy assignment, then click “Publish”.

41. ***N/A – OMIT***
42. ***N/A – OMIT***
43. ***N/A – OMIT***
44. ***N/A – OMIT***
45. ***N/A – OMIT***
46. ***N/A – OMIT***

47. ***Define the unlock banner policy:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”
b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.
c. Click in the “Assigned Groups” field.
d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
e. Click the “Lock Screen Message” tab.
f. Specify the “If lost return to” Message”.
g. Click “Save & Publish”.

h. Confirm devices/users for the policy assignment, then click “Publish”.

48. ***N/A – OMIT***

49. ***Enable/disable USB data transfer without user authentication:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”
b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.
c. Click in the “Assigned Groups” field.
d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
e. Click the “Restrictions” tab.
f. Check “Allow USB Restricted Mode”.
g. Click “Save & Publish”.
h. Confirm devices/users for the policy assignment, then click “Publish”.

50. ***Enable/disable backup of all applications and configuration data to remote system:***

a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”
b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill out any other optional information as warranted.
c. Click in the “Assigned Groups” field.
d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
e. Click “Restrictions” then “Configure”.
f. Uncheck all selections under the iCloud category.
g. Click “Save and Publish”.
h. Verify the devices/users to be assigned the modified Application configuration, then click “Publish”.
i. Verify that iCloud backups are disabled.

51. ***N/A – OMIT***
52. ***N/A – OMIT***
53. ***N/A – OMIT***

54. ***N/A – OMIT***

55. ***Enable/disable policy for use of Biometric Authentication Factor:***
a. Create a new Profile by selecting from the top navigation pane, choose “Add” > “Profile” > “Apple iOS” > “Device Profile”
b. In the “General” tab, give the Profile an assigned organizational name in the “Name” field, fill

out any other optional information as warranted.

c. Click in the “Assigned Groups” field.
d. Choose the appropriate Organization, Smart Group, or User Group to assign Profile to applicable users/devices.
e. Click “Restrictions” then “Configure”.
f. Uncheck “Allow Biometric ID to unlock device”.
g. Click “Save and Publish”.
h. Verify the devices/users to be assigned the modified Application configuration, then click “Save”.

56. ***N/A – OMIT***

57. ***N/A – OMIT***

58. ***N/A – OMIT***
59. ***N/A – OMIT***

60. ***Application installation policy by specifying a set of allowed applications (an application allow list):***

Refer to [MDMPP]FAU_ALT_EXT.1.1 – Test 3 (Test Case 003I, 003P). The configuration of “an application allow list” is conducted in conjunction with the testing performed in: Presence of apps on deny list, presence of apps not on allow list, absence of required apps.

61. ***iOS Hub Agent passcode authentication policy (minimum passcode length):***

This is tested in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054I (TOE MDM Agent (iOS)).
This is tested in [MDMPP]FMT_POL_EXT.1.1 – Test Case 054P (TOE MDM Agent (iOS)).

Verify that the mobile device enforces the passcode authentication policy complexity policy.

Test Results:	The evaluator verified the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed in the ST. - PASS
Execution Method:	Manual

059	[MDMPP]FMT_SMF.1.1(2)/ANDROID – Specification of Management Functions (Server Configuration of Server) – Test 1

Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 1: The evaluator shall configure the TSF authentication certificate(s) and verify that the correct certificate is used in established trusted connections (FPT_ITT.1(1), FPT_ITT.1(2), FTP_ITC.1(1), and FTP_TRP.1(2)).

1. From the MDM Server platform, launch Certificate Manager as the Computer account.
2. Import the certificate to be used by the MDM Server into the "Personal" certificate category.
3. From the MDM Server platform, launch Internet Information Services (IIS) Manager.
    a. In the Connections pane, expand "<Server-Name>" > "Sites" > "Default Web Site".
    b. Right-click on "Default Web Site" and choose "Edit Bindings…".
    c. Specify the port 443 SSL certificate imported from Step 2.
    d. Specify the port 8443 SSL certificate imported from Step 2.
4. Restart IIS services by executing the following command: iisreset

The use of the configured TSF authentication certificate is performed in conjunction with testing performed in FPT_ITT.1(2), FTP_ITC.1(1), and FTP_TRP.1(2).

Test Results:	The evaluator performed this test by configuring the TSF authentication certificate via the MDM server platform. Once the certificate was configured, the evalutor verified that this certificate was used in established trusted connections in conjunction with testing performed in FPT_ITT.1(2), FTP_ITC.1(1), and FTP_TRP.1(2). – PASS
Execution Method:	Manual

060	[MDMPP]FMT_SMF.1.1(2)/ANDROID – Specification of Management Functions (Server Configuration of Server) – Test 2
Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 2: (conditional) The evaluator shall configure the periodicity for the assigned list of commands to the agent for several configured time periods and shall verify that the MDM Server performs the commands schedule.

1. Authenticate to the MDM Server console as the administrator.
2. In order to configure the periodicity of the following tasks:

    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

3. Navigate to "Groups & Settings" > "Devices & Users" > "Android" > "Intelligent Hub Settings".
4. Specify the "Heartbeat Interval" value to 1 hour.
5. Verify that audit records are generated for the successful configuration of this function.
6. Wait at least for two Check-Ins to occur to establish that the interval between Check-In is 1 hour.
7. Repeat Steps 4 – 5, except in Step 4, configure the "Heartbeat Interval" to 30 minutes.
8. Wait at least for two Check-Ins to occur to establish that the interval between Check-Ins is 30 minutes.

Test Results:	The evaluator verified that an administrator is able to configure the periodicity for the assigned list of commands to the agent for several configured time periods and verified that the MDM Server performs the commands schedule. – PASS
Execution Method:	Manual

061	[MDMPP]FMT_SMF.1.1(2)/ANDROID – Specification of Management Functions (Server Configuration of Server) – Test 3
Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 3: (conditional) The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the MDM Server.

Configure the privacy-sensitive information that will and will not be collected from particular mobile devices

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Privacy”
3. Toggle the selection for “Personal Applications” to “Do Not Collect”.
4. Click “SAVE” and then confirm by entering the Console Security PIN.
5. Navigate to “Devices” > “List View”.
6. Choose the specific device to query the connectivity status under the “General Info” column.
7. Choose “Query” from the top toolbar.
8. Confirm the “Query” on the confirmation page.
9. Navigate to the “Apps” tab.
10. Confirm that Personal Apps are no longer collected.

Configure the interaction between TOE components

1. Authenticate to the Admin Console as an Administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “General” > “Enrollment” > “Authentication”.
3. Ensure “Devices Enrollment Mode” is set to “Registered Devices Only”.
4. Navigate to “Devices” > “Lifecycle” > “Enrollment Status” > “ADD” > “Allow Devices”.
5. Specify the permitted IMEIs.
6. Specify “IMEI” for device attribute.
7. Commit the selection.

Configure server administrator login session timeout

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Admin” > “Console Security” > “Session Management”
3. Specify 5 minutes for “Idle Session Timeout” and click “Save”.
4. Authenticate to the MDM Server in a new session.
5. Leave session inactive for 5 minutes.
6. Verify that immediately after 5 minutes, the session has timed out.

7. Authenticate to the MDM Server as the administrator.
8. Navigate to “Groups & Settings” > “All Settings” > “Admin” > “Console Security” > “Session Management”
9. Specify 7 minutes for “Idle Session Timeout” and click “Save”.
10. Authenticate to the MDM Server in a new session.
11. Leave session inactive for 7 minutes.
12. Verify that immediately after 7 minutes, the session has timed out.

Configure enterprise certificate to be used for signing policies (Android & iOS Hub MDM Agent)

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “System” > “Advanced” > “Policy Signing Certificate”
3. Click “Replace”, “Choose File” and “Upload”, and click “Save” to configure the enterprise certificate for signing policies.
4. Verify that the Policy Signing Certificate properties have been updated.

(Platform iOS/iPadOS MDM Agent)

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Apple” > “Profiles”.
3. Choose “REPLACE” for “Signing Certificate”.
4. Verify that the Policy Signing Certificate properties have been updated.

Configure MDM Agent/platform to perform a network reachability test

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Android” > “Intelligent Hub Settings”
3. Set “Heartbeat Interval” to 30 minutes and click “Save.”
4. Verify that the MDM Server configures the MDM Agent/platform to perform a network reachability test on the specified interval (30 minutes).
5. Set “Heartbeat Interval” to 1 hour and click “Save.”
6. Verify that the MDM Server configures the MDM Agent/platform to perform a network reachability test on the specified interval (1 hour).

Configure transfer of MDM server logs to another server for storage, analysis, and reporting

1. Authenticate to the MDM Server as the administrator.
2. Navigate to “Groups & Settings” > “All Settings” > “System” > “Enterprise Integration” > “Syslog”
3. Configure syslog server hostname, protocol, port, syslog facility, message tag, and message content.
4. Click “Save”
5. Verify changes save successfully and MDM Server can transfer audit logs to the new syslog server.

Test Results:	The evaluator performed this test by devising tests for each function defined in the ST for the assignment in the SFR, in addition to function c.4. For "configure the privacy-sensitive information that will and will not be collected from particular mobile devices", the evaluator successfully configured the TOE such that privacy-sensitive information (personal applications) are no longer collected. The evalutor then performed a device query to confirm that the intended behavior was enacted by the MDM server. For "configure the interaction between TOE components", the evaluator configured the MDM server to accept enrollment from registered devices only. The evaluator then specified the permitted registered devices and confirmed they were successfully registered. For "configure server administrator login session timeout", the evaluator configured the TOE (MDM server) idle session timeout value to "5" minutes. The evaluator terminated the session and re-established a new session. The evaluator left the session idle for 5 minutes and observed that the TOE successfully terminated the session due to inactivity. For "configure enterprise certificate to be used for signing policies (Android & iOS Hub MDM Agent)" and for "(Platform iOS/iPadOS MDM Agent)", the evaluator uploaded a new policy signing certificate used for signing policies sent to the enrolled mobile devices. The evaluator confirmed that the certificate used for signing policies was replaced. For "configure MDM Agent/platform to perform a network reachability test", the evaluator configured the TOE heartbeat interval to 30 minutes and then observed that a network reacability test was performed after 30 minutes elapsed. The evaluator then configured the TOE heartbeat interval to 1 hour and then observed that a network reacability test was performed after 1 hour elapsed. For "configure transfer of MDM server logs to another server for storage, analysis, and reporting", the evaluator configured the syslog server hostname, protocol, port, syslog facility, message tag, and message content and then confirmed that the MDM server transferred audit data to the configured syslog server. Finally, the evaluator confirmed that after performing each of these configurations, a corresponding audit record was generated by the TOE for the successful configuration operation. – PASS
Execution Method:	Manual

062	[MDMPP]FMT_SMF.1.1(2)/IOS – Specification of Management Functions (Server Configuration of Server) – Test 1
Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 1: The evaluator shall configure the TSF authentication certificate(s) and verify that the correct certificate is used in established trusted connections (FPT_ITT.1(1), FPT_ITT.1(2), FTP_ITC.1(1), and FTP_TRP.1(2)).
This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID – Test Case 059.	
Test Results:	This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID – Test Case 059. – PASS
Execution Method:	Manual

063	[MDMPP]FMT_SMF.1.1(2)/IOS – Specification of Management Functions (Server Configuration of Server) – Test 2
Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 2: (conditional) The evaluator shall configure the periodicity for the assigned list of commands to the agent for several configured time periods and shall verify that the MDM Server performs the commands schedule.

NOTE: Because the MDM Server user interface permits only values with hour time units, the enforcement aspect of this test is performed by directly modifying the database to specify minute time units and then applying the settings via the MDM Server user interface as opposed to modifying the values from the MDM Server user interface in order to achieve the test in a reasonable amount of time.

Periodicity Command Assignments configuration via the web UI:

1. Authenticate to the MDM Server as the administrator.
2. In order to configure the periodicity of the following tasks:

   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

3. Navigate to “Groups & Settings” > “All Settings” > “Devices and Users” > “Apple” > “MDM Sample Schedule”.
4. Specify the values to the options on the “MDM Sample Schedule” page.
5. Verify that audit records are generated for the successful configuration of this function.

Configuration via Database:

6. Execute the database query to update the iOS MDM Sample Schedule to 10 minutes.
7. Verify that the MDM Server performs the commands on the configured schedule from Step 6.
8. Repeat Steps 6 – 7, except in Step 6, execute the database query to update the iOS MDM Sample Schedule to 20 minutes.

Test Results:	The evaluator verified that an administrator is able to configure the periodicity for the assigned list of commands to the agent for several configured time periods and verified that the MDM Server performs the commands schedule. – PASS
Execution Method:	Manual

064	[MDMPP]FMT_SMF.1.1(2)/IOS – Specification of Management Functions (Server Configuration of Server) – Test 3
Test Objective:	The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: Test 3: (conditional) The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the MDM Server.
Configure the privacy-sensitive information that will and will not be collected from particular mobile devices **This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID Test 3 (Test Case 61).**	
Configure the interaction between TOE components 1. Add the iOS device to the Device Enrollment Program (DEP) list via the Apple Business Manager portal. 2. Authenticate to the Admin Console as an Administrator. 3. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Apple” > “Device Enrollment Program”. 4. Select “Fetch All Devices”.	
Configure server administrator login session timeout **This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID Test 3 (Test Case 61).**	
Configure enterprise certificate to be used for signing policies **This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID Test 3 (Test Case 61).**	
Configure MDM Agent/platform to perform a network reachability test **This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/IOS Test 2 (Test Case 063).**	
Configure transfer of MDM server logs to another server for storage, analysis, and reporting **This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(2)/ANDROID Test 3 (Test Case 61).**	
Test Results:	The evaluator performed this test by performing a fetch operation on the MDM server to ensure that the interaction between the TOE components was performed by validating that the set of iOS/iPadOS mobile devices intended for enrollment, were successfully registered and deemed as such by the MDM server. – PASS
Execution Method:	Manual

<table>
<tr><th>065</th><th>[MDMPP]FMT_SMF.1.1(3) – Specification of Management Functions (MAS Server)</th></tr>
<tr><td><b>Test Objective:</b></td><td>The evaluator shall ensure that the MAS client can only access the applications specified for the group they are enrolled in. The evaluator shall create a user group, making sure that the MAS client user is excluded from the group. Verify that an application accessible to that group cannot be accessed. The evaluator shall include the MAS client user in the group and assure that the application can be accessed.</td></tr>
<tr><td colspan="2">1. On the mobile device, launch the MDM Agent > “App Catalog”.
2. Verify that the application from Setup step 17 is not available via the MAS client.
3. Navigate to “Groups & Settings” > “Groups” > “Assignment Groups”.
4. Select the pencil icon next to the Assignment Group created in Setup step 13.
5. Uncheck the user from “Exclusions” -> “Excluded Users”, click “Next”, and “Publish”.
6. On the mobile device, launch the MDM Agent > “App Catalog”.
7. Verify that the application from Setup step 17 is now available via the MAS client.</td></tr>
<tr><td><b>Test Results:</b></td><td>The evaluator performed this test by defining a user group with a specific user assigned to it. The evaluator then defined an assignment group containing a smart group that included the user group defined previously. The evalutor marked the specific user from the user group as an exception to this assignment group. The evaluator then associated a specific application from the MAS server to the assignment group. Using a mobile device that was previously enrolled using the specific user defined earlier, the evaluator attempted to access the MDM / MAS App Catalog and observed that the application was not available via the MAS client. The evaluator then removed the user exclusion rule from the assignment group that was created earlier. Next, the evaluator re-launched the MDM / MAS App Catalog and observed that the application was accessible via the MAS client. – PASS</td></tr>
<tr><td><b>Execution Method:</b></td><td>Manual</td></tr>
</table>

<table>
<tr><th>066</th><th>[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 1</th></tr>
<tr><td><b>Test Objective:</b></td><td>Test 1: In conjunction with the evaluation activities in the Base-PP, the evaluator shall attempt to configure each administrator-provided management function and shall verify that the mobile device executes the commands and enforces the policies.</td></tr>
<tr><td colspan="2"><b>This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(1)/ANDROID Test 1 (Test Case 057) and [MDMPP]FMT_SMF.1.1(1)/IOS Test 1 (Test Case 058).</b>

<b>Configure whether users can unenroll from management</b>

1. Refer to the Setup in Test Case 073A, 073I, 073P to configure whether users can unenroll from management (prevent unenrollment by a user).
2. For Android, perform the following steps to enable unenrollment by a user:
a. Authenticate to the MDM Console.
b. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Android” > “Intelligent Hub Settings”.
c. Choose “DISABLED” for “Block User Unenrollment”.
d. Click “SAVE”.

3. For iOS, perform the following steps to enable unenrollment by a user:
a. Authenticate to the MDM Console.
b. Navigate to “Groups & Settings” > “All Settings” > “Devices & Users” > “Apple” > “Device Enrollment Program”.</td></tr>
</table>

c. Click “DISABLE”.
d. Enter the Security PIN.

Test Results:	This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(1)/ANDROID Test 1 (Test Case 057) and [MDMPP]FMT_SMF.1.1(1)/IOS Test 1 (Test Case 058). The evaluator was able to configure whether users can unenroll from management. – PASS
Execution Method:	Manual

067	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 2 – TD0491
Test Objective:	Test 2: The evaluator shall configure the MDM Agent authentication certificate in accordance with the configuration guidance. The evaluator shall verify that the MDM Agent uses this certificate in performing the tests for FPT_ITT.1(2) (MDM as Base-PP) or FTP_ITC_EXT.1(2) (MDF as Base-PP).

1. Perform the test steps in [MDMPP]FPT_ITT.1.1(2) – Test Case 075.
2. Inspect the packet capture and review it to ensure the mobile device (TLS client) client certificate Common Name (CN) identifier corresponds to the UDID assigned to the mobile device.

Test Results:	The evaluator performed this test in conjunction with the test assurance activity in [MDMPP]FPT_ITT.1.1(2) – Test Case 075. The evaluator then inspected the packet capture and confirmed that the mobile devices' TLS client certificate common name identifier corresponded to the UDID assigned to the mobile device by the MDM server. - PASS
Execution Method:	Manual

068	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 3
Test Objective:	Test 3: In conjunction with other evaluation activities, the evaluator shall attempt to enroll the MDM Agent in management with each interface identified in the TSS, and verify that the MDM Agent can manage the device and communicate with the MDM Server.

This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(1)/ANDROID Test 1 (Test Case 057) and [MDMPP]FMT_SMF.1.1(1)/IOS Test 1 (Test Case 058).

Test Results:	This Assurance Activity is satisfied by testing performed in [MDMPP]FMT_SMF.1.1(1)/ANDROID Test 1 (Test Case 057) and [MDMPP]FMT_SMF.1.1(1)/IOS Test 1 (Test Case 058). – PASS
Execution Method:	Manual

069	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 4
Test Objective:	Test 4: [conditional] In conjunction with the evaluation activity for FAU_ALT_EXT.2.1, the evaluator shall configure the periodicity for reachability events for several configured time periods and shall verify that the MDM Server receives alerts on that schedule.

Android:

This is performed as part of testing in [MDMPP] FMT_SMF.1.1(2)/ANDROID – Test Case 060 and [MDMPP] FMT_SMF.1.1(2)/ANDROID – Test Case 061: “Configure MDM Agent/platform to perform a network reachability test”

iOS:

This is performed as part of testing in [MDMPP] FMT_SMF.1.1(2)/IOS – Test Case 063.	
Test Results:	Android: This is performed as part of testing in [MDMPP] FMT_SMF.1.1(2)/ANDROID – Test Case 060 and [MDMPP] FMT_SMF.1.1(2)/ANDROID – Test Case 061: “Configure MDM Agent/platform to perform a network reachability test” iOS: This is performed as part of testing in [MDMPP] FMT_SMF.1.1(2)/IOS – Test Case 063. - PASS
Execution Method:	Manual

070A	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 5
Test Objective:	Test 5: [conditional] The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the mobile device.
N/A – The Security Target does not specify an assigned function for this SFR.	
Test Results:	N/A
Execution Method:	N/A

070I	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 5
Test Objective:	Test 5: [conditional] The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the mobile device.
N/A – The Security Target does not specify an assigned function for this SFR.	
Test Results:	N/A
Execution Method:	N/A

070P	[AGENTMOD]FMT_SMF_EXT.4 – Specification of Management Functions – Test 5
Test Objective:	Test 5: [conditional] The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the mobile device.
N/A – The Security Target does not specify an assigned function for this SFR.	
Test Results:	N/A
Execution Method:	N/A

071	[MDMPP]FMT_SMR.1.2(1) – Security Management Roles
Test Objective:	In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or HTTPS then both methods of administration must be exercised during the evaluation team's test activities.
In order to manage the administrative functions, the TOE is administered through a TLS/HTTPS	

web interface for both the administrator (MDM Server console) and the user (MDM Server Self-Service Portal). All testing activities that involve configuration, such as MDM policies and profiles, are performed throughout the evaluation using the TOE TLS/HTTPS web interface (e.g. [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058).	
Test Results:	In order to manage the administrative functions, the TOE is administered through a TLS/HTTPS web interface for both the administrator (MDM Server console) and the user (MDM Server Self-Service Portal). All testing activities that involve configuration, such as MDM policies and profiles, are performed throughout the evaluation using the TOE TLS/HTTPS web interface (e.g. [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058). – PASS
Execution Method:	Manual

072	[MDMPP]FMT_SMR.1.2(2) – Security Management Roles (MAS Server)
Test Objective:	In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or HTTPS then both methods of administration must be exercised during the evaluation team's test activities.
According to the Security Target for [MDMPP]FMT_SMR.1(2): **"[SERVER] The MAS Server is logically integrated with the UEM Server. It is accessed by Administrators using the Apps & Books > Applications > Native Apps tab in the Admin Console. Since this is not accessed separately from the remainder of the UEM Server capabilities, the administrative roles that can interact with the MAS Server are defined in the same manner as for FMT_SMR.1(1) above. The UEM Server also maintains the roles of enrolled mobile devices and application access groups."** **In order to manage the administrative functions, the TOE is administered through a TLS/HTTPS web interface for both the administrator (MDM Server console) and the user (MDM Server Self-Service Portal). All testing activities that involve configuration, such as MDM policies and profiles, are performed throughout the evaluation using the TOE TLS/HTTPS web interface (e.g. [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058).**	

Test Results:	According to the Security Target for [MDMPP]FMT_SMR.1(2): "[SERVER] The MAS Server is logically integrated with the UEM Server. It is accessed by Administrators using the Resources > Apps > Native Apps tab in the Admin Console. Since this is not accessed separately from the remainder of the UEM Server capabilities, the administrative roles that can interact with the MAS Server are defined in the same manner as for FMT_SMR.1(1) above. The UEM Server also maintains the roles of enrolled mobile devices and application access groups." In order to manage the administrative functions, the TOE is administered through a TLS/HTTPS web interface for both the administrator (MDM Server console) and the user (MDM Server Self-Service Portal). All testing activities that involve configuration, such as MDM policies and profiles, are performed throughout the evaluation using the TOE TLS/HTTPS web interface (e.g. [MDMPP]FMT_SMF.1.1(1)/ANDROID – Test Case 057 and [MDMPP]FMT_SMF.1.1(1)/IOS – Test Case 058). - PASS
Execution Method:	Manual

073A	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 1
Test Objective:	Test 1: If 'prevent the unenrollment from occurring' is selected: The evaluator shall configure the Agent according to the administrative guidance for each available configuration interface, shall attempt to unenroll the device, and shall verify that the attempt fails.
1. On the Android mobile device, navigate to "Settings" > "Biometrics and security" > "Other security settings" > "Device admin apps" > "Hub". 2. Tap "Deactivate". 3. Observe that "Deactivate" is not possible to tap or is unavailable. 4. On the Android mobile device, launch the Hub MDM Agent app. 5. Tap "This Device" > "Enrollment". 6. Verify that the "Unenroll Device" button is absent from the interface. 7. Attempt to uninstall the Hub MDM Agent app from the mobile device. 8. Verify that the uninstall is unsuccessful. 9. Observe that there is no unenrollment option.	
Test Results:	The evaluator successfully configured the Agent according to the administrative guidance for each available configuration interface and attempted to unenroll the device. The evaluator was able to verify that the attempts failed. – PASS
Execution Method:	Manual

073I	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 1
Test Objective:	Test 1: If 'prevent the unenrollment from occurring' is selected: The evaluator shall configure the Agent according to the administrative guidance for each available configuration interface, shall attempt to unenroll the device, and shall verify that the attempt fails.
1. Wait 30 days from when the iOS device was added to the Apple DEP portal. 2. On the iOS mobile device, navigate to "Settings" > "General" > "Device Management". 3. Tap "Device Manager". 4. Observe that "Remove Management" is absent.	

5. Verify that the mobile device is still enrolled in mobile device management by repeating Step 2 and observing that "Device Manager" is listed under "Mobile Device Management".	
Test Results:	The evaluator successfully configured the Agent according to the administrative guidance for each available configuration interface and attempted to unenroll the device. The evaluator was able to verify that the attempts failed. – PASS
Execution Method:	Manual

073P	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 1
Test Objective:	Test 1: If 'prevent the unenrollment from occurring' is selected: The evaluator shall configure the Agent according to the administrative guidance for each available configuration interface, shall attempt to unenroll the device, and shall verify that the attempt fails.
1. Wait 30 days from when the iOS device was added to the Apple DEP portal. 2. On the iOS mobile device, navigate to "Settings" > "General" > "Device Management". 3. Tap "Device Manager". 4. Observe that "Remove Management" is absent. 5. Verify that the mobile device is still enrolled in mobile device management by repeating Step 2 and observing that "Device Manager" is listed under "Mobile Device Management".	
Test Results:	The evaluator successfully configured the Agent according to the administrative guidance for each available configuration interface and attempted to unenroll the device. The evaluator was able to verify that the attempts failed. – PASS
Execution Method:	Manual

074A	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 2
Test Objective:	Test 2: If 'apply remediation actions' is selected: If any configuration allows the user to unenroll, the evaluator shall configure the Agent to allow user unenrollment, attempt to unenroll, and verify that the remediation actions are applied.
The ST does not select "apply remediation actions"; therefore, this test assurance activity does not apply.	
Test Results:	N/A
Execution Method:	N/A

074I	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 2
Test Objective:	Test 2: If 'apply remediation actions' is selected: If any configuration allows the user to unenroll, the evaluator shall configure the Agent to allow user unenrollment, attempt to unenroll, and verify that the remediation actions are applied.
The ST does not select "apply remediation actions"; therefore, this test assurance activity does not apply.	
Test Results:	N/A
Execution Method:	N/A

074P	[AGENTMOD]FMT_UNR_EXT.1 – User Unenrollment Prevention – Test 2

Test Objective:	Test 2: If 'apply remediation actions' is selected: If any configuration allows the user to unenroll, the evaluator shall configure the Agent to allow user unenrollment, attempt to unenroll, and verify that the remediation actions are applied.
The ST does not select "apply remediation actions"; therefore, this test assurance activity does not apply.	
Test Results:	N/A
Execution Method:	N/A

4.3.6 Protection of the TSF

075	[MDMPP]FPT_ITT.1.1(2) – Internal TOE TSF Data Transfer (MDM Agent) – Test 1
Test Objective:	Test 1: The evaluator shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. Further evaluation activities are associated with the specific protocols.
Repeat this test for each claimed MDM Agent in the ST: 1. Launch Wireshark on the MDM Server and begin capturing packets between the MDM Server and MDM Agent. 2. Initiate communication between the MDM Server and the MDM Agent device. 3. Stop capturing packets from the MDM Server. 4. Communication between the MDM Server and the MDM Agent device is successful. 5. Inspect the packet capture and review it to ensure the data is encrypted using TLS v1.2.	
Test Results:	The evaluator performed this test, for each claimed MDM agent in the ST, by first capturing packets between the MDM server and the MDM agent. Then, the evaluator stimulated the TOE such that communication between the MDM server and MDM agent was initiated. The evaluator terminated the packet capture and then inspected the packet capture and confirmed that communication was successful between the MDM server and MDM agent. Finally, the evaluator observed that the communication was encrypted using TLS v1.2. – PASS
Execution Method:	Manual

076	[MDMPP]FPT_ITT.1.1(2) – Internal TOE TSF Data Transfer (MDM Agent) – Test 2
Test Objective:	Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
This test is performed in [MDMPP]FPT_ITT.1.1(2) – Test Case 075.	
Test Results:	This test is performed in [MDMPP]FPT_ITT.1.1(2) – Test Case 075. – PASS
Execution Method:	Manual

077	[MDMPP]FPT_TST_EXT.1.2 – Functionality Testing – Test 1
Test Objective:	The evaluator shall perform the following tests: Test 1: The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful.

1. Perform an integrity check on all executables using signtool.exe.
2. Execute the following command:

   ```
   cd "C:\AirWatch\AirWatch 2209"

   for /R %v in (*.dll *.exe) do signtool.exe verify /a /pa /v "%v" >> signtool_output.txt 2>&1
   ```

3. Verify that the signtool.exe application returns with “Successfully verified: <filename>” for each *.dll and *.exe file.

Test Results:	The evaluator performed this test by validating the digital signature of TSF executables using a platform based tool (signtool.exe). The evaluator performed this validation against all TSF executables and observed that each was successfully verified. – PASS
Execution Method:	Manual

078	[MDMPP]FPT_TST_EXT.1.2 – Functionality Testing – Test 2
Test Objective:	The evaluator shall perform the following tests: Test 2: The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.

1. If any UEM services are running, stop the UEM services.
2. Create a backup copy of the main UEM executable.
3. Using a hex editor, open the main UEM executable and modify some of the data.
4. Save the modified executable.
5. Perform an integrity check on the executable using signtool.exe.
6. Execute the following command:

   ```
   signtool.exe verify /a /pa /v <executable>
   ```

7. Verify that the signtool.exe application returns with “The digital signature of the object did not verify”.

Test Results:	The evaluator performed this test by validating the digital signature of TSF executables using a platform based tool (signtool.exe). The evaluator performed this validation against a modified TSF executable and observed that it failed the verification check. – PASS
Execution Method:	Manual

079	[MDMPP]FPT_TUD_EXT.1.1 – Trusted Update
Test Objective:	The evaluator shall query the TSF for the current version of the software according to the AGD guidance and shall verify that the current version matches that of the documented and installed version.

MDM Server:

1. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch

2. Click on “About VMware AirWatch” to access the version information.
3. Verify the TOE version with that of the one in the guidance documentation.

MDM Server method to obtain MDM Agent (Android) version:

1. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
2. Navigate to “Devices” > “List View”.
3. Click on the enrolled mobile device details view.
4. Navigate to “More” > “Custom Attributes”.
5. Verify the TOE MDM Agent version value for Application “com.airwatch.androidagent.identity.xml”, Attribute “identity.agentVersion”.

MDM Server method to obtain MDM Agent (iOS) version:

1. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
2. Navigate to “Devices” > “List View”.
3. Click on the enrolled mobile device details view.
4. Click on “Apps” tab.
5. Verify the TOE MDM Agent version.

MDM Server method to obtain MDM Agent (iPadOS) version:

1. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
2. Navigate to “Devices” > “List View”.
3. Click on the enrolled mobile device details view.
4. Click on “Apps” tab.
5. Verify the TOE MDM Agent version.

MDM Agent (Android):

1. Query the current version of the MDM Agent (Android):
   a. Launch the MDM Agent.
   b. Tap “About”.

MDM Agent (iOS):

1. Query the current version of the MDM Agent (iOS):
   a. Launch the Hub MDM Agent.
   b. Tap “About”.

MDM Agent (iPadOS):

1. Query the current version of the MDM Agent (iOS):
   a. Launch the Hub MDM Agent.
   b. Tap “About”.

Test Results:	For each TOE component, the evaluator queried the TSF for the current version of the software and successfully validated that the current version matches that of the documented and installed version. – PASS
Execution Method:	Manual

080	[MDMPP]FPT_TUD_EXT.1.3 – Trusted Update – Test 1
Test Objective:	The evaluator shall perform the following tests: Test 1: The evaluator shall attempt to initiate an update digitally signed by the vendor and verify that the update is successfully installed.

MDM Server:

1. Record the current version of the MDM Server:
    a. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
    b. Click on “About VMware AirWatch” to access the version information.
2. Attempt to install an update that is digitally signed by VMware.
3. Verify that the update installation succeeded.
4. Verify that the version number increased:
    a. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
    b. Click on “About VMware AirWatch” to access the version information.
    c. Verify the version number increased.

MDM Agent (Android):

1. Record the current version of the MDM Agent (Android):
    a. Launch the MDM Agent.
    b. Tap “About”.
2. Attempt to install an update that is digitally signed by VMware.
3. Verify that the update installation succeeded.
4. Verify that the version number increased:
    a. Launch the MDM Agent.
    b. Tap “About”.

MDM Agent (iOS):

1. Record the current version of the MDM Agent (iOS):
    a. Launch the iOS Hub MDM Agent.
    b. Tap “About”.
2. Attempt to install an update that is digitally signed by VMware.
3. Verify that the update installation succeeded.
4. Verify that the version number increased.

MDM Agent (iPadOS):

1. Record the current version of the MDM Agent (iOS):
    a. Launch the iOS Hub MDM Agent.
    b. Tap “About”.
2. Attempt to install an update that is digitally signed by VMware.
3. Verify that the update installation succeeded.
4. Verify that the version number increased.

Test Results:	- PASS
Execution Method:	Manual

081	[MDMPP]FPT_TUD_EXT.1.3 – Trusted Update – Test 2
Test Objective:	The evaluator shall perform the following tests: Test 2: The evaluator shall attempt to install an update not digitally signed by the vendor and verify that either the signature can be checked (allowing the update to be aborted) or the update is not installed.

MDM Server:

1. Record the current version of the MDM Server:
    a. Navigate to the MDM Server Console: https://uem.cctl.company.com/AirWatch
    b. Click on "About VMware AirWatch" to access the version information.
2. Attempt to install an update that is not digitally signed by VMware.
3. Stop the TOE MDM Server services by executing the following command on the MDM Server:

    ```
    iisreset -stop
    ```

4. Create a backup of the existing "AirWatchFoundation.dll" file.
5. Manually copy and overwrite the existing "AirWatchFoundation.dll" with the unsigned version into the AirWatch bin directory.
6. Start the MDM Server services by executing the following command on the MDM Server:

    ```
    iisreset -start
    ```

7. Verify that the update failed to install.
8. Restore the backed up "AirWatchFoundation.dll" file from Step 3.
9. Repeat Step 6.
10. Repeat Step 1 and verify that the version number did not change.

MDM Agent (Android):

1. Record the current version of the MDM Agent (Android):
    a. Launch the MDM Agent.
    b. Tap "About".
2. Attempt to install an update that is not digitally signed by VMware.
3. Verify that the update failed to install.
4. Repeat Step 1 and verify the version did not change.

MDM Agent (iOS):

1. Record the current version of the MDM Agent (iOS):
    a. Launch the iOS Hub MDM Agent.
    b. Tap "About".
2. Attempt to install an update that is not digitally signed by VMware.
3. Verify that the update installation failed.
4. Repeat Step 1 and verify the version did not change.

MDM Agent (iPadOS): 1. Record the current version of the MDM Agent (iOS): a. Launch the iOS Hub MDM Agent. b. Tap "About". 2. Attempt to install an update that is not digitally signed by VMware. 3. Verify that the update installation failed. 4. Repeat Step 1 and verify the version did not change.	
Test Results:	MDM server: The evaluator performed this test by first recording the current version of the MDM server as reported by the MDM server. The evaluator then attempted to install an update to the MDM server that was not digitally signed by the developer. The evaluator confirmed that the update failed to install via observation of the failure of the application to start, confirmation that the version number did not change, and via operating system generated audit records for the failure of the signature verification. MDM agent (Android, iOS, iPadOS): For each of the referenced platforms, the evaluator performed this test by first recording the current version of the MDM agent software as reported by the MDM agent. The evaluator then attempted to install an update to the MDM agent software that was not digitally signed by the developer. The evaluator observed that the installation of the update was unsuccessful, that the version number remained the same, and audit records were generated by the operating system for the failure of the signature verification. - PASS
Execution Method:	Manual

4.3.7 TOE Access

082	[MDMPP]FTA_TAB.1.1 – Default TOE Access Banners
Test Objective:	The evaluator shall also perform the following test: The evaluator shall start up or unlock the TSF. The evaluator shall verify that the notice and consent warning message is displayed in each instance described in the TSS.
MDM Server Console: 1. Authenticate to the MDM Server Console: a. Navigate to: https://uem.cctl.company.com/AirWatch b. Verify that the notice and consent warning message is displayed. ***MDM Self-Service Portal:*** 1. Authenticate to the MDM Self-Service Portal: a. Navigate to: https://uem.cctl.company.com/MyDevice	

2. Verify that the notice and consent warning message is displayed.	
Test Results:	The evaluator performed this test by navigating to the TOE (MDM server) web administration UI and self-service portal and then verified that a notice and consent warning message was displayed prior to authentication. – PASS
Execution Method:	Manual

4.3.8 Trusted Path/Channels

083	[MDMPP]FTP_ITC_EXT.1.1 – Trusted Channel
Test Objective:	This testing can be completed in conjunction with the testing for FPT_ITT.1(1)/FPT_ITT.1(2), FTP_ITC.1(2) or FTP_ITC.1(3).
N/A	
Test Results:	N/A
Execution Method:	N/A

084	[MDMPP]FTP_ITC.1.3(1) – Inter-TSF Trusted Channel (Authorized IT Entities) – Test 1
Test Objective:	Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. Further evaluation activities are associated with the specific protocols.
MDM Server – Syslog Audit Server: 1. Using Wireshark, begin capturing packets between the MDM Server and the syslog audit server. 2. Perform some activity on the MDM Server that causes audit data to be transmitted to the remote syslog audit server. 3. Stop capturing packets between the MDM Server and the syslog audit server. 4. Inspect the packet capture and ensure that the communications are successful and encrypted with TLS v1.2. ***MDM Server – AD/LDAP Server:*** 1. Using Wireshark, begin capturing packets between the MDM Server and the AD/LDAP server. 2. Authenticate to the MDM Server using AD/LDAP authentication. 3. Stop capturing packets between the MDM Server and the AD/LDAP server. 4. Inspect the packet capture and ensure that the communications are successful and encrypted with TLS v1.2.	
Test Results:	For each inter-TSF trusted channel, the evaluator performed this test by capturing packets between the MDM server and the remote endpoint (remote audit server; AD/LDAP server). Then, the evaluator stimulated the MDM server such that it initiated communication with the remote endpoint. Next, the evaluator terminated the packet capture. Finally, the evaluator inspected the captured packets and validated that communication was successful and encrypted using TLS v1.2. – PASS
Execution Method:	Manual

085	[MDMPP]FTP_ITC.1.3(1) – Inter-TSF Trusted Channel (Authorized IT Entities) – Test 2
Test Objective:	Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
This test assurance activity is tested in [MDMPP]FTP_ITC.1.3(1) – Test Case 084.	
Test Results:	This test assurance activity is tested in [MDMPP]FTP_ITC.1.3(1) – Test Case 084. – PASS
Execution Method:	Manual

086	[MDMPP]FTP_ITC.1.3(1) – Inter-TSF Trusted Channel (Authorized IT Entities) – Test 3
Test Objective:	Test 3: The evaluator shall ensure, for each communication channel with the MDM Server, that a protocol analyzer identifies the traffic as the protocol under testing. Further evaluation activities are associated with the specific protocols.
This test assurance activity is tested in [MDMPP]FTP_ITC.1.3(1) – Test Case 084.	
Test Results:	This test assurance activity is tested in [MDMPP]FTP_ITC.1.3(1) – Test Case 084. – PASS
Execution Method:	Manual

087	[MDMPP]FTP_TRP.1.3(1) – Trusted Path (for Remote Administration) – Test 1
Test Objective:	The evaluator shall also perform the following tests: Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. Further evaluation activities are associated with the specific protocols.
1. Using Wireshark, begin capturing packets between the test machine web browser and the MDM Server. 2. Using the web browser, navigate to the MDM web administration console URL: https://uem.cctl.company.com/AirWatch 3. Authenticate to the MDM Web Administration Console. 4. Stop capturing packets between the web browser and the MDM Server. 5. Inspect the packet capture and ensure that the communication is successful, and that the data are encrypted.	

Test Results:	The evaluator performed this test by first capturing packets between the MDM server and the test machine used for remote administration of the TOE (MDM server). Next, the evaluator navigated to the MDM server's web administration console URL and authenticated to it. The evaluator terminated the packet capture. Finally, the evaluator inspected the packet capture and validated that communication was successful and that the data are encrypted. – PASS
Execution Method:	Manual

088	[MDMPP]FTP_TRP.1.3(1) – Trusted Path (for Remote Administration) – Test 2
Test Objective:	The evaluator shall also perform the following tests: Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish remote administrative sessions without invoking the trusted path. Further evaluation activities are associated with the specific protocols.

1. Execute the following command from test machine administrator command prompt to perform a comprehensive TCP port scan against the TOE:

   ```
   nmap -sS -sV -p 0-65535 -O -T4 -A -v -oN MDM_Server_TCP.txt 10.137.2.36
   ```

2. Execute the following command from the test machine administrator command prompt to perform a comprehensive UDP port scan against the TOE:

   ```
   nmap -sU -sV -p 0-65535 -max-rtt-timeout 25ms --max-retries 2 --max-scan-delay 10ms --min-hostgroup 32 --version-intensity 0 -O -T4 -A -v -oN MDM_Server_UDP.txt 10.137.2.36
   ```

3. Execute the following command on the MDM Server underlying Windows operating system administrator command prompt:

   ```
   netstat -ano >> netstat.txt
   tasklist >> tasklist.txt
   ```

4. Review the results of the TCP and UDP port scans against the results from Step 3 and verify that port 8443 and 443 is listening and running a HTTPS web server for the MDM Server process and that there are no other ports or services that allow a remote administrative session without invoking the trusted path.

Test Results:	The evaluator was able to verify that no available interface can be used by a remote user to establish remote administrative sessions without invoking the trusted path. – PASS
Execution Method:	Manual

089	[MDMPP]FTP_TRP.1.3(1) – Trusted Path (for Remote Administration) – Test 3
Test Objective:	The evaluator shall also perform the following tests: Test 3: The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
This test assurance activity is tested in [MDMPP]FTP_TRP.1.3(1) – Test Case 087.	

Test Results:	This test assurance activity is tested in [MDMPP]FTP_TRP.1.3(1) – Test Case 087. – PASS
Execution Method:	Manual

090	[MDMPP]FTP_TRP.1.3(2) – Trusted Path (for Enrollment) – Test 1
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) enrollment method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. Further evaluation activities are associated with the specific protocols.

MDM Agent (Android/iOS/iPadOS) to MDM Server:

1. Using Wireshark, begin capturing packets between the MDM Server and MDM Agent.
2. Enroll a mobile device into MDM.
3. Stop capturing packets between the MDM Server and the MDM Agent.
4. Inspect the packet capture and ensure that the communication data are successful and encrypted.

Test Machine to MDM Self-Service Portal Web UI:

1. Using Wireshark, begin capturing packets between the test machine and MDM self-service portal.
2. Navigate to the MDM self-service portal:

   https://uem.cctl.company.com/MyDevice

3. Authenticate to the MDM self-service portal.
4. Stop capturing packets between the test machine and the MDM self-service portal.
5. Inspect the packet capture and ensure that the communication data are successful and encrypted.

Test Results:	For each MDM agent / platform, the evaluator captured packets between the MDM server and MDM agent. Next, the evaluator enrolled the mobile device into the MDM server. The evaluator then terminated the packet capture. Finally, the evaluator inspected the packet capture and validated that the communication was successful and encrypted. In addition, the test machine to MDM self-service portal web UI trusted path was evaluated as part of this test. For this, the evaluator captured packets between the test machine and the MDM self-service portal. Next, using the test machine, the evaluator navigated to the MDM self-service portal URL and authenticated to it. The evaluator terminated the packet capture. Finally, the evaluator inspected the packet capture and validated that the communication was successful and encrypted. – PASS
Execution Method:	Manual

091	[MDMPP]FTP_TRP.1.3(2) – Trusted Path (for Enrollment) – Test 2
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 2: For each method of enrollment supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish enrollment sessions without invoking the trusted path. Further evaluation activities are associated with the specific protocols.
MDM Server (Self-Service Portal): **Refer to [MDMPP]FTP_TRP.1.3(1) – Trusted Path (for Remote Administration) – Test 2 – Test Case 088, which demonstrates that there are no other available interfaces other than the ones described below for enrollment into management.** **Android:** **Refer to [MDMPP]FIA_ENR_EXT.1.1/ANDROID – Test Case 028, which tests the MDM Server's ability to enroll Android mobile devices.** **iOS/iPadOS:** **Refer to [MDMPP]FIA_ENR_EXT.1.1/IOS – Test Case 031, which tests the MDM Server's ability to enroll iOS/iPadOS mobile devices.**	
Test Results:	MDM Server (Self-Service Portal): Refer to [MDMPP]FTP_TRP.1.3(1) – Trusted Path (for Remote Administration) – Test 2 – Test Case 088, which demonstrates that there are no other available interfaces other than the ones described below for enrollment into management. Android: Refer to [MDMPP]FIA_ENR_EXT.1.1/ANDROID – Test Case 028, which tests the MDM Server's ability to enroll Android mobile devices. iOS/iPadOS: Refer to [MDMPP]FIA_ENR_EXT.1.1/IOS – Test Case 031, which tests the MDM Server's ability to enroll iOS/iPadOS mobile devices. - PASS
Execution Method:	Manual

092	[MDMPP]FTP_TRP.1.3(2) – Trusted Path (for Enrollment) – Test 3
Test Objective:	For each MDM Agent/platform listed as supported in the ST: Test 3: The evaluator shall ensure, for each method enrollment, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
This test assurance activity is tested in "[MDMPP] FTP_TRP.1.3(2) – Test Case 090".	
Test Results:	This test assurance activity is tested in "[MDMPP] FTP_TRP.1.3(2) – Test Case 090". – PASS
Execution Method:	Manual

5 Evaluation Activities for SARs

This section addresses assurance activities that are defined in the *Protection Profile for Mobile Device Management Version 4.0* [MDMPP] that correspond with Security Assurance Requirements. The *PP-Module for MDM Agent Version 1.0 [AGENTMOD]* does not define any SARs beyond those defined within the base-PP to which it must claim conformance.

AGD_OPE.1 – "*Some of the contents of the operational guidance will be verified by the evaluation activities in Sections 4.2, 4.3, and 4.4and evaluation of the TOE according to the CEM. The following additional information is also required.*

If cryptographic functions are provided by the TOE, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The documentation must describe the process for verifying updates to the TOE by verifying a digital signature - this may be done by the TOE or the underlying platform. The evaluator shall verify that this process includes the following steps:

Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities."

Section 6.3 of the AGD states that cryptographic services for the UEM Server are provided by the underlying Windows server platform. The Windows Server 2019 platform uses Microsoft's SymCrypt to perform all cryptographic services. Cryptographic services for the iOS and Android Hub Agents are mainly provided by the underlying mobile device platforms. The iOS Hub Agent uses Apple iOS platform's CoreCrypto Module to perform all claimed cryptographic services. The Android Hub Agent uses the Android platform's SCrypto and BoringSSL cryptographic modules to perform all claimed cryptographic services, except for the policy digital signature validation requirements. The Android Hub Agent implements OpenSSL for the specific purpose of performing the policy digital signature validation services.

Section 6.1 of the AGD contains all steps necessary to configure the cryptographic modules used by VMware Workspace ONE UEM in the evaluated configuration. There are no specific steps that are required to follow in order to configure key generation and establishment functionality; these functions are provided automatically by the underlying cryptographic modules and are specified by the specific protocols that require them.

Section 6.3 of the AGD states that the evaluation does not make any claims of cryptographic strength for any other cryptographic modules or configurations besides what is claimed in the VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target.

Sections 6.4.2 and 6.4.3 of the AGD describe the process for verifying updates to the TOE. The AGD states that verification is of updates is performed by the underlying platform with a digital signature verification process.

The AGD describes in these same sections where an update can be obtained, how to make it accessible to the TOE, initiating the update process, and how to determine whether the update was successful or unsuccessful including the verification of the update's digital signature.
Additionally, the AGD states in section 2 that any functionality that is not described in the AGD or in the VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target was not evaluated and should be exercised at the user's risk.

AGD_PRE.1 – "*As indicated in the introduction above, there are significant expectations with respect to the documentation, especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.*"

Section 5.1 of the AGD adequately addresses all platforms claimed for the TOE in the ST. The TOE components in the AGD match the TOE components defined in the ST.

ALC_CMC.1 – *"The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a website advertising the TOE, the evaluator shall examine the information on the website to ensure that the information in the ST is sufficient to distinguish the product."*

The evaluation team verified that the Security Target (ST), TOE, and Supplemental Administrative Guidance (AGD) were labeled consistently to correctly identify the operational environment and TOE software versions in the CC evaluation.

Specifically, Section 1.2 of the ST states in the TOE Reference that the TOE is the "The TOE is the VMware Workspace ONE Unified Endpoint Management version 2209 comprising of the Unified Endpoint Management Server and one or more VMware Intelligent Hub Agents installed on Apple and Android devices." Section 5.1 of the AGD states that the TOE is the VMware Workspace ONE Unified Endpoint Management version 2209 that includes the TOE components described in Table 1 of the AGD. Table 1 of the AGD lists the TOE components, which matches the TOE components listed in section 2.1 of the ST. Finally, the product web site, https://www.vmware.com/products/workspace-one/unified-endpoint-management.html, contains identifying product information including a whitepaper, infographic, and description of "Workspace ONE Unified Endpoint Management (UEM)". The ST states in the TOE Overview that "The TOE is a Mobile Device Management product and is comprised of an MDM Server component (UEM Server) and one or more agent components called VMware Intelligent Hubs (iOS Hub agent and Android Hub agent). In the evaluated configuration of the TOE, the UEM Server is deployed in an on-premises configuration. The UEM Server component provides a centralized enterprise level management capability for a collection of mobile devices running the iOS and Android Hub agents. The UEM Server is also a Mobile Application Store (MAS) Server that allows managed devices to download apps from a trusted repository that resides within the organization managing the device."

All of this information as stated above provides sufficient context to accurately identify the TOE as such in the ST, AGD, and vendor web site.

ALC_CMS.1 – *"The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the evaluation activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.*

Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation.

The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification."

The evaluation team confirmed the TOE had a unique identifier under ALC_CMC.1 which is VMware Workspace ONE Unified Endpoint Management Version 2209. This included a review of the TSF vendor's website to determine that the identifier was enough to distinguish the TOE from other products from the TSF vendor. The evaluation team also reviewed the following documentation provided by the vendor and confirmed that this identifier was consistently used to reference the TOE:

- VMware Workspace ONE Unified Endpoint Management Version 2209 Security Target, Version 1.0
- VMware Workspace ONE Unified Endpoint Management Version 2209 Supplemental Administrative Guidance Version 1.0
- Installing Workspace ONE UEM for on-premises and SaaS deployments VMware Workspace ONE UEM 2209
- Upgrade Guide for on-premises and SaaS deployments VMware Workspace ONE UEM 2209
- Console Basics VMware Workspace ONE UEM 2209
- Directory Service Integration VMware Workspace ONE UEM 2209
- Certificate Authority Integrations VMware Workspace ONE UEM 2209
- Integration with Apple Business Manager VMware Workspace ONE UEM 2209

AVA_VAN.1 – *"As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated."*

The evaluation team reviewed vendor documentation, formulated hypotheses, performed vulnerability analysis, and documented the hypotheses and analysis in accordance with the MDMPP and AGENTMOD requirements. Keywords were identified based upon review of the Security Target and AGD. The following keywords were identified:

Workspace ONE	This is a generic term for searching for known vulnerabilities for the overall TOE product.
Workspace ONE Console	This is a generic term for searching for known vulnerabilities for the Server/Console component of TOE
Workspace ONE hub	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE
Workspace ONE agent	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE focusing on previous naming convention of Agent.
Workspace ONE UEM	This is a generic term for searching for known vulnerabilities for the overall TOE product.
Workspace ONE UEM console	This is a generic term for searching for known vulnerabilities for the Server/Console component of TOE
Workspace ONE UEM hub	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE
Workspace ONE UEM agent	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE focusing on previous naming convention of Agent.
Workspace ONE Unified Endpoint Management	This is a generic term for searching for known vulnerabilities for the overall TOE product.
Workspace ONE Unified Endpoint Management console	This is a generic term for searching for known vulnerabilities for the Server/Console component of TOE
Workspace ONE Unified Endpoint Management hub	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE
Workspace ONE Unified Endpoint Management agent	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE focusing on previous naming convention of Agent.
Unified Endpoint Management	This is a generic term for searching for known vulnerabilities for the overall TOE product.
Unified Endpoint Management console	This is a generic term for searching for known vulnerabilities for the Server/Console component of TOE
Unified Endpoint Management hub	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE
Unified Endpoint Management agent	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE focusing on previous naming convention of Agent.
VMware UEM	This is a generic term for searching for known vulnerabilities for the overall TOE product.
VMware console	This is a generic term for searching for known vulnerabilities for the Server/Console component of TOE
VMware Hub	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE
VMware agent	This is a generic term for searching for known vulnerabilities for the Intelligent Hub component of TOE focusing on previous naming convention of Agent.
intelligent hub	Generic term without VMware specifics for Hub component
VMware SDK	Library
VMware Workspace ONE SDK	Library
openssl 2.0.16 (on Android Hub)	Library
Declared library list for FPT_LIB_EXT.1	Libraries

These keywords were used individually and as part of various permutations and combinations to search for vulnerabilities on public vulnerability sources (updated February 10, 2023). The following public vulnerability sources were searched:

a) NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search

b) Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
https://www.cvedetails.com/vulnerability-search.php

c) US-CERT: http://www.kb.cert.org/vuls/html/search

e) Tipping Point Zero Day Initiative http://www.zerodayinitiative.com/advisories

f) Offensive Security Exploit Database: https://www.exploit-db.com/

g) Rapid7 Vulnerability Database: https://www.rapid7.com/db/vulnerabilities

Upon the completion of the vulnerability analysis research, the team had identified several generic vulnerabilities upon which to build a test suite. These tests were created specifically with the intent of exploiting these vulnerabilities within the TOE or its configuration.

The team tested the following areas:

- Eavesdropping on Communications
  This test will attempt to intercept any TOE involved network traffic as evaluated in FPT_ITT.1(2), FTP_ITC.1(1), FTP_TRP.1(1), and FTP_TRP.1(2).
- Port Scanning
  This test will attempt to identify any way to subvert the security of the TOE by executing a side channel attack. A port scanner will be run against the TOE in an attempt to identify any open ports. Any port on a system that accepts external connections could potentially represent an attack vector. This test will identify any such ports and will attempt to enumerate them to determine their original purpose.
- Web Interface Vulnerability Identification
  This test looks for major vulnerabilities including cross-site scripting, SQL injection, directory traversal, unchecked file uploads, etc. as well as less critical vulnerabilities such as unnecessary information disclosure.

The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

6 Evaluating Additional Components for a Distributed TOE

This section addresses assurance activities that are defined in the *Protection Profile for Mobile Device Management Version 4.0* [MDMPP] that correspond with distributed TOE Requirements.

Evaluator Actions for Assessing the ST – *"The evaluator shall examine the TSS to identify any extra instances of TOE components allowed in the ST and shall examine the description of how the additional components maintain the SFRs to confirm that it is consistent with the role that the component plays in the evaluated configuration. For example: the secure channels used by the extra component for intra-TOE communications (FPT_ITT) and external communications (FTP_ITC) must be consistent, the audit information generated by the extra component must be maintained, and the management of the extra component must be consistent with that used for the original instance of the component in the minimum configuration."*

Section 8.0 of the ST describes the minimum configuration as one UEM Server, and one iOS Hub Agent installed on Apple device and/or one Android Hub Agent installed on an Android device. This section also states that additional iOS Hub Agents and Android Hub Agents can be installed on additional mobile devices of their respective platforms and these additional instances of these TOE components are consistent. The evaluation team reviewed all of Section 8 of the ST and determined that there are no instances where functionality is being described for these TOE components that would result in different SFR claims for adding one or more additional instances of these components. Therefore, the evaluation team has determined that adding additional instances of these TOE components would not impact the SFR claims made by the minimum configuration and the additional instances have consistent SFR claims to their minimum configuration equivalents.

Evaluator Actions for Assessing the Guidance Documentation – *"The evaluator shall examine the description of the extra instances of TOE components in the guidance documentation to confirm that they are consistent with those identified as allowed in the ST. This includes confirmation that the result of applying the guidance documentation to configure the extra component will leave the TOE in a state such that the claims for SFR support in each component are as described in the ST and therefore that all SFRs continue to be met when the extra components are present.*

The evaluator shall examine the secure communications described for the extra components to confirm that they are the same as described for the components in the minimum configuration (additional connections between allowed extra components and the components in the minimum configuration are allowed of course)."

Section 1 of the AGD describes the minimum configuration as one UEM Server, and one iOS Hub Agent installed on an Apple device and/or one Android Hub Agent installed on an Android device. This section also states that additional iOS Hub Agents and Android Hub Agents can be installed on additional mobile devices of their respective platforms. This is consistent with Sections 1.2 and 8 of the ST. The evaluation team reviewed the entire AGD and the description of the configuration and operation of the iOS Hub Agent and Android Hub Agent is plural throughout the document. This indicates that one or more instances of these TOE components can be included within the TOE's evaluated configuration and there would be no differences in the SFR claims compared to when there is only a single instance of these TOE components. The evaluation team has concluded that if there was one or many iOS Hub Agents the SFRs will continue to be met. The evaluation team has concluded that if there was one or many Android Hub Agents the SFRs will continue to be met.

Based upon the ST and AGD descriptions, communication between TOE components is a 'hub and spoke' topology between the UEM Server and Hub Agents. Additionally, there is no TOE communication between Hub Agents, even through the UEM Server. Section 6.1 of the AGD configures the UEM Server and its underlying platform for secure communications with the Hub Agents. Sections 6.2.6 and 6.2.7 of the AGD describe the ability to enroll Android devices and iOS devices into management. During enrollment, the Hub Agent is installed on the devices, which establishes a secure connection between the UEM Server and the Hub Agent. These sections describe the process in a manner that the procedures can be used for multiple devices and their respective Hub Agents. The evaluation team has determined that since the procedures for configuring a single iOS and Android Hub Agent is the same as multiple instances of these TOE components, the secure communications between iOS Hub Agents and the UEM Server are consistent and the secure communications between Android Hub Agents and the UEM Server are consistent.

Evaluator Actions for Testing the TOE – *"The evaluator tests the TOE in the minimum configuration as defined in the ST (and the guidance documentation).*

If the description of the use of extra components in the ST and guidance documentation identifies any difference in the SFRs allocated to a component, or the scope of the SFRs involved (e.g. if different selections apply to different instances of the component) then the evaluator tests these additional SFR cases that were not included in the minimum configuration.

In addition, the evaluator tests the following aspects for each extra component that is identified as allowed in the distributed TOE:

- *Communications: the evaluator follows the guidance documentation to confirm, by testing, that any additional connections introduced with the extra component and not present in the minimum configuration are consistent with the requirements stated in the ST (e.g. with regard to protocols and ciphersuites used). An example of such an additional connection would be if a single instance of the component is present in the minimum configuration and adding a duplicate component then introduces an extra communication between the two instances. Another example might be if the use of the additional components necessitated the use of a connection to an external authentication server instead of using locally stored credentials.*

- *Audit: the evaluator confirms that the audit records from different instances of a component can be distinguished so that it is clear which instance generated the record.*

- *Management: if the extra component manages other components in the distributed TOE then the evaluator shall follow the guidance documentation to confirm that management via the extra component uses the same roles and role holders for administrators as for the component in the minimum configuration."*

The evaluation team's review of ST and AGD determined that the description of the use of extra TOE components did not describe any difference in the SFRs allocated to the TOE components or increase the scope of the SFRs included within the evaluation. Therefore, the evaluation team determined that the claims for the additional TOE component instances were consistent with their equivalent components in the minimum configuration. For all the following test cases, at least one Android and one iOS device were already currently enrolled into MDM, such that, when testing the following *Communication* and *Audit* related test-cases on the second Android and second iOS device, there were at least two concurrently enrolled devices for each platform.

Communication:

- FCO_CPC_EXT.1 – Repeated these tests on a second Hub Agent on a second Android device and a second Hub Agent on a second iOS device
- FPT_ITT.1(2) – Repeated these tests on a second Hub Agent on a second Android device and a second Hub Agent on a second iOS device
- FTP_TRP.1(2) – Repeated these tests on a second Hub Agent on a second Android device and a second Hub Agent on a second iOS device
- FIA_ENR_EXT.2 – This was tested in conjunction with FTP_TRP.1(2) activities since that SFR covers agent enrollment
- FIA_ENR_EXT.1/ANDROID & FIA_ENR_EXT.1/IOS – This was tested in conjunction with FTP_TRP.1(2) activities since that SFR covers agent enrollment

The evaluation team found that when testing a second instance of the iOS Hub Agent and Android Hub Agent TOE components that no additional connections were introduced above those defined in the minimum configuration. The evaluation team also found the connections between the first and second iOS Hub Agents to the UEM Server were identical with regards to the SFR claims. The evaluation team also found the connections between the first and second Android Hub Agents to the UEM Server were identical with regards to the SFR claims.

Audit:

- FAU_GEN.1(2):
    - FIA_ENR_EXT.2
    - FAU_ALT_EXT.2

In conjunction with performing the communication tests, the required level of audit was generated for the above SFRs. The evaluation team verified that the records corresponding to the second Android Hub Agent and second iOS Hub Agent were distinguishable from their minimum configuration TOE component equivalents.

Management:

The set of Management SFRs from AGENTMOD are: FMT_POL_EXT.2, FMT_UNR_EXT.1, and FMT_SMF_EXT.4

According to section E.3 of the MDMPP:

> “Management: if the extra component manages other components in the distributed TOE then the evaluator shall follow the guidance documentation to confirm that management via the extra component uses the same roles and role holders for administrators as for the component in the minimum configuration.”

Additional management testing beyond the minimum configuration would only apply in the case where any of the TOE Hub Agent components managed other TOE components. In FMT_POL_EXT.2, FMT_UNR_EXT.1, and FMT_SMF_EXT.4 the Hub Agent is responsible only for management of its own TOE component functions. Therefore, additional management testing beyond the minimum configuration is not required.

The evaluation team determined through testing that adding additional iOS Hub Agents and/or additional Android Hub Agents to the minimum configuration did not impact the SFR claims for this evaluation, and that SFR functions being addressed by the additional TOE components were able to be distinguished from their minimum configuration counterparts.

7 Conclusions

The TOE was evaluated against the ST and has been found by this evaluation team to be conformant with the ST. The overall verdict for this evaluation is: Pass.

8 Glossary of Terms

Acronym	Definition
APNS	Apple Push Notification Service
CA	Certificate Authority
CC	Common Criteria
CPU	Central Processing Unit
CSP	Critical Security Parameter
DEP	[Apple] Device Enrollment Program
FCM	[Android] Firebase Cloud Messaging [Service]
FQDN	Fully Qualified Domain Name
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure over a bidirectional TLS encrypted tunnel
IMEI	International Mobile Equipment Identity
IP	Internet Protocol
IIS	Internet Information Services
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
MAS	Mobile Application Store
MD	Mobile Device
MDM	Mobile Device Management
NFC	Near-Field Communication
NIAP	National Information Assurance Partnership
OS	Operating System
OSP	Organizational Security Policy
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
UEM	Unified Endpoint Management
UI	User Interface
USB	Universal Serial Bus
VPN	Virtual Private Network
WLAN	Wireless Local Area Network

Table 7-1: Acronyms

Term	Definition
End User	An individual who possesses a mobile device that is managed by VMware and who has limited authority to perform management functions using the Self-Service Portal
Internal Apps	Apps which are stored on the TOE’s MAS Server.
Managed Apps	Apps which are stored on the Google Play Store and/or Apple App Store that are installed on the mobile device due to TOE policies.

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 a VMware environment, including the underlying Windows Server 2019 platform.

Table 7-2: Customer Specific Terminology

Term	Definition
Administrator	The claimed Protection Profile defines an Administrator as the person who is responsible for management activities, including setting the policy that is applied by the enterprise on the mobile device. This TOE defines separate user roles.
Authorized Administrator	Synonymous with Administrator.
MD User	User with a mobile device (MD).
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 7-3: CC-Specific Terminology